



Jordanian Journal of Computers and Information Technology

April 2016

VOLUME 02

NUMBER 01

ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

J
J
C
I
T

PAGES

PAPERS

1 - 16

A COMPACT PRINTED UWB PACMAN-SHAPED MIMO ANTENNA WITH TWO FREQUENCY REJECTION BANDS

Shaimaa Naser and Nihad Dib

17 - 36

HIGHLY EFFICIENT IMAGE STEGANOGRAPHY USING HAAR DWT FOR HIDING MISCELLANEOUS DATA

Hamad A. Al-Korbi, Ali Al-Ataby, Majid A. Al-Tae and Waleed Al-Nuaimy

37 - 54

CHARACTERIZATION OF SHARED-MEMORY MULTI-CORE APPLICATIONS

Mohammed Sultan Mohammed and Gheith A. Abandah

55 - 67

A LOW COMPLEXITY DIRECTION FINDING SYSTEM BASED ON A SIX-PORT INTEGRATED MIMO ANTENNA SYSTEM

Rifaqat Hussain, Ali H. Muqaibel, Wajih Abu-Al-Saud and Mohammad S. Sharawi

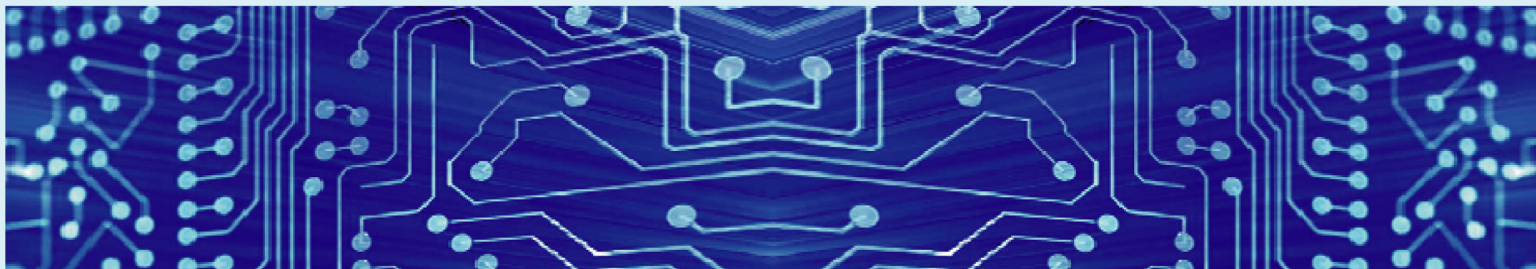
68 - 85

PERFORMANCE EVALUATION OF META-HEURISTICS IN ENERGY AWARE REAL-TIME SCHEDULING PROBLEMS

Ashraf Suyyagh, Jason G. Tong and Zeljko Zilic

www.jjcit.org

jjcit@psut.edu.jo



An International Peer-Reviewed Scientific Journal
Financed by the Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

AIMS AND SCOPE

The JJCIT aims to publish the most current developments in the form of original articles and review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide.

The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

INDEXING

JJCIT is indexed in:

- ScopeMed: www.scopemed.org
- Index Scholar: www.indexscholar.com

EDITORIAL BOARD SUPPORT TEAM

LANGUAGE EDITOR

Haydar Al-Momani

EDITORIAL BOARD SECRETARY

Eyad Al-Kouz

JJCIT ADDRESS

WEBSITE: www.jcit.org

EMAIL: jjcit@psut.edu.jo

ADDRESS: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha.

B.O. BOX: 1438 Amman 11941 Jordan.

TELEPHONE: +962-6-5359949.

FAX: +962-6-7295534.

EDITORIAL BOARD

Ahmad Hiasat (EIC)

Dia Abu-Al-Nadi

"Moh'd Belal" Al-Zoubi

Sameer Bataineh

Ahmad Alshamali

Ismail Ababneh

Mohammad Mismar

Taisir Alghanim

INTERNATIONAL ADVISORY BOARD

Ahmed Yassin Al-Dubai
UK

Chip Hong Chang
SINGAPORE

Fawaz Al-Karmi
JORDAN

Gian Carlo Cardarilli
ITALY

João Barroso
PORTUGAL

Khaled Assaleh
UAE

Lewis Mackenzies
UK

Marc Dacier
QATAR

Martin T. Hagan
USA

Michael Ullman
USA

Mohammed Benaissa
UK

Nadim Obaid
JORDAN

Omar Al-Jarrah
JORDAN

Paul G. Plöger
GERMANY

Shambhu J. Upadhyaya
USA

Albert Y. Zomaya
AUSTRALIA

Enrique J. Gomez Aguilera
SPAIN

George Ghinea
UK

Issam Za'balawi
JORDAN

Karem Sakallah
USA

Laurent-Stephane Didier
FRANCE

Makhlouf Omar
JORDAN

Marco Winzker
GERMANY

Marwan M. Krunz
USA

Mohammad Alhaj Hasan
JORDAN

Mowafaq Al-Omosh
JORDAN

Nazim Madhavji
CANADA

Othman Khalifa
MALAYSIA

Shahrul Azman Mohd Noah
MALAYSIA

Wejdan Abu Elhaija
JORDAN

"Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund".

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي".

EDITOR'S NOTE

This issue is a special one, dedicated to the 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT 2015). AEECT 2015 is a listed IEEE conference and is the third of a series of conferences organized by the IEEE – Jordan Section. This year, AEECT has been jointly organized with Princess Sumaya University for Technology (PSUT). The authors of papers that were rated as the best in the tracks of Computers & Networks, IT Applications & Systems and Communications were invited to submit an extended version of their papers and research results to JJCIT.

Nevertheless, the same criteria of high quality, originality and significance of research results that are part of the Journal's policy have been applied to the invited articles in this Special Issue. All submitted manuscripts have undergone a very rigorous peer review process, based on initial Editorial Board screening and reviewing by at least three expert reviewers. The acceptance rate of the invited papers was 63%.

I would like to take this opportunity to sincerely express my thanks and gratitude to all the reviewers of the Journal for their remarkable efforts to guarantee a timely review and a high-quality process. Moreover, I would like to thank the Editorial Board's Secretary Mr. Eyad Al-Kouz and the Language Editor Mr. Haydar Al-Momani for their commitment, efforts and the professionalism they have shown at the highest level to establish and maintain the high standard and quality of the Journal.

While JJCIT released its first issue late last year, three issues will be released in 2016; April's issue, August's and December's. Starting 2017, we are expecting to reach our steady state of publishing four issues per annum; that is, one issue per quarter.

My colleagues in the Editorial Board and I are still and will continue to eagerly receive your ideas and thoughts to improve the content quality and Journal presentation at the email of the Journal.

Editor-in-Chief

Ahmad Hiasat

A COMPACT PRINTED UWB PACMAN-SHAPED MIMO ANTENNA WITH TWO FREQUENCY REJECTION BANDS

Shaimaa Naser¹ and Nihad Dib²

Department of Electrical Engineering, Jordan University of Science and Technology
P. O. Box 3030, Irbid 22110, Jordan.

²At the present time, on sabbatical leave at German Jordanian Univ.
naser.shaimaa@yahoo.com¹, nihad@just.edu.jo²

(Received: 23-Nov.-2015, Revised: 13-Dec.-2015, Accepted: 07-Jan.-2016)

ABSTRACT

In this paper, the design, analysis and prototyping of a microstrip-fed, low profile, compact ultra-wideband (UWB) monopole antenna with two band notches are presented. The antenna is then used in two multiple-input multiple-output (MIMO) configurations. The antennas are mounted on a low cost FR-4 substrate of a dielectric constant of 4.4. The original shape of the single antenna element is circular with a radius of 11.5 mm, then a sector is removed from the patch (making it a Pacman-shaped antenna) to improve the impedance bandwidth. The proposed antennas provide an impedance bandwidth between 2.9-15 GHz with better than 10 dB return loss and isolation of more than 16 dB and 19 dB for the first and the second MIMO configurations, respectively. Additionally, the antennas can reject the interferences from Worldwide Interoperability for Microwave Access (WiMAX) (3.5 GHz center frequency) and Wireless Local Area Network (WLAN) (5.5 GHz center frequency).

KEYWORDS

Ultra-wideband (UWB) antennas, Circular monopole antenna, Multiple-Input Multiple-Output (MIMO).

1. INTRODUCTION

Modern mobile systems require high speed and reliable transmission of data without an increment in bandwidth or transmitted power. Multiple-input multiple-output (MIMO) communication is the way to achieve the aforementioned goals through using multiple antennas, which are suitable for modern standard communications, such as WiFi, WiMAX, 4G, High Speed Packet Access (HSPA+) and UWB. MIMO is based on the use of multiple transmitting and receiving antennas to achieve spatial diversity or spatial multiplexing. Nowadays, UWB MIMO antennas are widely used due to the advantages of providing reliable and high data rate transmission.

Many studies have been conducted on the design and analysis of UWB MIMO antennas. In [1], a planar monopole UWB MIMO antenna that consists of two identical monopoles and a Y-shaped decoupling network fixed on the ground plane was investigated. The Y-shaped decoupling network provides an isolation of more than 20 dB and a correlation of less than 0.01. In [2], a compact UWB MIMO antenna with better than 15 dB isolation was proposed. The antenna elements were circular disc monopole antennas with a common ground. The isolation was improved through using an inverted-Y stub inserted on the ground. MIMO antennas can be designed to have two polarizations to achieve diversity. In [3], a single radiator was shared between two antenna elements, while diversity was achieved through having two different

polarizations. A stub in the ground and a T-shaped slot were etched from the radiator to enhance isolation. In [4], a dual polarization MIMO antenna was proposed through using two monopole antenna elements perpendicular to each other. Rejection of the WLAN band was achieved through etching an H-shaped slot and a resonant L-shaped strip.

In [5], two identical antenna elements were used to form an UWB MIMO antenna with 17 dB isolation. The single element consisted of seven circles surrounding a central circle. In [6], a UWB diversity slot antenna was investigated. The structure of the antenna consisted of two modified coplanar waveguides (CPWs) feeding staircase-shaped radiating elements for orthogonal radiation patterns, where a rectangular stub was placed between the two feeding CPWs to ensure high isolations. By etching two split-ring resonator (SRR) slots on the radiators, the band-notched property was achieved. In [7], an ultra-wideband MIMO antenna, which consisted of two elliptical-shaped monopoles, was proposed. Two stubs and a slot were introduced to reduce the mutual coupling between the two elements. Results showed that the antenna works in the band 3.1-10.6 GHz and has an isolation of more than 20 dB. In [8], a compact MIMO antenna that covers the WLAN (2.4 GHz) and UWB range was presented. The proposed antenna consisted of two open L-shaped slot antenna elements and a narrow slot on the ground plane. The isolation was larger than 20 dB in the WLAN band and 18 dB in the UWB range. Finally, [9] proposed a MIMO antenna in which each element consisted of a planar-monopole antenna printed on one side of the substrate, where the elements were placed perpendicular to each other. To enhance isolation and increase impedance bandwidth, two long protruding ground stubs were added to the ground plane. The antenna achieved an isolation larger than 15 dB and a correlation of less than 0.2 in the UWB range.

2. SINGLE UWB ANTENNA

Figure 1 illustrates the structure of the single UWB antenna. The antenna is mounted on a compact size FR-4 substrate of dimensions $25 \times 38 \text{ mm}^2$, a dielectric constant of 4.4, a loss tangent of 0.02 and a thickness of 1.6 mm. The original patch has a circular shape, since it has the largest bandwidth among the other regular shapes and has good radiation characteristics [10]. The radius was approximated to be $\lambda/4$ at the lower frequency edge of the UWB range [11]. A partial ground plane is used with a notch cut near the feeding line to improve the impedance bandwidth. It has been found that the distance between the feeding point and the ground plane ($p = L_{feeding} - W_{gnd}$) has an effect on the antenna performance. Its value was chosen to be 0.2 mm. Then, a sector was removed from the circular patch (making it a *Pacman*-shaped antenna) to improve the impedance bandwidth. Finally, a U-shaped slot and a straight slot were etched in the patch to reject the interference from WiMAX and WLAN [12], respectively. The total lengths of the slots were approximated to be $\lambda/2$ at the notched frequencies [13]-[15]. Several simulations were performed using HFSS version 14 [16] to get the optimized parameters listed in Table 1. The simulated and the measured VSWR of the proposed UWB antenna are illustrated in Figure 2.

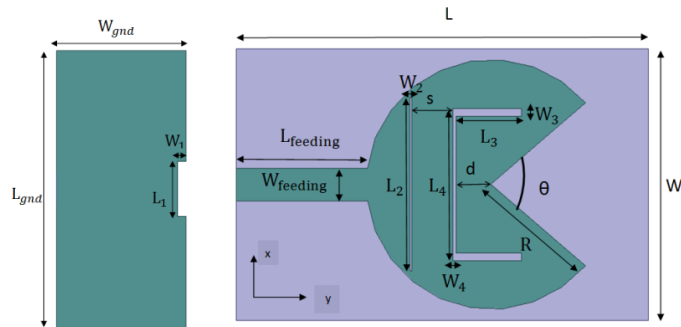


Figure 1. The structure of the proposed UWB antenna.

Table 1. The optimized parameters of the proposed UWB antenna.

Parameter	Value	Parameter	Value
L	38 mm	W	25 mm
$L_{feeding}$	12 mm	$W_{feeding}$	3 mm
L_{gnd}	25 mm	W_{gnd}	11.8 mm
R	11.5 mm	W_1	0.8 mm
L_1	5 mm	W_2	0.2 mm
L_2	15.8 mm	W_3	0.7 mm
L_3	6 mm	W_4	0.3 mm
L_4	14 mm	θ	80°
S	3.8 mm	d	3.1 mm

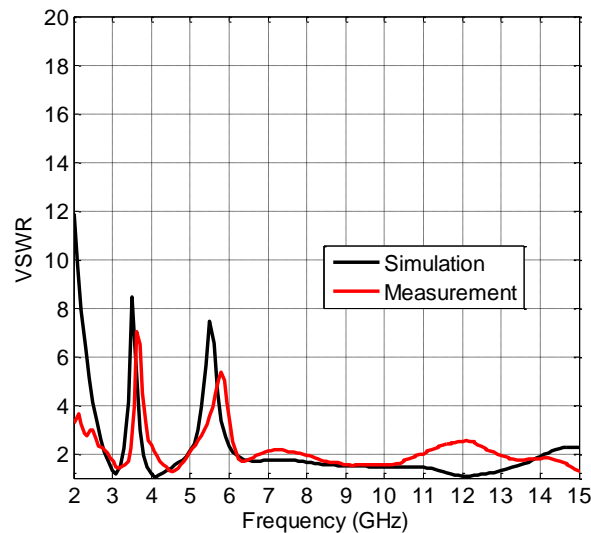


Figure 2. The simulated and measured VSWR of the proposed UWB antenna.

It can be observed that measurement agrees well with simulation, except in the range 11-13 GHz, which could be due to experimental tolerances, fabrication tolerances and the effect of the connector. The antenna works in the frequency band 2.9-15 GHz with the VSWR being less than 2, except around the notched frequencies. As desired, the antenna has filter characteristics around 3.5 GHz (the center frequency of WiMAX) and 5.5 GHz (the center frequency of WLAN). A small shift in the measured notched frequencies can be noticed, because the simulation environment is different from the real environment, as well as the fact that the substrate relative permittivity decreases as the frequency increases [17].

3. UWB MIMO ANTENNAS

In this section, utilizing the designed Pacman-shaped UWB antenna, two MIMO configurations are proposed. The first configuration is shown in Figure 3, in which the antenna elements are placed side by side a distance D from each other with separate ground planes and mounted on the same substrate. The two antenna elements are symmetric and have the same optimized parameters obtained for the single Pacman-shaped UWB antenna. The other configuration is

shown in Figure 4, in which the antenna elements are orthogonally placed on the same substrate and have separate ground planes. The centers of the circles are placed a vertical distance V and a horizontal distance h from each other, while having the same optimized parameters obtained for the single Pacman-shaped UWB antenna. After running several simulations and performing a parametric study, the optimum distances between the antenna elements are chosen as follows: $D = 23$ mm, $h = 24$ mm and $V = 2$ mm downward (i.e., the center of the right antenna is lower than the center of the left antenna).

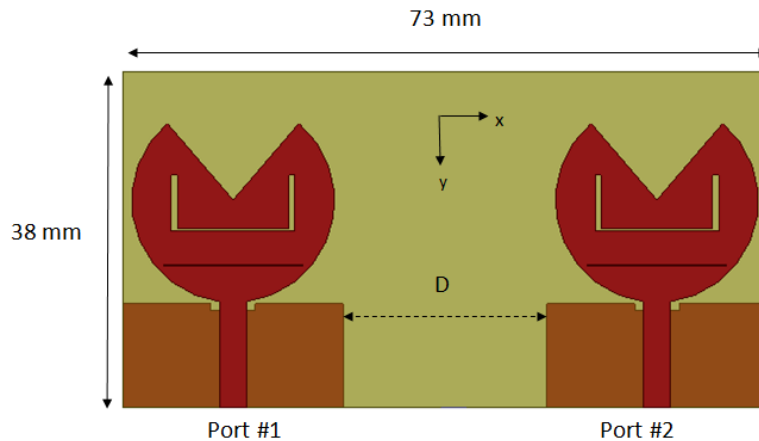


Figure 3. The structure of the proposed MIMO configuration #1.

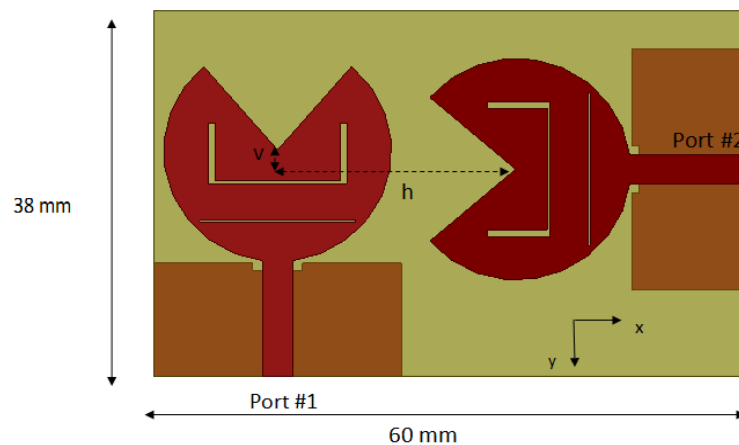


Figure 4. The structure of the proposed MIMO configuration #2.

The optimized distances were used to build the two configurations, then measurements were performed in the laboratory using Agilent VNA. Figure 5 shows a picture of the fabricated MIMO antennas.

Figure 6 illustrates the voltage standing wave ratio (VSWR) of the first MIMO configuration. It can be noticed that measurements agree well with simulation, except in the band 11-13 GHz, which could be due to the connectors and fabrication tolerance. Also, a shift in the notched frequencies occurs due to the reasons mentioned before. It is difficult to get a symmetric structure (i.e., $S_{22} = S_{11}$) in practice due to prototyping tolerances and connectors, which is clear from Figure 6 (i.e., the measured VSWR values of the two ports are somewhat different from each other). Figure 7 illustrates the VSWR of the second MIMO configuration.



Figure 5. Pictures of the fabricated MIMO configurations.

A good agreement exists between simulation and measurement when port 1 is excited, but a small difference between measurement and simulation appears in the band 11-15 GHz when port 2 is excited, which could be due to prototyping tolerances. Also, a shift in the notched frequencies appears as in the first configuration. It is clear that both antenna elements of the two configurations work in the UWB range.

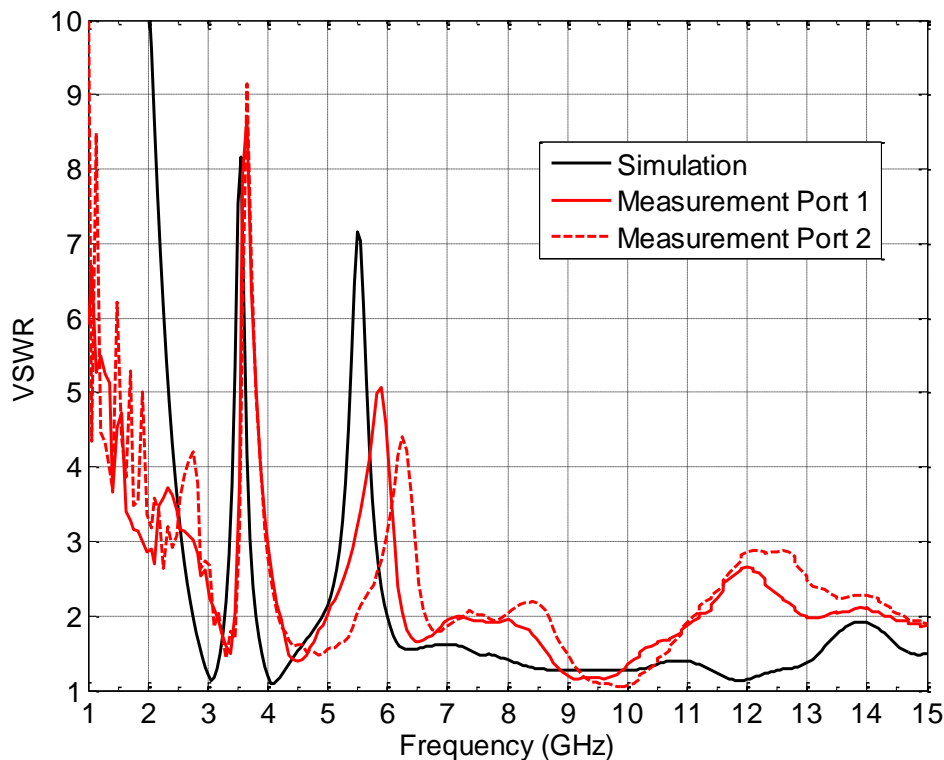


Figure 6. The simulated and measured VSWR of configuration #1.

One of the most important parameters while studying MIMO antennas is the isolation between the input ports. Figure 8 illustrates the isolation of the first and second MIMO configurations (i.e., S_{21} or S_{12} , since the networks are reciprocal ones). A good agreement exists between simulation and measurement with the isolation having measured values of more than 16 dB and 19 dB for the first and the second configuration, respectively. So, the orthogonal placement of the antenna elements achieves better isolation with smaller antenna size.

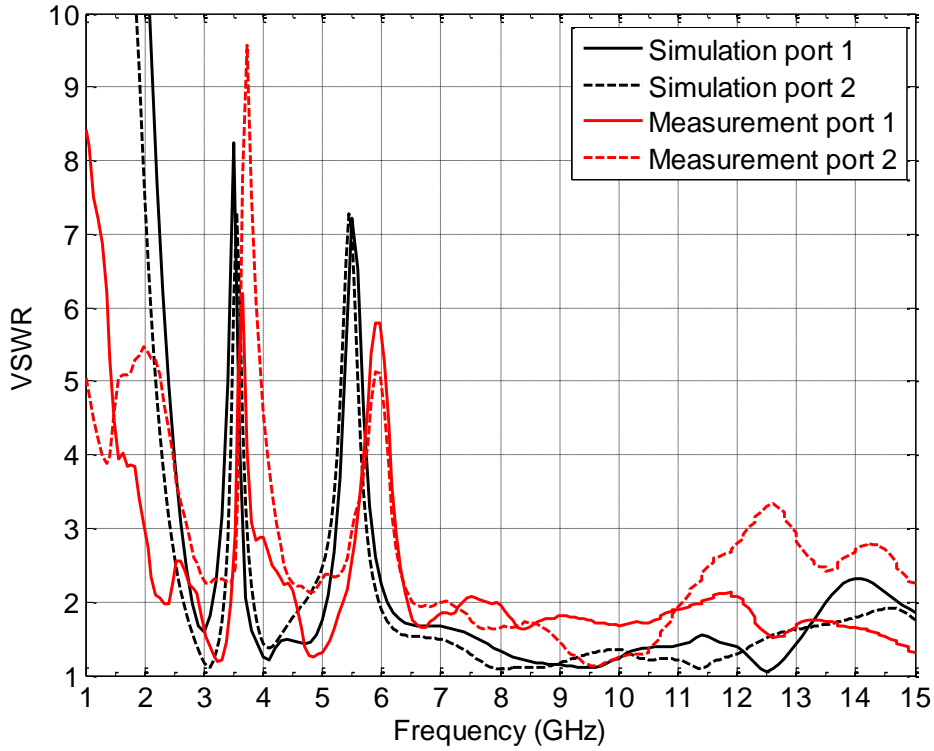
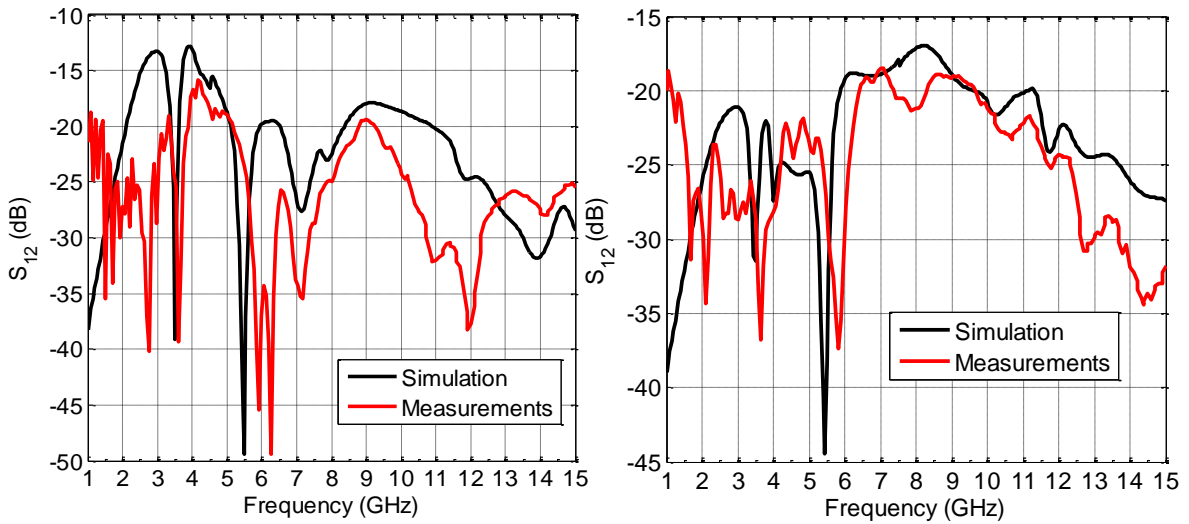


Figure 7. The simulated and measured VSWR of configuration #2.



(a)

(b)

Figure 8. S_{12} (in dB) of the proposed MIMO configurations.

(a) Configuration #1, (b) Configuration #2.

The E-plane and the H-plane patterns of the first configuration are illustrated in Figure 9. The radiation patterns of the first antenna element in Figure 9 (a) are obtained by exciting port 1 and terminating port 2 with a matched load. It is clear that the xz -plane is the H-plane and the yz -plane is the E-plane. As the frequency increases, the radiation pattern becomes distorted. The radiation patterns of the second antenna element in Figure 9 (b) are obtained by exciting the

second port and terminating the first port with a matched load. Due to symmetry, both antenna elements have the same radiation patterns, except that the H-plane of the second element has a 180° shift. This is due to the way that the two antenna elements are placed beside each other. So, this configuration provides only spatial diversity.

The E-plane and the H-plane patterns of the second proposed UWB MIMO antenna are illustrated in Figure 10. The radiation patterns of the first antenna element in Figure 10 (a) are obtained by exciting port 1 and terminating port 2 with a matched load. It is difficult to obtain a pure omni-directional pattern, due to coupling between the antenna elements. It can be noticed that the xz -plane is the H-plane and the yz -plane is the E-plane. The radiation patterns of the second antenna element in Figure 10 (b) are obtained by exciting the second port and terminating the first port with a matched load. Here, the xz -plane is the E-plane and the yz -plane is the H-plane, which is the opposite of the first element, and this is due to the orthogonal placement of the antennas in the second configuration. Since the two elements have different patterns and opposite E-plane and H-plane, this configuration provides pattern and polarization diversity in addition to spatial diversity.

Now, the realized peak gain for each antenna element of both configurations is considered. In Figure 11 (a), the realized peak gain of the first element of configuration #1 is computed by exciting port 1 and terminating the other port with a matched load. The peak gain of the second element is almost the same due to symmetry. So, the result for port 1 is only shown. It can be noticed that the gain increases up to nearly 7 dBi in the whole band, but drops to nearly -8 dBi at the center frequency of WiMAX and to -3 dBi at the center frequency of the WLAN. In Figure 11 (b), the realized peak gain of both antennas in configuration #2 is computed by the same way. The gains of the two elements are slightly different due to asymmetry and they increase up to 7 dBi in the whole band, but drop to -5.5 dBi at the center frequency of WiMAX and to -3.8 dBi at the center frequency of the WLAN.

The current distribution is used to further study the operation of the UWB MIMO antennas. In Figure 12, the current distribution of the first configuration is obtained by exciting port 2 and terminating port 1 with a matched load. In Figure 13, the current distributions of both elements of the second configuration are computed by exciting the desired port and terminating the other with a matched load. It is clear from Figures 12 and 13, that the current is mainly concentrated at the edges of the circular patch and the feeding line of the excited element, except at 5.5 GHz, where the current is mainly concentrated at the WLAN notch and at 3.5 GHz, where the current is mainly concentrated at the U-shaped slot (WiMAX slot), and the current couples from port 1 to port 2 and *vice versa*.

Group delay has an important role in the dispersion characteristics of each antenna element. Due to symmetry in the first configuration, the simulated group delay of both elements is almost the same. So, only the result using S_{11} of the first antenna element is shown in Figure 14 (a), while in measurement one cannot guarantee symmetry; so, the group delay for both elements is shown in the same figure. In Figure 14 (b), the simulated and measured group delays of both elements of the second configuration are obtained using S_{11} and S_{22} . Both figures show almost a constant group delay, indicating that the dispersion is very small.

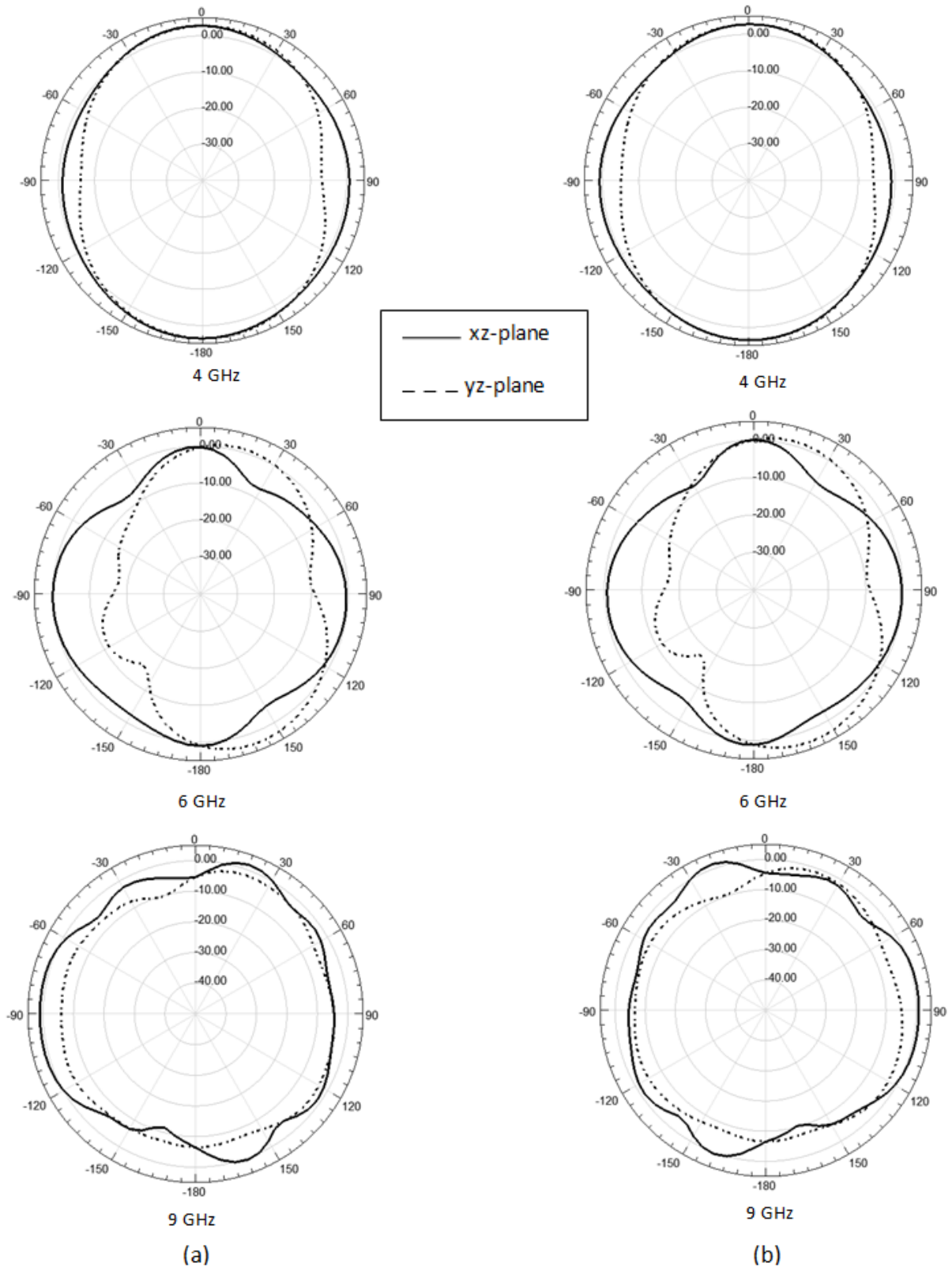


Figure 9. The simulated E-plane and H-plane radiation patterns (in dB) at 4, 6 and 9 GHz for the first MIMO configuration.

(a) Port 1 excited, (b) Port 2 excited.

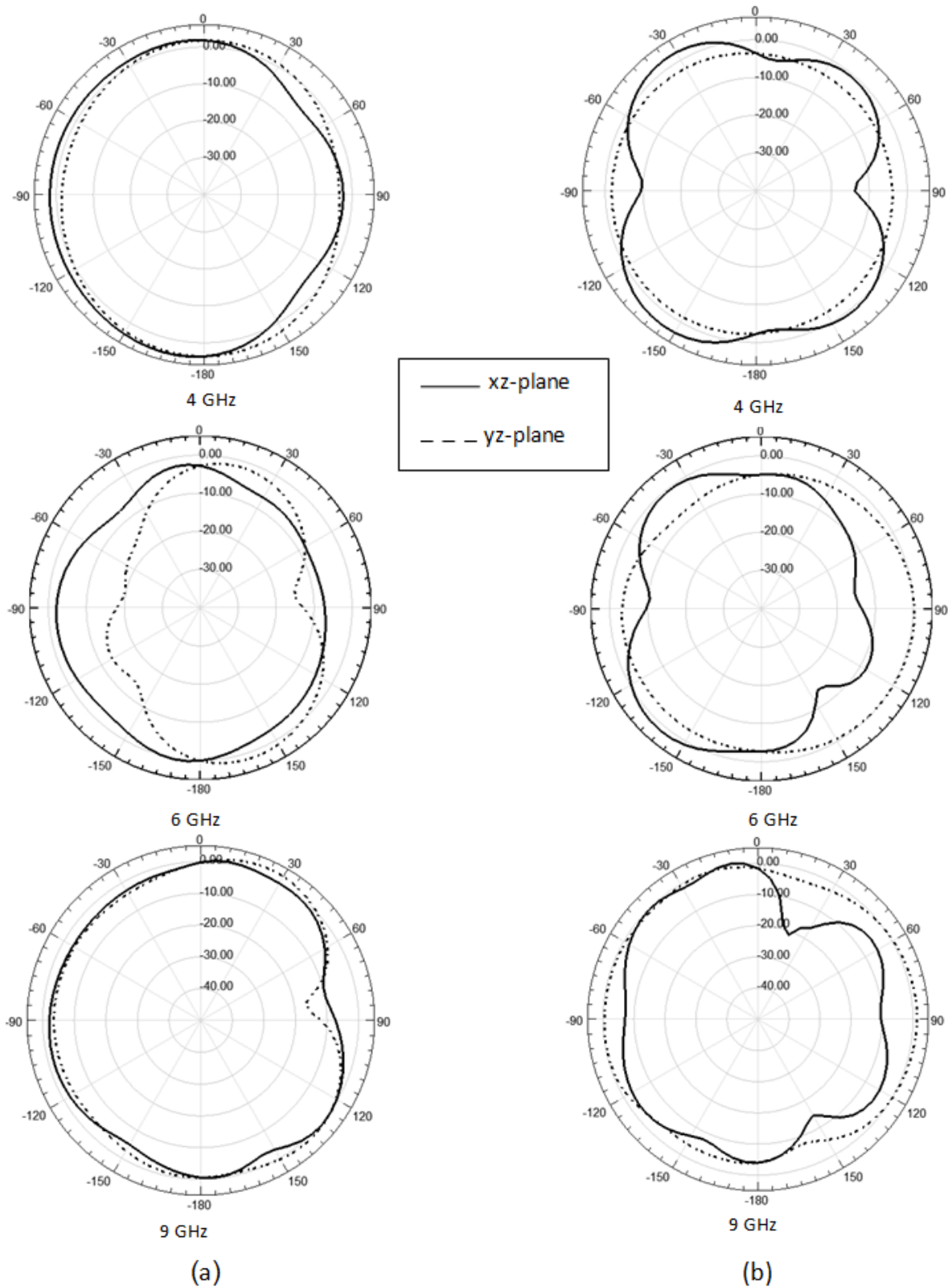
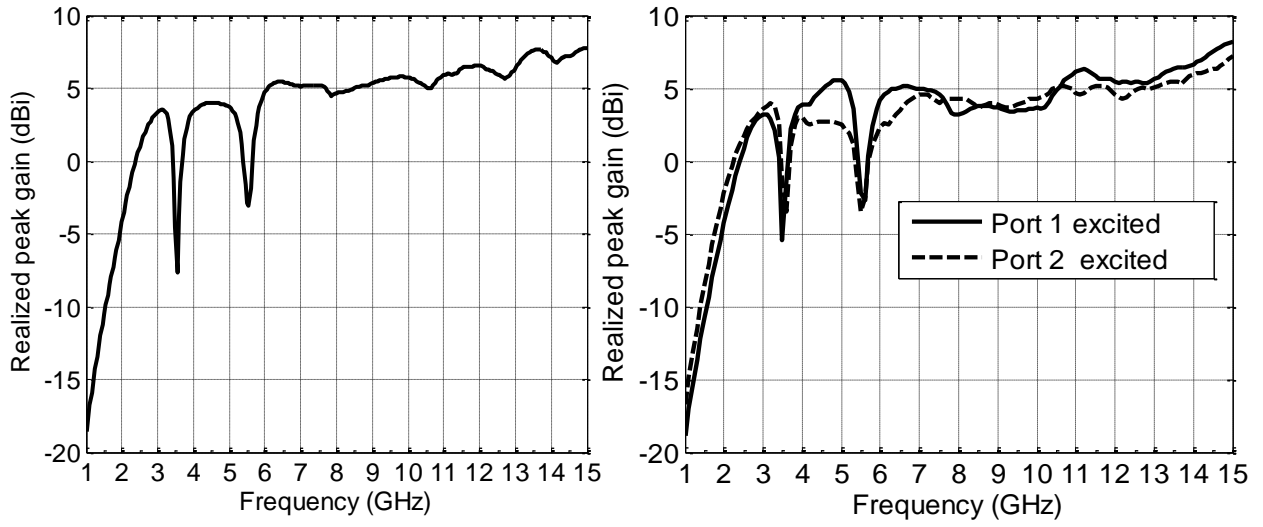


Figure 10. The simulated E-plane and H-plane radiation patterns (in dB) at 4, 6 and 9 GHz for the second MIMO configuration.

(a) Port 1 excited, (b) Port 2 excited.



(a) (b)
 Figure 11. The realized peak gain for both MIMO configurations.
 (a) Configuration #1, (b) Configuration #2.

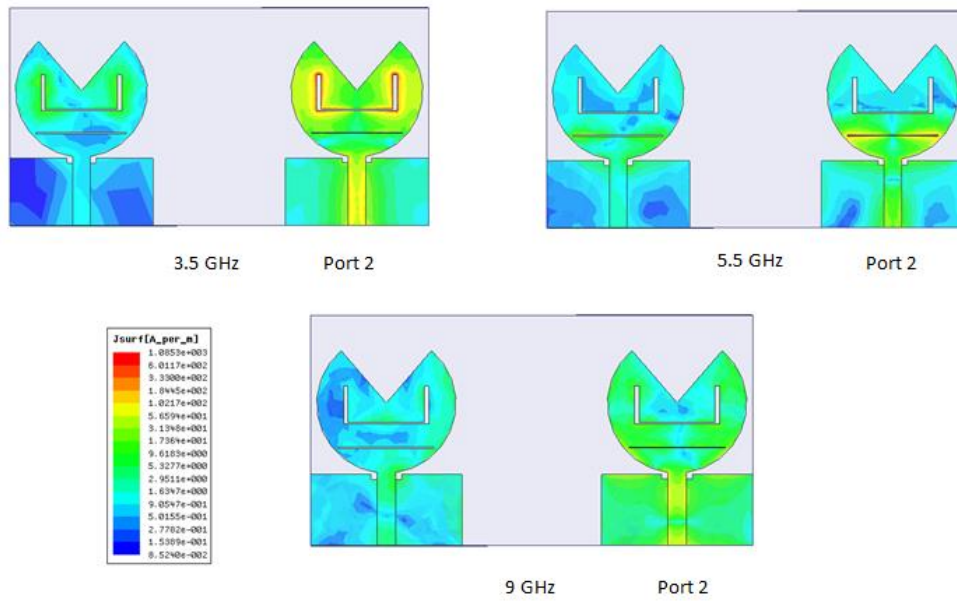


Figure 12. The current distribution of configuration #1 (port 2 excited) at 3.5, 5.5 and 9 GHz.

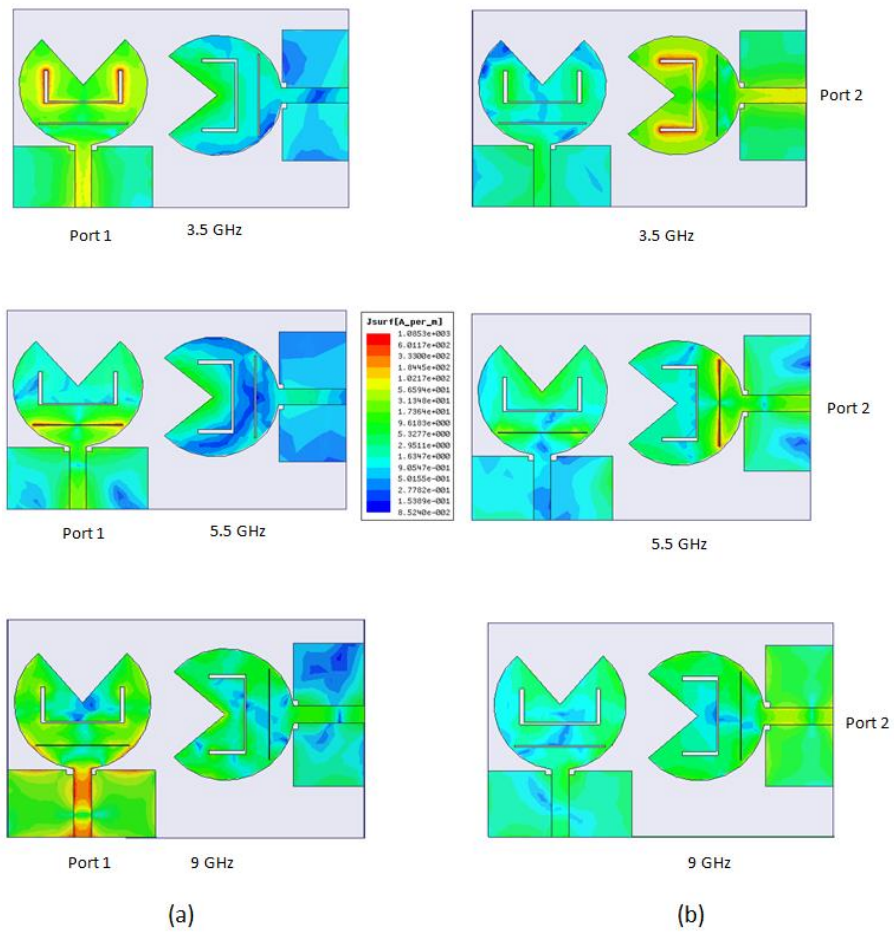


Figure 13. The current distribution of configuration #2 at 3.5, 5.5 and 9 GHz. (a) Port 1 excited, (b) Port 2 excited.

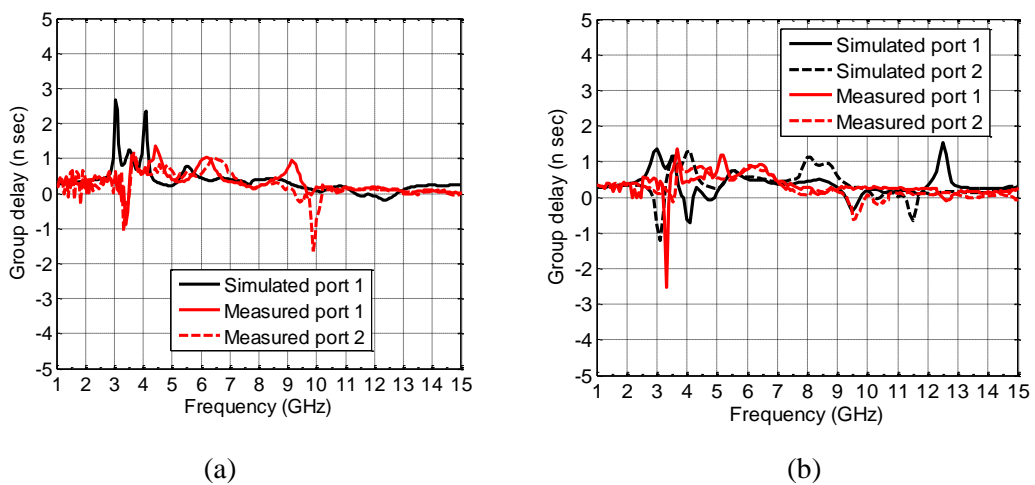


Figure 14. Simulated and measured group delay for both MIMO configurations. (a) Configuration #1, (b) Configuration #2.

Another important parameter for MIMO antennas is the envelope correlation coefficient (ECC), which determines how much the communication channels are isolated. In other words, it describes how much the radiation patterns affect each other. For antennas having efficiency larger than 50%, the ECC can be computed from the scattering parameters [18]-[20]. The efficiency for both configurations is shown in Figure 15. Due to symmetry in the first configuration, the efficiency for both antenna elements is almost the same, so the results when port 1 is excited are shown in Figure 15 (a). In Figure 15 (b), the results for both antenna elements are shown. It is obvious that both configurations have efficiencies larger than 50% in the whole UWB range, except at the notched bands.

A value of 0.5 or less is adequate for low correlation between the antenna elements. The envelope correlation coefficients of the first and the second MIMO configurations were computed using the scattering parameters and are illustrated in Figure 16. The ECC of the first configuration in Figure 16 (a) is less than 0.05 in the whole band, which indicates a low correlation between the two elements' radiation patterns. So, the diversity gain will be high.

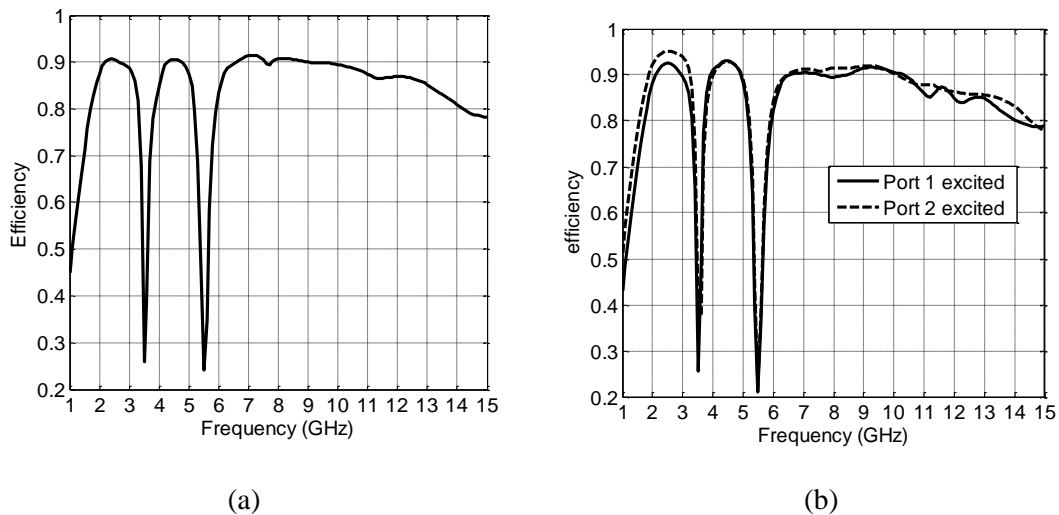


Figure 15. The antenna efficiency for both MIMO configurations.
(a) Configuration #1, (b) Configuration #2.

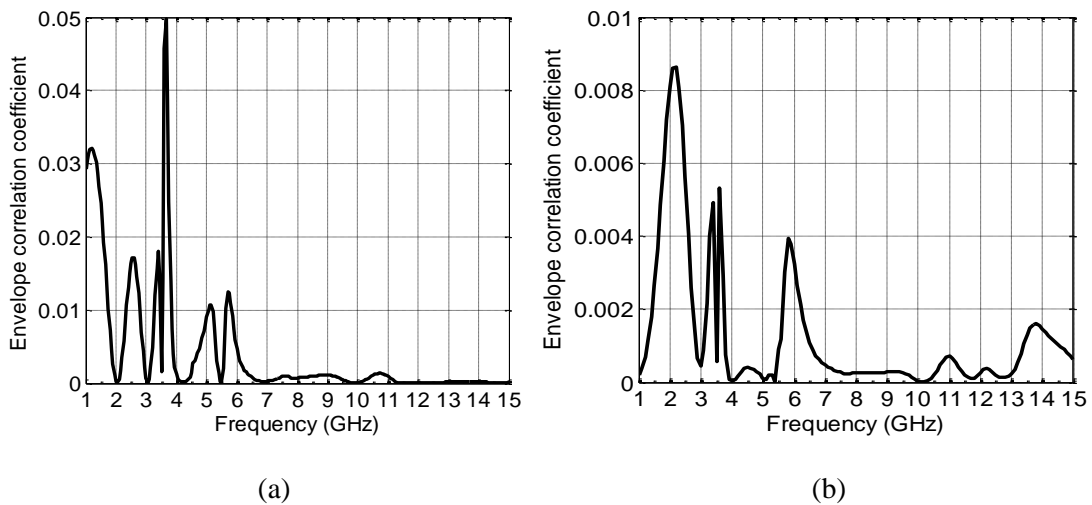


Figure 16. The ECC for both MIMO configurations.
(a) Configuration #1, (b) Configuration #2.

On the other hand, in Figure 16 (b), the second configuration has an ECC of less than 0.009 in the whole band, which indicates an even lower correlation between the two elements' radiation patterns. So, the diversity gain will be even higher. The ECC for the second configuration is smaller than that for the first one, and this is due to the orthogonal placement of the antennas.

Finally, the total active reflection coefficient (TARC) is considered. TARC is used to describe effectively the bandwidth and the efficiency of MIMO antennas. It accounts for coupling and random signal combinations between ports, as well as the effect of a feeding phase to the antenna port. For a desired port excitation, TARC is defined as the square root of the available power generated by all excitations minus radiated power, divided by the available power [18]-[21]. For lossless MIMO antenna, TARC can be computed from the scattering parameters of the antenna. The amplitude of all ports was kept at unity, while the excitation phases were varied with respect to port 1. For various phase differences between the ports' excitations, TARC curves were obtained to see the effect of the phase variation of the two ports on the antenna performance [22]. The TARC curves of the first and the second MIMO configurations are illustrated in Figure 17. It is clear that the bandwidth of the antenna and the notched frequency bands are slightly affected by the phase difference between the two input ports. A good property can be noticed, which is that the center frequency of the notched band is fixed when the phase difference is varied.

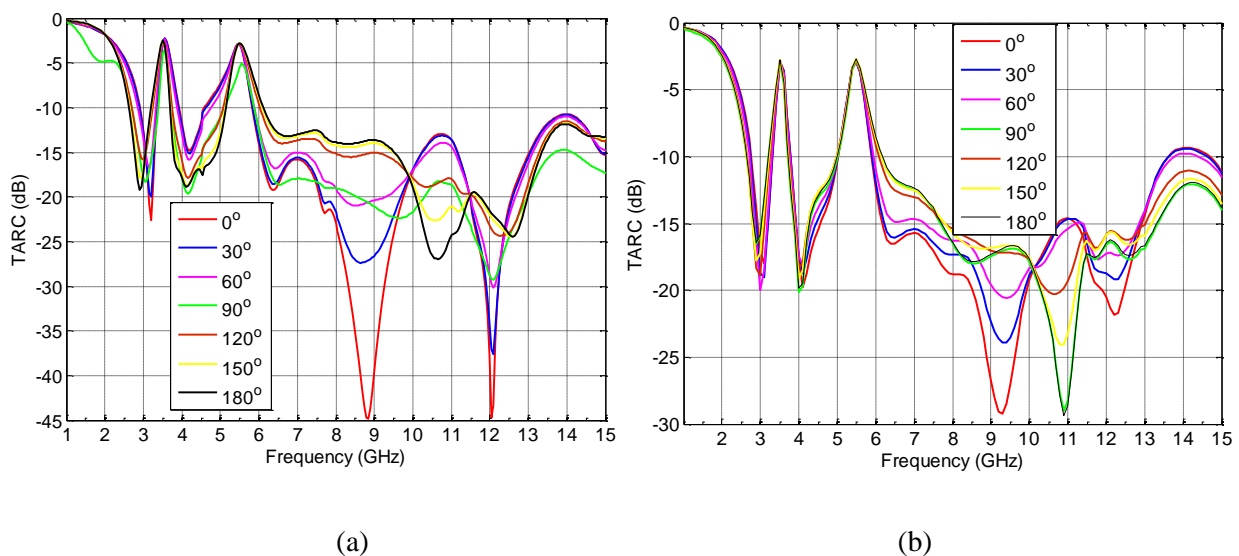


Figure 17. TARC for both MIMO configurations.

(a) Configuration #1, (b) Configuration #2.

Table 2 lists a comparison between the proposed MIMO antennas and some other designs that have appeared in the literature. It is obvious that the proposed MIMO antennas have the largest bandwidth among the others with comparable isolation and ECC. In addition, the proposed antennas provide band rejection characteristics at the WLAN and WiMAX.

Table 2. Comparison between the proposed designs and previous designs.

Ref.	Size (mm ²)	Type	Bandwidth (GHz)	Isolation (dB)	ECC
[1]	27 × 47	Monopole	3.1 -10.6	> 17	< 0.01
[2]	40 × 68	Monopole	3.1 -10.6	> 15	< 0.01
[3]	40 × 40	Monopole	3.1-10.6	> 10	< 0.02
[4]	27 × 37	Monopole	2.35-10.82	> 18	< 0.05
[5]	38 × 91	Monopole	2.8-8	> 17	< 0.006
[6]	48 × 48	Slot	2.5-12	> 15	< 0.005
[7]	45 × 62	Monopole	3.1-10.6	> 20	-
[8]	40 × 40	Slot	3.1-10.6	> 18	< 0.005
[9]	26 × 40	Monopole	3.1-10.6	> 15	< 0.2
Configuration #1	38 × 73	Monopole	2.9-15	> 16	< 0.05
Configuration #2	38 × 60	Monopole	2.9-15	> 19	< 0.009

4. CONCLUSION

In this paper, the design and analysis of compact UWB MIMO antennas with two rejection bands were carried out. Two MIMO configurations were presented. In the first configuration, the two elements were placed beside each other, while in the second configuration, the two antenna elements were placed orthogonal to each other. Both elements work in the range 2.9-15 GHz with better than 10 dB return loss and an isolation of more than 16 dB and 19 for the first and the second MIMO configuration, respectively.

REFERENCES

- [1] M. Khanl, M. F. Shafique, A. Capobianco, E. Autizi and I. Shoaib, "Compact UWB-MIMO Antenna Array with a Novel Decoupling Structure," IEEE, International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 347-350, Islamabad, Pakistan, 15-19 Jan. 2013.
- [2] A. Najam, Y. Duroc and S. Tedjni, "UWB-MIMO Antenna with Novel Stub Structure," Progress in Electromagnetics Research C, vol. 19, pp. 245-257, 2011.
- [3] Ch. Mao and Q. Chu, "Compact Coradiator UWB-MIMO Antenna with Dual Polarization," IEEE Transactions on Antennas and Propagation, vol. 62, no. 9, pp. 4474-4480, September 2014.
- [4] J. Zhao, Z. Zhang, Q. Liu, G. Fu and Sh. Gong, "Printed UWB MIMO Antenna with Different Polarizations and Band-Notch Characteristics," Progress in Electromagnetics Research Letters, vol. 46, pp. 113-118, 2014.
- [5] M. Jusoh, M. Jamlos, M. Kamarudin and F. Malek, "A MIMO Antenna Design Challenges for UWB Applications," Progress in Electromagnetics Research B, vol. 36, pp. 357-371, 2012.
- [6] P. Gao, Sh. He, X. Wei, Z. Xu, N. Wang and Y. Zheng, "Compact Printed UWB Diversity Slot Antenna with 5.5-GHz Band-Notched Characteristics," IEEE Antennas and Wireless Propagation Letters, vol. 13, pp. 367-379, 2014.

- [7] Y. Cheng, W. Lu and Ch. Cheng, "Printed Diversity Antenna for Ultra-Wideband Applications," 2010 IEEE International Conference on Ultra-Wideband (ICUWB2010), vol. 1, pp. 1-4, 20-23 September 2010.
- [8] J. Ren, D. Mi and Y. Yin, "Compact Ultra-Wideband MIMO Antenna with WLAN/UWB Bands Coverage," Progress in Electromagnetics Research C, vol. 50, pp. 121-129, 2014.
- [9] L. Liu, S. Cheung and T. I. Yuk, "Compact MIMO Antenna for Portable Devices in UWB Applications," IEEE Transactions on Antennas and Propagation, vol. 61, no. 8, pp. 4257-4264, August 2013.
- [10] J. Liang, Ch. Chiau, X. Chen and C.-G.Parini, "Study of a Printed Circular Disc Monopole Antenna for UWB Systems," IEEE Transactions on Antennas and Propagation, vol. 53, no. 11, pp. 3500-3504, November 2005.
- [11] T. Wu, H. Bai, P. Li and X. Shi, "A Simple Planar Monopole UWB Slot Antenna with Dual Independently and Reconfigurable Band-Notched Characteristics," International Journal of RF and Microwave Computer-Aided Engineering, vol. 24, issue 6, pp. 706-712, November 2014.
- [12] J. Ren and Y.-Z. Yin, "A Compact Dual Band-Notched Ultra Wideband Antenna with $\lambda/4$ Stub and Open Slots," Progress in Electromagnetics Research C, vol. 49, pp. 133-139, 2014.
- [13] Y.-S. Li, X.-D. Yang, Q. Yang and C.-Y. Liu, "Compact Coplanar Waveguide Fed Ultra-Wideband Antenna with a Notch Band Characteristic," International Journal of Electronics and Communications, vol. 65, no. 11, pp. 961-966, 2011.
- [14] J. Zang and X. Wang, "A Compact C-shaped Printed UWB Antenna with Band-Notched Characteristic," Progress in Electromagnetics Research Letters, vol. 43, pp. 15-23, 2013.
- [15] M. Ojaroudi, G. Ghanbari, N. Ojaroudi and C. Ghobadi, "Small Square Monopole Antenna for UWB Applications with Variable Frequency Band-Notch Function," IEEE Antennas and Wireless Propagation Letters, vol. 8, pp. 1061-1064, 2009.
- [16] ANSYS-High Frequency Structure Simulator (HFSS), Ansys, Inc., Canonsburg, Pennsylvania, USA, 2011.
- [17] S. Mumby and J. Yuan, "Dielectric Properties of FR-4 Laminates as a Function of Thickness and the Electrical Frequency of the Measurement," Journal of Electronic Materials, vol. 18, issue 2, pp. 287-292, March 1989.
- [18] M. S. Sharawi, "Printed Multi-Band MIMO Antenna Systems and Their Performance Metrics," IEEE Antennas and Propagation Magazine, vol. 55, no. 5, pp. 218-232, October 2013.
- [19] M. S. Sharawi, "Printed MIMO Antenna Systems: Performance Metrics, Implementations and Challenges", (invited paper), Forum in Electromagnetic Research Methods and Application Technologies (FERMAT), ART-2014-01-010, pp. 1-11, February 2014.
- [20] M. S. Sharawi, Printed MIMO Antenna Engineering, Artech House, ISBN: 978-1-60807-681-9, 2014.
- [21] M. Manteghi and Y. Rahmat-Samii, "Broadband Characterization of the Total Active Reflection Coefficient of Multiport Antennas," IEEE Antennas and Propagation Society International Symposium, vol. 3, pp. 20-23, 22-27 June 2003.
- [22] M. Sharawi, "A 5-GHz 4/8-Elements MIMO Antenna System for IEEE 802.11AC Devices," Microwave and Optical Technology Letters, vol. 55, no. 7, pp. 1589-1594, July 2013.

ملخص البحث:

في هذا البحث، يتم تصميم هوائي أحادي القطب ذي نطاق فائق العرض وتحليله وبناء نموذج أولي له. والهوائي موضوع البحث هوائي مدمج ذو قطاع جانبي منخفض، مع تلمين في نطاقه الترددي. بعدئذ، يجري استخدام الهوائي في شكلين متعددي المداخل والمخارج. ويتم تركيب الهوائيان على طبقة أساس منخفضة التكاليف ذات ثابت عزل يساوي ٤,٤. الشكل الأصلي لعنصر الهوائي المنفرد دائري بنصف قطر مقداره ١١,٥ مم، ثم أزيل قطاع من التوصيلة المؤقتة (مما يجعله هوائياً على شكل باكمان)؛ من أجل تحسين عرض نطاق الممانعة. وتوفر الهوائيات المقترحة عرض نطاق ممانعة يتراوح من ٢,٩ إلى ١٥ غيغاهيرتز، بفقد إرجاع أفضل من ١٠ ديسيبل، وعزل أكثر من ١٦ ديسيبل و ١٩ ديسيبل للشكلين متعددي المداخل والمخارج الأول والثاني، على الترتيب.

علاوة على ذلك، يمكن للهوائيات أن تتبذ التداخلات الناجمة عن إمكانية التشغيل المتبادل لمنفذ الميكروويف على مستوى العالم (واي ماكس) بتردد مركزي مقداره ٣,٥ غيغاهيرتز، وشبكة المنطقة المحلية اللاسلكية بتردد مركزي مقداره ٥,٥ غيغاهيرتز.

HIGHLY EFFICIENT IMAGE STEGANOGRAPHY USING HAAR DWT FOR HIDING MISCELLANEOUS DATA

Hamad A. Al-Korbi¹, Ali Al-Ataby², Majid A. Al-Tae³ and Waleed Al-Nuaimy⁴

Department of Electrical Engineering and Electronics

University of Liverpool, Liverpool, UK

hamad.a.qa@ieee.org¹; {aliataby², altaeem³, wax⁴}@liv.ac.uk

(Received: 15-Dec.-2015, Revised: 22-Jan.-2016, Accepted: 31-Jan.-2016)

ABSTRACT

Protecting private data exchanged over the Internet and controlling access to this data have become a growing privacy and confidentiality concern. Digital image steganography helps conceal private data within a cover image to obtain a new image, practically indistinguishable from the original, in such a way that unauthorized individuals cannot detect the presence of the concealed data in the new cover. Capacity size of the cover image and imperceptibility are therefore considered critical requirements to assess the performance of steganography algorithms. This paper presents a highly efficient steganography algorithm that is capable of hiding a large size of miscellaneous data (text files, binary images, coloured images or a combination of these data types) in a single cover image using Haar Wavelet transform. Details of the proposed embedding and extraction algorithms for different data types are presented and discussed. The performance of the proposed steganography method is assessed in terms of the capacity of the cover image, imperceptibility and robustness. The obtained experimental results and observations demonstrated that the developed algorithms are highly efficient in terms of the capacity size of the cover image while maintaining a relatively low mean square error (MSE), high peak signal-to-noise ratio (PSNR) and a reasonable robustness against various attacks.

KEYWORDS

Data hiding, Haar Wavelet transform, Information security, LSB, MSE, Pseudo random number, PSNR, Robustness, Steganography.

1. INTRODUCTION

With the worldwide growth of Internet users, security and confidentiality have become a prime importance to protect personal and sensitive data from unauthorized access. Numerous data hiding methods have been reported in the literature to increase the level of information security. Of these, cryptography [1]–[4], steganography [5]–[7] and watermarking [8]–[10] are the most common methods in practice today. Searching Google for cryptography, steganography and watermarking has recently returned 3.9, 0.519 and 0.743 million results, respectively. This provides evidence for the growing importance of information hiding. Unlike cryptography in which the sender converts plaintext to cipher-text (or vice versa) by using an encryption/decryption key, steganography and watermarking are about embedding data within another object known as a cover by tweaking its properties. Steganography and watermarking however differ in their goals, implementations, applications, size of embedded data and robustness requirements.

The term steganography was extracted from a Greek word, meaning covered writing, where ‘stegano’ means ‘cover’, while ‘graphos’ is known as ‘writing’ in English. Its main goal is to

hide a message m in a cover data c , to obtain new data c' , practically indistinguishable from c , by people, in such way that unauthorized individuals cannot *detect the presence of m* in c' . In contrast, the main goal of watermarking is to hide a message m in a cover data c , to obtain new data c' , practically visible or invisible, in such a way that unauthorized individuals cannot *remove or replace m* in c' . Thus, steganography methods usually do not need to provide strong security against removal or modification of the hidden message, while watermarking methods need to be robust enough against attempts to remove or modify the hidden message [8]–[10]. Furthermore, steganography is typically used to conceal a message in one-to-one communications, while watermarking is used whenever the cover-data is available to many parties who are aware of the presence of the hidden data [4].

Popular applications of watermarking are copyright protection and ownership verifications of digital data by embedding copyright statements (visible or invisible), monitoring data transmission in order to control royalty payments or simply tracking the distribution to localize the data for marketing [4]. Image steganography applications on the other hand follow one general principle of hiding a large-size secret data in a single cover image that is exchanged between the communicating parties. The capacity size of the cover image (c) and imperceptibility of the stego image (c') are therefore considered the main critical requirements for steganography.

Peak signal-to-noise ratio (PSNR) and the mean square error (MSE) have been widely used metrics to evaluate the imperceptibility of stego images. In [11], the authors suggested that one secret image in the spatial domain can be concealed within the cover image using the least significant bit (LSB) technique. Random pixels of the cover image will be selected in order to modify their LSB with the most significant bits (MSB) of the secret image or private text; hence the stego image is formed. However, this method provides low PSNR, low MSE as well as low level of the overall security. Another model proposed that an image could be hidden into another image using pixel-value differencing [12]. The PSNR and MSE values were equal to 41.79 dB and 2.07, respectively. Moreover, It has been suggested that discrete cosine transform (DCT) combined with LSB method can be used for enhanced steganography technique [13]. The idea of this method is to convert the images into the frequency domain by applying DCT, and then the secret data will be hidden in the LSB of the DCT coefficients. However, in this method, PSNR value was about 38 dB.

In an effort to develop the system, a steganography algorithm based on the Wavelet transform has also been introduced. In this method, both secret and cover images will be converted from the spatial domain into frequency domain using Wavelet transform. Then, the secret image will be concealed within the cover image using LSB method. The inverse of the Wavelet transform is applied in order to obtain the cover image in the spatial domain. Efficient PSNR and MSE were achieved [5], [7], [14]–[15]. Another method was also developed where both cover and secret images will be decomposed into their three-colour layers R, G and B [16]. Using discrete wavelet transform (DWT), each layer is divided into four levels. After that, alpha combination method is applied to conceal each layer of the secret image. The PSNR value of this method was 29 dB.

Steganography system performance can be improved by applying both DWT and DCT [17]. The cover image can be sub-divided into four sub-bands using DWT and the DCT is applied to the HH sub-band. Secret image is dispersed into HH using session key and sequences pseudo random. Outcome PSNR of this method is 27.39 dB. Moreover, it was suggested that Wavelet transform and genetic algorithm can be used to achieve high capacity image steganography [18]. It was argued that the genetic algorithm based mapping could be used to embed the secret information into the coefficients of the DWT in 4×4 blocks cover image. A high value of PSNR was achieved, along with the capacity; both are equal to 45.2 dB and 50%, respectively.

High capacity data hiding using LSB steganography and encryption is a new field of steganography. This technique using LSB and encryption aimed to have high capacity as well as an acceptable level of the overall security [19]. Furthermore, a new model was developed, where a robust and highly secure steganography algorithm using dual Wavelet and blending mode was applied [20]. Further research was carried out on steganography techniques in order to increase the capacity as well as the PSNR using DWT and Arnold Transform [21]. The capacity and the PSNR were equal to 75% and about 50 dB, respectively. Another steganography technique was also proposed in [22], using DWT and Huffman coding. The achieved PSNR and capacity from this technique were 54.93 dB and 64.5%, respectively. In conclusion, the performance of steganography techniques has been a trade-off among capacity, security, robustness and distortion.

In [23], the authors reported a steganography technique based on the discrete wavelet transform. This technique was capable of hiding a secret message and a small-size image into a large-size image. Another steganography method was also reported in [15] that was capable of hiding one secret image within another single image. However, most of the previously reported steganography techniques support hiding size images or text messages or a combination of both.

Wavelet transform-based steganography techniques are usually criticized because of the inherent complexity and cost of the incorporated algorithms, in such a way that the trade-off between complexity and performance is not justified. In this paper, we present a more efficient steganography technique that extends a previously reported work by the authors in [5] based on Haar wavelet transform. The proposed technique allows hiding any combination of secret images (black and white (B&W) or coloured) and large secret text files can be concealed within a single cover image. All these types of private data can be concealed in a single stego image of a size of $512 \times 512 \times 3$ pixels. In addition, the stego image is formed to be always equal to the cover image.

The remaining of this paper is organized as follows. Background information on the wavelet transform, least significant bit (LSB) and pseudo random number techniques that are adopted in the proposed steganography are presented in Section 2. Details of the proposed embedding and extraction algorithms are given in Section 3. Data hiding scenarios for different combinations of data types are discussed in Section 4. Performance metrics that are used to evaluate the proposed steganography are presented in Section 5. The obtained evaluation results are presented and discussed in Section 6. Finally, the work is concluded in Section 7.

2. BACKGROUND

This section provides theoretical background on wavelet transform, least significant bit (LSB) and pseudo random number techniques.

2.1 Haar DWT

One of the most developed transforms that can be used to transform a signal from the spatial to the frequency domain and vice versa is the Wavelet transform. The Wavelet transform, and other related transforms, can be considered a second generation of transforms. Wavelets are defined as oscillations of short waves that decay rapidly over time [24]. Moreover, they have an enormous number of applications that can be implemented in various fields such as signal processing, data compressing, fingerprint verification, smoothing, image de-noising and speech recognition. It has been reported that the Wavelet transform can be applied to the steganography technique in order to increase the capacity as well as the robustness [25]. One of the Wavelet transform families known as “Haar” has been implemented in this work. It converts an image from spatial domain to frequency domain by applying horizontal and vertical operations, respectively.

The Haar DWT is used in the proposed steganography technique. It is the simplest transform in wavelet mathematics, because it uses square pulses to approximate the original function. It is used to convert the cover image into four sub-bands that are approximation, vertical, horizontal and diagonal coefficients, which represent low-low, high-low, low-high and high-high frequencies, respectively. Approximation coefficients will not be used to conceal secret information, since human eyes are very sensitive to small changes in the low-low frequency. However, the rest of the coefficients contain high frequencies, thus secret data will be corrected and concealed within these bands by the use of both least significant bit and pseudo random number techniques. Once the embedding process is completed, the inverse Haar DWT is applied in order to form the stego image. The vertical and horizontal operations are shown in Figure 1 and described briefly as follows [11].

2.1.1 Horizontal Operation

In this operation, an image will be divided into two bands that are low and high frequencies. Pixels are scanned from left to right in the horizontal direction. Addition and subtraction operations are performed on the neighboring pixels. The results of addition are on the left side that represents the low frequency band. However, subtraction, which represents the high frequency band, is held on the right side, as illustrated in Figure 1 (a).

2.1.2 Vertical Operation

Low and high frequencies obtained from the horizontal operation are further sub-divided into low-low, low-high, high-low and high-high frequencies. All pixels will be scanned over for the addition and subtraction operations, but in the vertical direction. The addition of the neighboring pixels will be held in the top, while the subtraction result will be located in the bottom, as illustrated in Figure 1 (b).

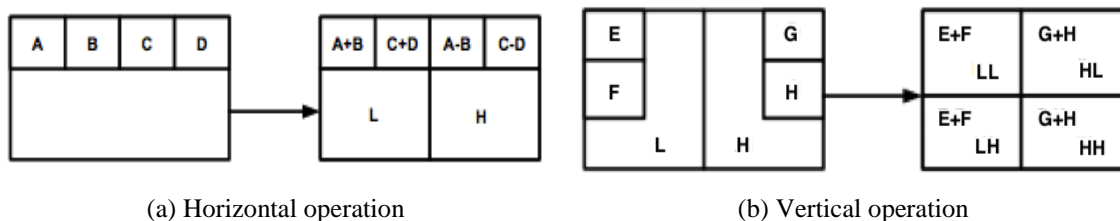


Figure 1. Horizontal and vertical operations.

2.2 LSB Technique

Least significant bit (LSB) is one of the common techniques being used for embedding data. It operates by replacing the least significant bit on one byte by another bit of a secret data. Images are made up of a large number of dots known as pixels and each pixel contains three bytes represented as RGB. Relying on the abundance of each colour, these three bytes will design the various colours of each pixel in the image that will result in changing the whole image colour. For example, the decimal RGB values for the black colour are (0, 0, 0), respectively. In contrast, the decimal RGB values for the white colour are (255, 255, 255). Table 1 shows that as the number of bytes changes, the colour of the pixel will be changed, which will result in changing the colour of the whole image. Furthermore, the range of colour for one byte will be from 0 to 255; that are from black to white.

Table 1. Resultant pixel colour relying on the abundance of each RGB.

	Red Layer (R)	Green Layer (G)	Blue Layer (B)	Resultant Colour
Binary	00000000	00000000	00000000	Black
Decimal	0	0	0	Black
Binary	11111111	11111111	11111111	White
Decimal	255	255	255	White

Changing the least significant bit-plane of one byte will not cause a visible effect on the overall colour of the pixels. The embedding process of the least significant bit technique will therefore replace the least significant bit of the cover medium with the secret data bits. Table 2 shows the effect on the overall colour of the pixel by altering the least significant bit of the cover image. It can be seen that the overall colour of the pixel will remain constant, even if the LSB has been changed. Therefore, hiding the bits of the secret data into the least significant bit of the cover image will not catch the attention of the eavesdroppers.

Table 2. Effect on the final colour of the pixel by changing LSB.

	Red Layer (R)	Green Layer (G)	Blue Layer (B)	Resultant Colour
Binary	10100101	00101010	00101010	Brown
Decimal	165	42	42	Brown
Binary	10100100	00101011	00101011	Brown
Decimal	164	43	43	Brown

2.3 Pseudo Random-Number Technique

In this technique, the secret image is embedded and extracted by a conventional way. The secret image is initially converted into binary representation and resized according to the cover image size in order to be concealed. When the coefficients of the secret image equal 0, a pseudo-random number will be added to the coefficients of the cover image. However, when the secret image equals 1, the cover image will be kept as it is. Nevertheless, at the decoder side, a correlation theory will be implemented in which the original cover image is compared to the stego image. If the coefficients of the stego image equal the coefficients of the original cover image, the value of the secret image will be 1; otherwise, it will be 0. The embedding (or encoding) and extracting (or decoding) process can be explained by the following examples.

Example 1: Embedding process

$$\text{Secret image} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{Cover image} = \begin{bmatrix} 10 & 13 \\ 11 & 14 \end{bmatrix} \quad \text{Stego-image} = \begin{bmatrix} 11 & 13 \\ 12 & 14 \end{bmatrix}$$

Example 2: Extracting process

$$\text{Stego image} = \begin{bmatrix} 11 & 13 \\ 12 & 14 \end{bmatrix} \text{ compares to Cover image} = \begin{bmatrix} 10 & 13 \\ 11 & 14 \end{bmatrix}$$

$$\text{Secret image} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

3. STEGANOGRAPHY ALGORITHMS

The algorithms presented in this section were designed and developed using MATLAB™ environment. Details of the embedding and extraction algorithms of the proposed steganography are described as follows.

3.1 Embedding Algorithm

Figure 2 shows a block diagram for the proposed embedding process of hiding multiple types of information. Steps of the encoding process can be summarized as follows.

Step 1: Select and read cover image, three secret B&W images, one secret colour image and one secret text file.

Inputs: Cover image, three B&W secret images, one coloured secret image and one secret text file.

Process: The user will be asked to choose the cover image, three B&W secret images, one secret colour image and one secret text file in order to be read.

Outputs: Cover image, three B&W secret images, one coloured secret image and one secret text file are read.

End

Step 2: Separate cover image into three planes R, G and B.

Input: Cover image.

Process: Firstly, cover image will be resized to 512×512. After that, planes of the cover image will be separated into three layers that are red, blue and green. These layers will be used to hide various secret data.

Outputs: Separated cover image planes.

End

Step 3: Corrections of R, G and B planes of the cover image.

Inputs: Separated cover image planes.

Process: Firstly, Wavelet transform will be applied to the red-plane where the coefficient sizes excluding approximation will be determined. Secondly, green-plane will be converted into binary vector. Finally, B-plane will be reshaped, and then each pixel will be reduced by one bit.

Outputs: Separated cover image planes corrected.

End

Step 4: Correction of the secret data.

Inputs: Three B&W images, one secret colour image and one secret text file

Process: Firstly, B&W images will be resized according to horizontal, vertical and diagonal coefficients of the R-plane of the cover image. Secondly, secret colour image will be resized to 145 × 150, then reshaped into binary vector. Finally, secret text file will be converted into binary vector. After that, the length of this binary vector will be equalized according to the length of the B-plane of the cover image.

Outputs: Secret data corrected.

End

Step 5: Hide three B&W secret images into red layer.

Inputs: Red layer of the cover image.

Process: Three B&W secret images will be concealed within the red layer of the cover image using the techniques discussed in Section 2.

Output: Stego-red layer.

End

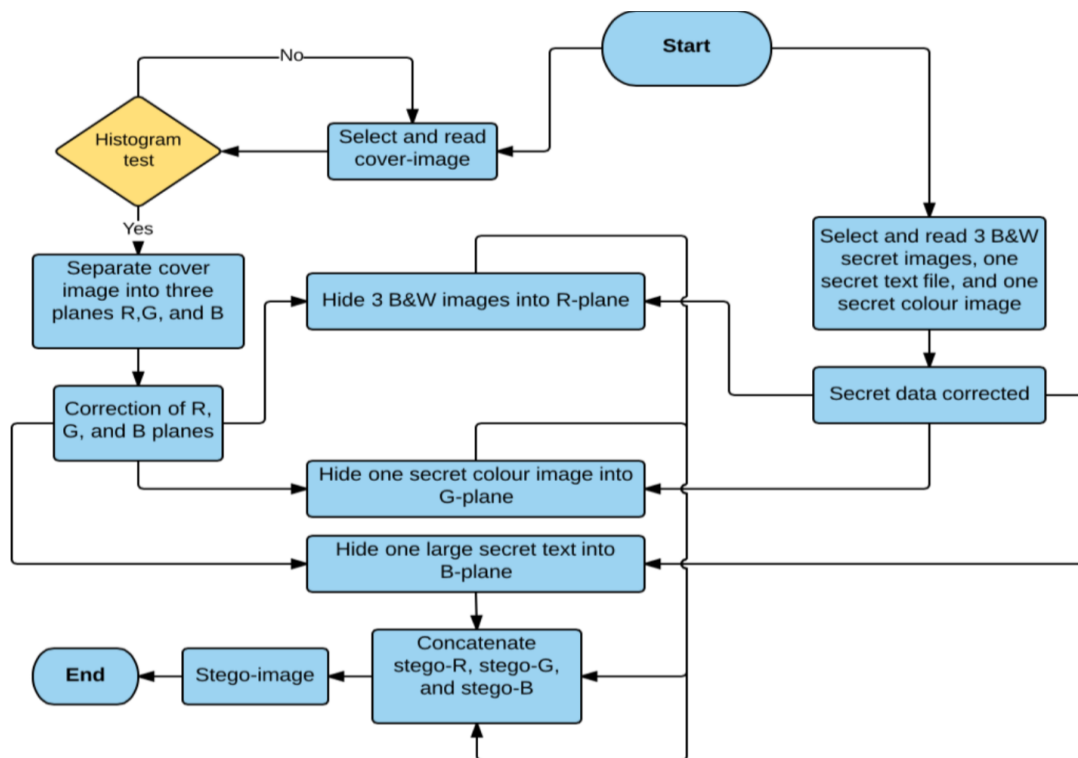


Figure 2. Encoding process of hiding multiple types of information [5].

Step 6: Hide one secret colour image into the blue layer.

Input: Blue layer.

Process: One secret colour image will be concealed within the blue layer.

Output: Stego-blue layer.

End

Step 7: Hide large secret text file into the green layer.

Input: Green layer.

Process: Following the same process that has been applied in step 5, the secret text file can be hidden within the green layer of the cover image.

Output: Stego-green layer.

End

Step 8: Concatenating three stego-layers together in order to create the stego image.

Inputs: Stego-red, Stego-blue and Stego-green layers.

Process: Stego-layers that are carrying various data will be concatenated in order to produce the stego image.

Outputs: Stego image.

End

3.2 Extraction Algorithm

Figure 3 shows a block diagram for the proposed extraction process of hiding multiple types of information. Steps of this process can be summarized as follows.

Step 1: Select and read stego and cover images.

Inputs: Stego and cover images.

Process: The user will be asked to select stego and cover images in order to be read.

Outputs: Stego image and cover image are read.

End

Step 2: Separation of the layers of stego and cover images.

Inputs: Stego-image and cover image.

Process: Stego and cover image layers will be separated into three layers that are R, G and B.

Outputs: Separated layers of the stego and cover images.

End

Step 3: Correction of the layers of stego and cover images.

Inputs: Separated layers of the stego and cover images.

Process: Firstly, Haar Wavelet transform will be applied to the R-layer of the stego and cover images in order to figure out the horizontal, vertical and diagonal coefficients. Secondly, the blue layer of the stego image will be converted into binary vector. Finally, Both green layers of the stego and cover images will be reshaped to 1D.

Outputs: Layers of the stego and cover images are corrected.

End

Step 4: Retrieving process of three B&W secret images.

Inputs: Horizontal, vertical and diagonal coefficients of the R-planes.

Process: Compare the horizontal coefficient of the stego image with the equivalent coefficient of the cover image. Using the pseudo random number technique, the secret binary image concealed within this coefficient can be retrieved. Similar process will be applied to the rest of the coefficients in order to extract other secret images.

Outputs: Three B&W secret images.

End

Step 5: Extraction process of one secret colour image.

Inputs: The blue layer of the stego image in binary vector.

Process: Secret colour image can be extracted by the use of least significant bits (LSB) technique. Binary vector can be obtained by taking two bits every 6 bits of the blue layer of the stego image in binary vector. Then, this binary vector will be divided into three equivalent binary vectors. Finally, these three binary vectors will be converted into 2D and concatenated in order to figure out the secret colour image in 2D format.

Outputs: One secret colour image.

End

Step 6: Extraction process of large secret text file.

Inputs: Reshaped green layers of the stego and cover images.

Process: Implementing pseudo random number techniques, the reshaped stego and cover image pixels will be compared in order to figure out a binary vector. This binary vector will be converted into a string of characteristics and saved as 'secret text file.txt'.

Output: Large secret text file.

End

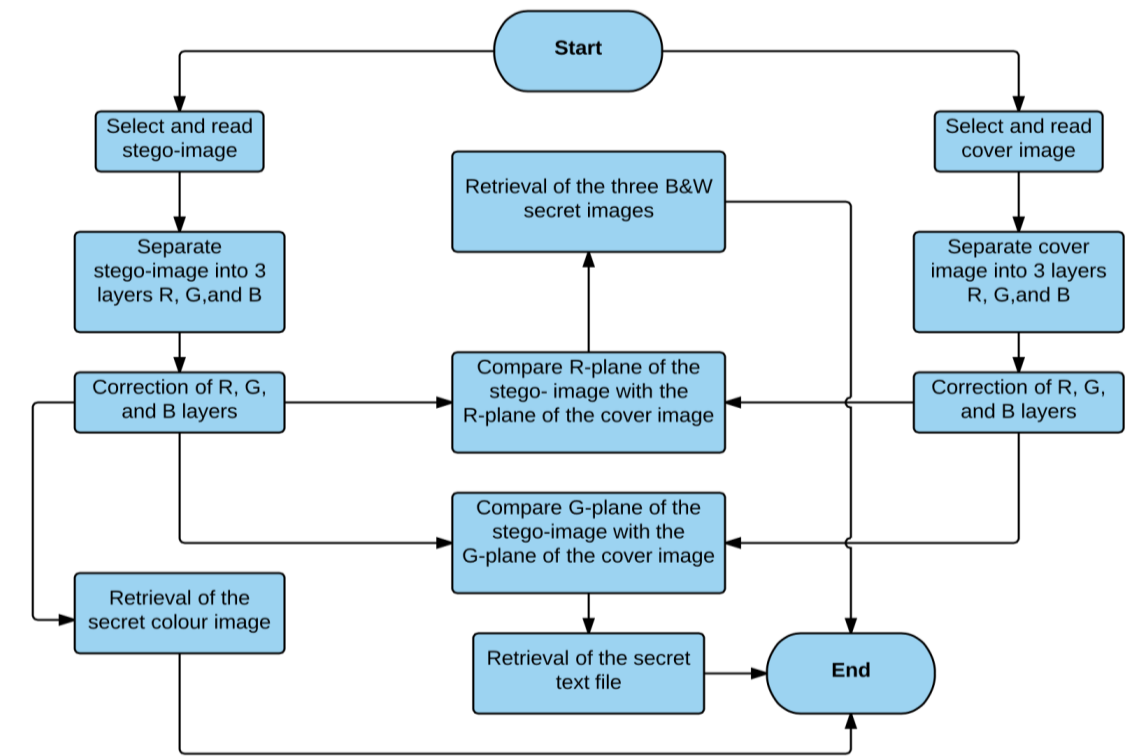


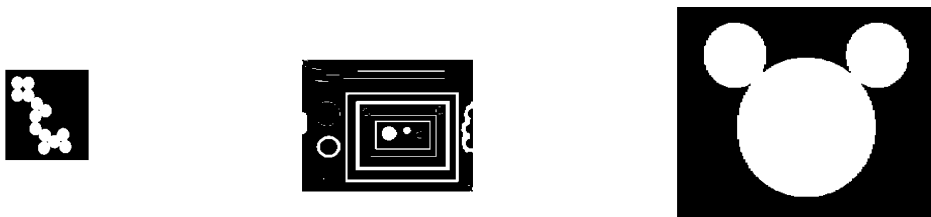
Figure 3. Decoding process of hiding multiple types of information [5].

4. DATA HIDING SCENARIOS

Scenarios of hiding different combinations of data types in a single cover image are presented and discussed in this section. These include hiding colour images, B&W images and larger text files or combinations of these data types in a single cover image.

4.1 Multiple B&W Images

In this scenario, the user can hide multiple B&W images as shown in Figure 4. The B&W images illustrated in Figure 4 (a) are embedded in a single cover image (Sailboat.tif), see Figure 4 (b). The obtained stego image is shown in Figure 4 (c). As illustrated, the stego image is indistinguishable from the original cover by the naked eye. Furthermore, the size of the stego image is identical to that of the original cover.



(a) Example of secret binary images

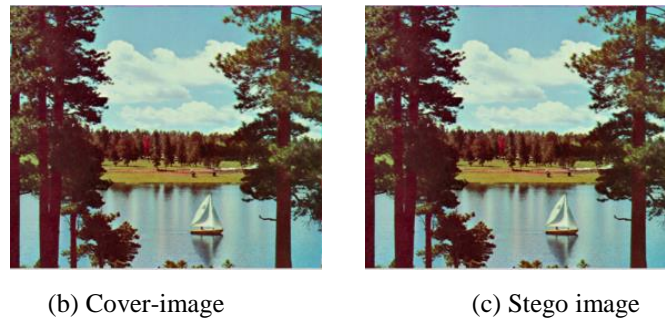


Figure 4. Hiding multiple B&W images in a single cover image (Sailboat.tif).

4.2 Multiple Coloured Images

In this scenario, the user can hide multiple coloured images as shown in Figure 5. The three coloured images of Figure 5 (a) are embedded in a single coloured image (Goldhill.bmp), see Figure 5 (b). In this case, two layers are used to hide the two images; each layer carries one secret colour image. As shown in Figure 5 (c), the resultant stego image is again indistinguishable from the original cover.



(a) Examples of secret coloured images



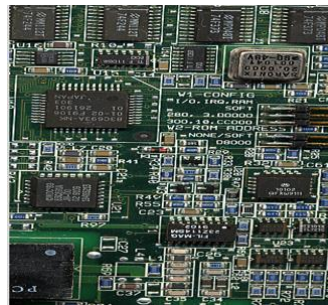
Figure 5. Hiding multiple coloured images in a single cover image (Goldhill.bmp).

4.3 Multiple Text Files

In this scenario, layers of the cover image are used to hide large text files. Figure 6 shows an example of hiding three copies of Shakespeare's *Tempest* (Figure 6 (a)) in a single cover image (board.tif) shown in Figure 6 (b). Each layer of the cover image carries one secret file. As illustrated, the appearance and size of the stego image of Figure 6 (c) are indistinguishable from those of the original image. It should be noted here that the size of each *Tempest*'s file comprises 107520 characters; hence more than 322000 characters can be concealed in a single cover image of a size of $512 \times 512 \times 3$ pixels.



(a) Secret text files (Shakespeare’s Tempest text) [26]



(b) Cover image

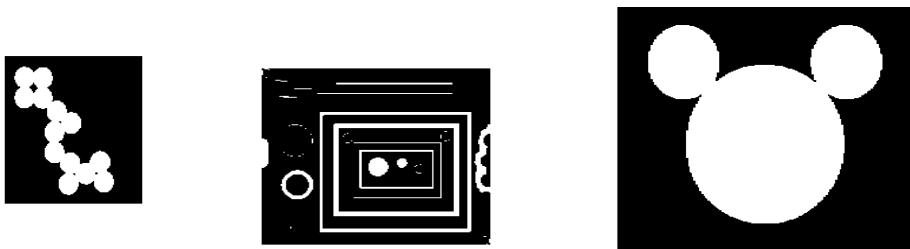


(c) Stego image

Figure 6. Hiding multiple text files in a single cover image (board.tif).

4.4 Miscellaneous Data Types

Figure 7 shows an example of hiding miscellaneous data types (i.e., B&W and coloured images and a large text file) in a single cover image (Lena.bmp). Figures 7 (d) and (e) depict the original cover and stego images, respectively. In this example, the size of the stego image is identical to that of the cover image, as illustrated.



(a) Example of secret binary images



(b) Secret coloured image



(c) Large text file



Figure 7. Hiding miscellaneous data in a single cover image (Lena.bmp).

5. PERFORMANCE EVALUATION

There are a various metrics that can be used in order to evaluate the performance of various steganography methods. In this paper, a number of performance evaluation metrics are used to evaluate the capacity (payload size) and imperceptibility of the cover image, as well as the robustness against various attacks. Measuring the PSNR and MSE assesses the imperceptibility, while measuring the correlation factor between the original and extracted secret images, after attacking the stego image by various noise attacks, assesses the robustness.

5.1 Capacity (Payload Size)

There is no specific definition for the capacity of the cover image. However, there are a various number of capacity expressions that can be used relying on different steganography approaches. In this paper, the capacity (C) has been defined as the amount of the cover image used for the embedding purpose [22].

$$C(\%) = \frac{\text{Pixels used for embedding purpose}}{J(i,j)} \times 100\% \quad (1)$$

where $J(i,j)$ represents the total rows and columns of the cover image (i.e., total number of the cover image pixels).

5.2 PSNR

Power signal-to-noise ratio is a measure of the difference between the original cover image and the stego image. It can be mathematically expressed as:

$$PSNR = 10 \times \log_{10} \left(\frac{n^2}{MSE} \right) \quad (2)$$

where n is the maximum pixel value for 8 bits.

5.3 MSE

Mean square error can be defined as the average square error between the cover image and the stego image. The mathematical expression for the MSE is given by:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^m \sum_{j=1}^n [J(i,j) - J'(i,j)]^2 \quad (3)$$

where $J(i,j)$ represents the cover image dimensions and $J'(i,j)$ represents the dimensions of the stego image.

6. RESULTS AND DISCUSSION

This section presents and discusses the results obtained from various experiments that were carried out to evaluate the imperceptibility of the stego image against different payload sizes for hiding various combinations of data types. It also presents some results on the robustness of the proposed algorithm against various kinds of attack.

6.1 Imperceptibility/Capacity Evaluation for Hiding Multiple B&W Images

Table 3 shows the performance evaluation when 3 B&W secret images are concealed within a single cover image. Various cover images with different formats such as .bmp and .tif have been used to demonstrate the efficiency of this algorithm. Moreover, there will be a small variation in the PSNR and the MSE, since the capacity is almost constant.

Table 3. Performance of hiding 3 logical images using various cover images.

Cover Image	PSNR (dB)	MSE	Capacity (%)
Lena.bmp	55.83	0.169	75
Goldhill.bmp	55.82	0.170	75
Sailboat.tif	55.78	0.171	75
Board.tif	55.96	0.164	75

6.2 Imperceptibility/Capacity Evaluation for Hiding Multiple Coloured Images

Table 4 shows the PSNR, MSE and capacity when hiding three coloured images. This algorithm has high capacity; hence the value of the PSNR will be decreased, while that of the MSE will be increased. Therefore, as the capacity increases, the PSNR and the MSE will be affected. Various image formats have been implemented and provided almost the same results as demonstrated.

Table 4. Performance of hiding 3 secret coloured images using various cover images.

Cover image	PSNR (dB)	MSE	Capacity (%)
Lena.bmp	43.70	0.925	99.56
Goldhill.bmp	43.90	0.882	99.56
Sailboat.tif	43.90	0.882	99.56
Board.tif	43.90	0.882	99.56

6.3 Imperceptibility/Capacity Evaluation for Hiding Multiple Text Files

Table 5 shows how the PSNR and the MSE are changing when the capacity increases. As the text entered by the user increases, the PSNR will be decreased, while the MSE will be increased. For example, when the number of bits is equal to 840; that is equal to 120 text letters, the PSNR and the MSE are equal to 80.74 dB and 0.00055, respectively. However, when the number of the embedded bits is equal to 752640, which is equivalent to 107520 text letters, PSNR and MSE are equal to 51.30 dB and 0.8233, respectively. Figures 8 and 9 illustrate clearly the relation between the capacity and PSNR, as well as the relation of the capacity and MSE, respectively.

From Figure 8, it can be seen that the PSNR drops with the increase of the payload (size of message to be hidden with respect to the total cover medium size). In fact, this is typical, but increasing the payload from about 100 Kbit to about 700 Kbit will result in a drop of the PSNR of about 10 dB; a drop that is basically not huge. This illustrates the effectiveness of the proposed algorithm. Same conclusion can be drawn from Figure 9, where the MSE increases (from about 0.05% to approximately 0.45%) when the payload goes up by the same amount mentioned above.

Table 5. Performance of hiding various sizes of secret text.

Cover Image	PSNR (dB)	MSE (%)	Embedded Text Size (Kbits)
Sailboat.tif	80.74	0.00055	0.840
Sailboat.tif	77.73	0.00109	1.680
Sailboat.tif	74.71	0.00219	3.360
Sailboat.tif	71.68	0.00442	6.720
Sailboat.tif	68.66	0.00885	13.440
Sailboat.tif	65.65	0.01769	26.880
Sailboat.tif	62.64	0.03539	53.760
Sailboat.tif	59.63	0.07079	107.520
Sailboat.tif	56.83	0.13495	215.040
Sailboat.tif	53.82	0.26992	430.080
Sailboat.tif	51.30	0.48233	752.640

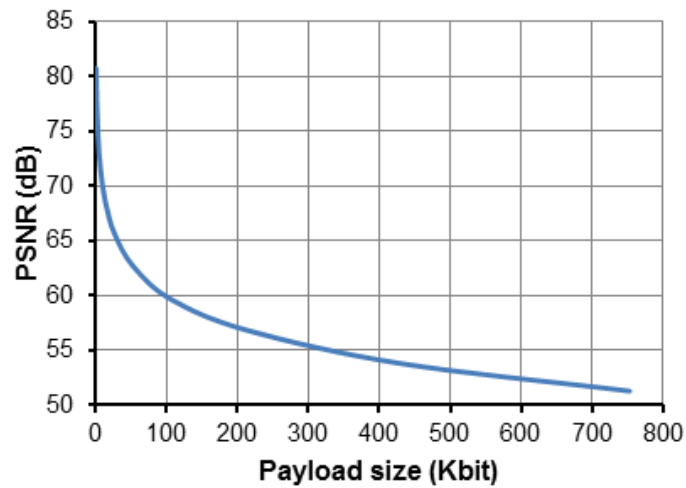


Figure 8. PSNR against payload size for hiding private text.

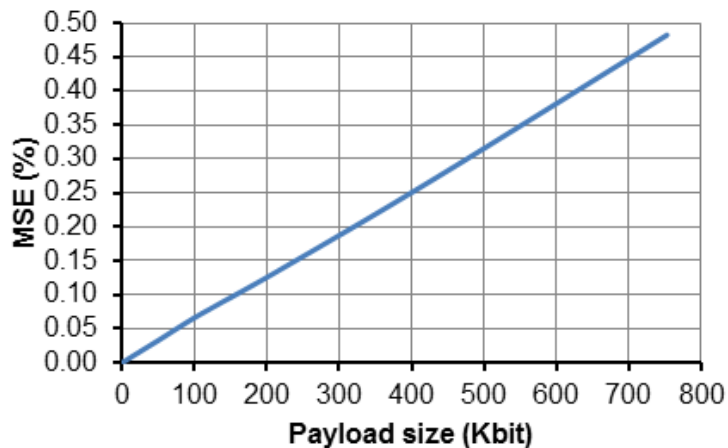


Figure 9. MSE against capacity for hiding private text.

6.4 Imperceptibility/Capacity Evaluation for Hiding Miscellaneous Data Types

Table 6 shows the values of the PSNR and the MSE when hiding multiple types of private data. It can be seen that as the capacity increases, MSE and PSNR will be affected. Figures 10 and 11 show how the capacity affects the PSNR and the MSE, respectively.

Table 6. Performance of hiding multiple private data.

Cover Image	PSNR (dB)	MSE (%)	Embedded Data Size (Kbits)
Sailboat.tif	48.180	0.32950	719.448
Sailboat.tif	48.178	0.32973	720.288
Sailboat.tif	48.173	0.33001	721.968
Sailboat.tif	48.163	0.33084	728.328
Sailboat.tif	48.144	0.33232	732.008
Sailboat.tif	48.106	0.33527	745.488
Sailboat.tif	48.010	0.34117	772.368
Sailboat.tif	47.881	0.35297	826.128
Sailboat.tif	47.625	0.37435	933.648
Sailboat.tif	47.524	0.38320	973.968

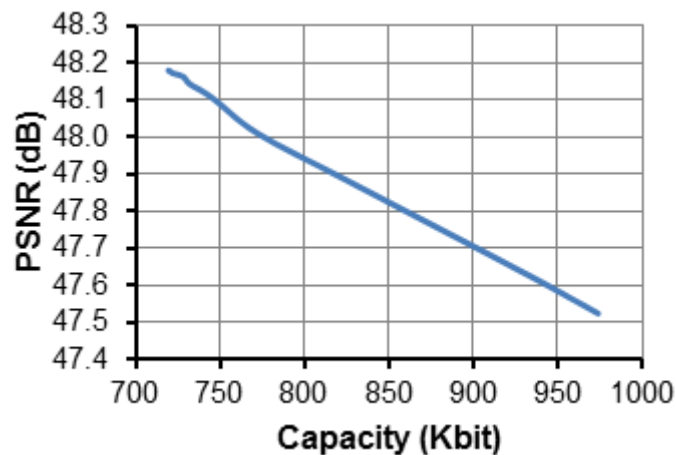


Figure 10. PSNR against capacity for hiding multiple types of private data.

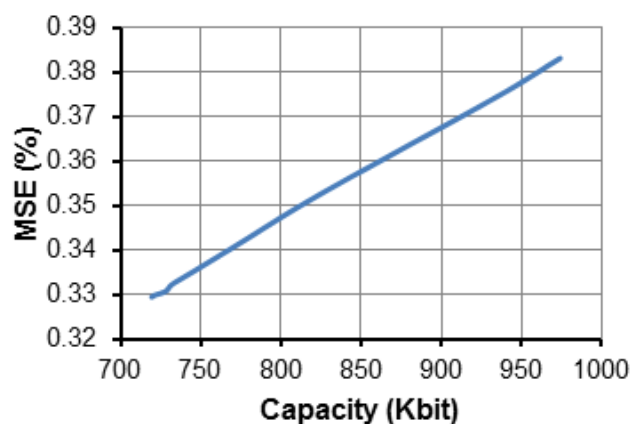


Figure 11. MSE against capacity for hiding multiple types of private data.

The proposed steganography techniques provide not only high capacity, but also high PSNR and low MSE. The size of the concealed secret text file (i.e., Shakespeare's *Tempest*) equals about 752640 bits which is equivalent to 107520 letters. Moreover, sizes of the stego image formed in all the proposed techniques were equal to the size of the cover image, which is 512×512×3 pixels. Capacity values are found to be high except for that of hiding B&W images,

since the approximation coefficients that represent the low-low frequencies of the Wavelet transform have not been used for the hiding process. However, as the capacity increases, PSNR decreases and MSE increases. Table 7 summarizes the obtained performance parameters for the developed algorithm.

Table 7. Performance summary.

Hidden Data	PSNR (dB)	MSE (%)	Capacity (%)
Multiple data	47.52	0.383	89.42
3 B&W images	55.78	0.171	75
3 Colour images	43.93	0.877	99.56
Text file	51.43	0.467	95.70

Embedding three binary secret images within a single cover image by the use of Wavelet transform has been proposed and successfully implemented. The PSNR, MSE and capacity achieved in this technique were found to be equal to 55.78 dB, 0.171 and 75%, respectively. Moreover, another technique has also been proposed, where the user will have the ability to hide a single secret colour image within one cover image. PSNR, MSE and capacity are equal to 43.93 dB, 0.877 and 99.56%, respectively. Another technique to hide large secret text files was also proposed. The capacity in this technique can vary depending on the size of the secret text file. However, the results of hiding large pdf files have been presented. Furthermore, a method to conceal three binary images, one secret colour image and one large text file has been suggested in this paper. Figures 10 and 11 illustrate the effect of the payload size on the PSNR and the MSE, respectively.

6.5 Robustness Evaluation for Hiding Miscellaneous Data Types

Robustness of the proposed steganography algorithm is tested through embedding miscellaneous secret data (i.e., three B&W images, a coloured image and a text file) in a cover (Lina image). The obtained stego image shown in Figure 12 is then exposed to Gaussian with white noise, Poisson noise, and salt and pepper noise. The attacked stego images and the miscellaneous secret data retrieved from each of the attacked images are summarized in Table 8.

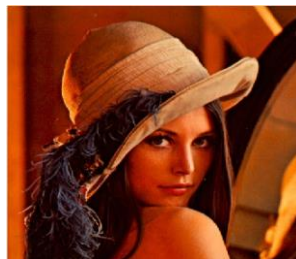



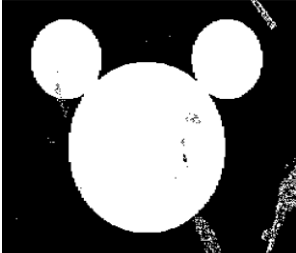
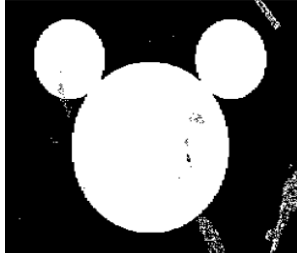






Figure 12. Original stego image.

The obtained results clearly demonstrate that the robustness of the developed steganography algorithm varies depending on the type of the embedded data and the type of attack. For example, it is shown that unlike B&W images, colour images and texts are not affected by the attacks used in this study. Furthermore, B&W images are only slightly distorted by the Gaussian with white noise and Poisson noise attacks, but are found fragile against salt and pepper noise attacks. The correlation between the original and the extracted secret images is measured and presented in Table 9. It should be mentioned here that the correlation factor measurement that is applied to evaluate robustness of the secret images is not applicable to secret text files.

Table 8. Summary of robustness tests against various attacks

	Gaussian with white noise	Poisson noise	Salt and pepper noise
Attacked stego image			
Retrieved secret B&W image			
Retrieved secret coloured image			
Retrieved secret B&W text	<p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p>	<p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p>	<p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p> <p>William Shakespeare (1564 - 1616) was born at Stratford-upon-Avon in a house in Henley Street. This is preserved intact.</p>

The nature of salt and pepper noise is to affect the B&W pixels; hence the binary images are highly affected by this kind of noise. In addition, the adopted pseudo random numbers that are dependent on the B&W pixels also affect concealing B&W images. Partial or total distortion is therefore expected on the canceled B&W images when exposed to salt and pepper noise due to the fact that pixels of such images have binary values that are easily corrupted with a probability value of 50%. However, the effect of this noise can be reduced by the use of median or morphological filters. Nevertheless, as discussed earlier in the introduction, robustness in steganography is less important than in watermarking, since the attacker is mainly concerned with the discovery of hidden data rather than with removing or modifying them.

Table 9. Correlation factor measurement for various types of attacks.

	Gaussian with white noise	Poisson noise	Salt and pepper noise
Colour image	1	1	1
B&W image (blobs.png)	1	1	0.20
B&W image (Circles.png)	1	1	0.25
B&W image (Binary_circles.png)	1	0.997	0.06
B&W text	N/A to secret texts		

7. CONCLUSION

In this paper, a high-capacity image steganography algorithm based on Haar wavelet transform that is capable of hiding various data types (i.e., B&W and coloured images, as well as text files) has been presented. All these types of private data are concealed in stego images of a unified size of $512 \times 512 \times 3$ pixels. The stego image is formed to be always equal to the cover image. Experimental evaluation has proven that the proposed steganography is highly efficient in terms of capacity size of the cover image while maintaining a relatively low MSE and high PSNR and is reasonably robust against external attacks. The provided results have confirmed this conclusion.

The developed algorithms can also be further improved by the use of Huffman coding in order to have more area for hiding extra data; hence increasing the capacity. Moreover, robustness, which can be defined as how long the stego carrier can withstand before an eavesdropper can extract the concealed data, is another area for future improvement. For example, the binary vector of the secret image can be divided into a number of small blocks where each block can then be concealed randomly within the cover image. Thus, even if eavesdroppers discover the stego image, they will not have the ability to assemble the hidden blocks of the secret image. Cryptography can also be added to improve security through allocating unused pixels for cryptography bits. These potential improvements among others are currently part of the ongoing research of the authors. Also, measures for image quality assessment like structural similarity index (SSI) or structural similarity index mean (SSIM) can be used to assess the quality of data embedding.

Finally, steganography is an open area for further research, as many algorithms can still be proposed and practically implemented. However, it is worth mentioning that security, robustness and payload (capacity) will always conflict with each other. A fourth factor that can be added here is the performance or algorithm execution time that can add more challenge in this research area.

REFERENCES

- [1] H. V. Desai, "Steganography, Cryptography, Watermarking: A Comparative Study," *Journal of Global Research in Computer Science*, vol. 3, no. 12, pp. 33-35, 2012.
- [2] M. A. Al-Tae, N. H. Al-Hassani, B. S. Bamajbour and D. Al-Jumeily, "Biometric-Based Security System for Plaintext E-mail Messages," in: *Proc. International Conference on Developments in eSystems Engineering*, Abu Dhabi, UAE, , pp. 1-6, 14 – 16 December 2009.
- [3] N. Qasrawi, M. A. Al-Tae, H. l'emair and R. Al-Asa'd, "Multilevel Encryption of Plaintext Messages Using a Smart Card Connected to PC Parallel Port," in: *Proc. 3rd International*

- Conference on Modelling, Simulation and Applied Optimization, Sharjah-UAE, , pp. 1-6, 20-22 January 2009.
- [4] S. Katzenbeisser and F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Inc, 2000.
- [5] H. A. Al-Korbi, A. Al-Ataby, M. A. Al-Tae and W. Al-Nuaimy, "High-Capacity Image Steganography Based on Haar DWT for Hiding Miscellaneous Data," in: Proc. IEEE/AEECT'2015 Jordan Conference on Applied Electrical Engineering and Computing Technologies, Amman, Jordan, pp. 1-6, 3-5 November 2015.
- [6] S. Jayasudha, "Integer Wavelet Transform Based Steganography Method Using Opa Algorithm," *International Journal of Engineering and Science*, vol. 2, no. 4, pp. 31–35, 2013.
- [7] A. Al-Ataby and F. M. Al-Naima, "A Modified High Capacity Image Steganography Technique Based on Wavelet Transform," *International Arab Journal of Information Technology*, vol. 7, no. 4, pp. 358–364, 2010.
- [8] S. Banerjee, S. Chakraborty, N. Dey, A. K. Pal and R. Ray, "High Payload Watermarking Using Residue Number System," *International Journal of Image, Graphics and Signal Processing*, vol. 3, pp. 1-8, 2015.
- [9] M. S. Al-Yaman, M. A. Al-Tae and H. Alshammas, "Audio-Watermarking Based Ownership Verification System Using Enhanced DWT-SVD Technique," in: Proc. IEEE/SSD2012 Multi-Conference on Systems, Signals and Devices, Chemnitz-Germany, pp. 1-5, 20-23 March 2012.
- [10] M. S. Al-Yaman, M. A. Al-Tae, A. T. Shahrour and I. A. Al-Husseini, "Biometric Based Audio Ownership Verification Using Discrete Wavelet Transform and SVD Techniques," in: Proc. IEEE/SSD2011 Multi-Conference on Systems, Signals and Devices, Tunisia, pp. 1-5, 22-25 March 2011.
- [11] N. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Computer*, vol. 31, pp. 26–34, 1998.
- [12] D. Wu and W. Tsai, "A Steganography Method for Images by Pixel-value Differencing," *Pattern Recognition Letters*, vol. 24, pp. 1613–1626, 2002.
- [13] A. Hashad, A. Madani and A. Wahdan, "A Robust Steganography Technique Using Discrete Cosine Transform Insertion," in: Proc. Int. Conf. on Information and Communications Technology, Cairo, Egypt, pp. 255–264, 5-6 December 2005.
- [14] P. Chen and H. Lin, "A DWT Approach for Image Steganography," *International Journal of Applied Science and Engineering*, vol. 4, no. 3, pp. 275–290, 2006.
- [15] H. S. Reddy and K. B. Raja, "High Capacity and Security Steganography Using Discrete Wavelet Transform," *International Journal of Computer Science and Security*, vol. 3, no. 6, pp. 462-472, 2010.
- [16] N. Dey, A. Roy and S. Dey, "A Novel Approach of Colour Image Hiding Using RGB Colour Planes and DWT," *International Journal of Computer Applications*, vol. 36, no. 5, pp. 19–24, 2011.
- [17] T. Bhattacharya, N. Dey and S. Chaudhuri, "A Session Based Multiple Image Hiding Technique Using DWT and DCT," *International Journal of Computer Applications*, vol. 38, no.5, pp. 18–21, 2012.
- [18] E. Ghasemi, J. Shanbehzadeh and N. Fassihi, "High Capacity Image Steganography Using Wavelet Transform and Genetic Algorithm," in: Proc. Int. Multi-Conference of Engineering and Computer Scientists (IMECS), vol. 1, Hong Kong, pp. 1–4, 16-18 March 2011.
- [19] S. Laskar and K. Hemachandran, "High Capacity Data Hiding Using LSB Steganography and Encryption," *International Journal of Database Management Systems*, vol. 4, no. 6, pp. 57–68, 2012.
- [20] P. Ganesan and P. Bhavani, "A High Secure and Robust Image Steganography Using Dual Wavelet and Blending Model," *Journal of Computer Science*, vol. 9, no. 3, pp. 277–284, 2013.

- [21] M. Parul and R. Harish, "Optimized Image Steganography Using Discrete Wavelet Transform (DWT)," International Journal of Recent Development in Engineering and Technology, vol. 2, no. 2, pp. 75–81, 2014.
- [22] A. Nag, S. Biswas, D. Sarkar and P. Sarkar, "A Novel Technique for Image Steganography Based on DWT and Huffman Encoding," International Journal of Computer Science and Security, vol. 4, no. 6, pp. 561–570, 2011.
- [23] I. Badescu and C. Dumitrescu, "Steganography in Image Using Discrete Wavelet Transformation," in: Proc. WSEAS Conf. on Advances in Mathematical Models and Production Systems in Engineering, Brasov, Romania, pp. 69-72, 26-28 June 2014.
- [24] M. Sifuzzaman, M. Islam and M. Z. Ali, "Application of Wavelet Transform and Its Advantages Compared to Fourier Transform," Journal of Physical Science, vol. 13, pp. 121–134, 2009.
- [25] L. Jing, K. Zhi-wei and H. Yi-gang, "A Steganography Method Based on Wavelet Contrast and LSB," Chinese Journal of Electronics, vol. 35, pp. 1391–1393, 2007.
- [26] W. Shakespeare, "The Tempest," Online: <http://sparks.eserver.org/books/shakespeare-tempest.pdf>, last accessed 15 January 2016.

ملخص البحث:

أصبحت حماية البيانات الخاصة التي يجري تبادلها على الإنترنت ومن يمكنه الوصول إلى تلك البيانات أمراً بالغ الأهمية والضرورة لما ينطوي عليه من مسائل تتعلق بالخصوصية والسرية. ويساعد إخفاء الصور أو أجزاء منها في تحصين البيانات الخاصة داخل صورة غلاف للحصول على غلاف جديد لا يمكن عملياً تمييزه عن الغلاف الأصلي، بطريقة تمنع الأشخاص غير المخولين من كشف البيانات المحصنة في الغلاف الجديد. لذا؛ فإن سعة صورة الغلاف وانعدام قابليتها للإدراك يُعدّان من المتطلبات الحاسمة لتقويم أداء خوارزميات الإخفاء.

تقدم هذه الورقة خوارزمية إخفاء عالية الفعالية لها القدرة على إخفاء حجم كبير من البيانات المتفرقة؛ مثل ملفات النصوص، والصور الثنائية، والصور الملونة أو تركيبية من هذه الأنواع من البيانات، في صورة غلاف واحدة باستخدام تحويل "هار" للموجات. وقد تم عرض تفاصيل خوارزميات الإخفاء والاستخراج للأنواع المختلفة من البيانات ومناقشتها. وتم تقويم أداء طريقة الإخفاء المقترحة من حيث سعة صورة الغلاف، وانعدام قابلية الإدراك، والمتانة. وبيّنت النتائج والملاحظات التي تم الحصول عليها أنّ الخوارزميات التي جرى تطويرها ذات فعالية عالية من حيث سعة صورة الغلاف، مع الحفاظ على قيمة منخفضة نسبياً للخطأ التريبيعي المتوسط وقيمة عالية لأعلى نسبة للإشارة إلى الضجيج، إضافة إلى متانة معقولة ضد الهجمات المختلفة.

CHARACTERIZATION OF SHARED-MEMORY MULTI-CORE APPLICATIONS

Mohammed Sultan Mohammed¹ and Gheith A. Abandah²

Computer Engineering Department, the University of Jordan, Amman, Jordan

m.s.mohammed@ieee.org¹, abandah@ju.edu.jo²

(Received: 29-Nov.-2015, Revised: 21-Jan.-2016, Accepted: 01-Feb.-2016)

ABSTRACT

The multicore processor architectures have been gaining increasing popularity in the recent years. However, many available applications cannot take full advantage of these architectures. Therefore, many researchers have developed several characterization techniques to help programmers understand the behavior of these applications on multicore platforms and to tune them for better efficiency. This paper proposes an on-the-fly, configuration-independent characterization approach for characterizing the inherent characteristics of multicore applications. This approach is fast, because it does not depend on the details of any specific machine configuration and does not require repeating the characterization for every target configuration. It just keeps track of memory accesses and the cores that perform these accesses through piping memory traces, on-the-fly, to the analysis tool. We applied this approach to characterize eight applications drawn from SPLASH-2 and PARSEC benchmark suites. This paper presents the inherent characteristics of these applications, including memory access instructions, communication characteristics patterns, sharing degree, invalidation degree, communication slack and communication locality. The results show that two of the studied applications have high parallelization overhead, which are Cholesky and Fluidanimate. The results also indicate that the studied applications of SPLASH-2 have higher communication rates than the studied applications of PARSEC and these rates generally increase as the number of used threads increases. Most of the sharing and invalidation occurs in small degrees. However, two of SPLASH-2 applications have significant fraction of communication with high sharing degrees involving four or more threads. Most of the applications have some uniform communication component and the initial thread is generally involved in more communication compared to the other threads.

KEYWORDS

Multi-core processor, On-the-fly analysis, Shared memory applications, Communication patterns, Performance evaluation.

1. INTRODUCTION

The multicore architecture is the current and the foreseeable future approach that processor manufacturers use to build high-performance and low-power processors [1]-[2]. Most of the current processors are multicore processors; i.e., there are multiple processors on the processor chip. This approach is also the preferred approach in mobile devices [3]. Moreover, the number of cores on one chip increases with time [4]. To take advantage of the increasing number of cores, many parallel programming approaches were developed. One of these approaches is multithreading. Using parallel-multithreaded approach, various numbers of threads of one application can be concurrently executed on multiple cores and shared memory, thus facilitating implementing the algorithms that solve data-intensive problems such as searching and sorting [5]-[6]. Also, there are many applications that are developed to run on multicore systems, including some popular benchmarks. Nevertheless, there are many aspects that need to be tackled to improve multicore performance. Characterizing the benchmarks that represent

multicore applications on multicore systems is important to tune such applications for better performance and to design better multicore systems.

This paper proposes an on-the-fly configuration independent characterization approach for characterizing the inherent characteristics of multicore applications. This approach is carried out to characterize eight representative applications drawn from the popular SPLASH-2 and PARSEC application benchmarks. The proposed approach characterizes the inherent characteristics that are independent from any particular multicore configuration. The inherent characteristics are divided into two parts. The first part is the characteristics of the memory access instructions, which include the numbers of memory accesses and the percentages of memory accesses by type of access and access data size. This characterization is useful to find the amount of parallelization overhead. The second part is the communication characteristics, which include communication patterns, sharing degree, invalidation degree, communication slack and communication locality. It is important to characterize the communication characteristics of the multicore applications, as high communication overhead is often responsible of bad parallelization efficiency [7].

The rest of this paper is organized as follows: Section 2 presents a short survey of some related work. Section 3 summarizes our methodology in characterizing multicore applications and the experimental setup used. Section 4 presents the characterization results. Finally, Section 5 presents some conclusions and future work.

2. RELATED WORK

Several studies have proposed different techniques to characterize parallel applications. These techniques are summarized in the following four categories.

2.1 Hardware-Assisted Characterization

Many characterization studies have used hardware performance counters, which are special registers on the processor that count hardware events to characterize various aspects of running applications.

Dongarra *et al.* used these counters to characterize data cache and translation lookaside buffer (TLB) behaviors of their microbenchmarks [8]. Bhadauria *et al.* characterized PARSEC on multiple aspects, including cache performance, sensitivity to DRAM speed and bandwidth, multithread scalability and micro-architecture design choices on a variety of real multicore systems [9]. Ferdman *et al.* used these counters to study the micro-architectural behavior of their CloudSuite benchmarks [10]. They concluded that existing processor micro-architectures are inefficient for running their benchmarks. Jia *et al.* also used these counters to characterize eleven data analysis workloads of a data center to determine their micro-architectural characteristics on systems equipped with modern superscalar, out-of-order processors [11]. They also developed a benchmark suite called DCBench to mimic typical data center workloads.

2.2 Message-Passing Characterization

Instrumented message-passing libraries are often used to characterize parallel applications running on multi-computer systems.

Cohen and Mahafzah proposed a utility to characterize NAS benchmarks, which are a group of programs developed by NASA Ames to help evaluate the performance of parallel supercomputers. Their results provide a deep look at how NAS benchmarks work on parallel computers [12]. Alam *et al.* characterized the scaling behavior of a set of micro-benchmarks, kernels and scientific workloads on HPC systems [13]. They used AMD Opteron multicore

processors and concluded that the Opteron cache coherence protocol is insufficient to exploit the full bandwidth capability of the memory interface. Chai *et al.* characterized micro-benchmarks and application-level benchmarks on an Intel dual-core cluster [14]. They suggested that the communication middleware and applications should be multicore to optimize intra-node and inter-node communication.

2.3 Configuration Dependent Analysis

Configuration dependent characterization techniques characterize applications depending on simulating the application execution on a specific system configuration. This approach is widely used in characterizing applications.

Abandah developed a configuration dependent analysis tool (CDAT) to characterize shared memory behavior, including cache misses and sharing that depend on system configuration parameters such as cache block size [15]. CDAT is a multiprocessor system simulator that has memory, cache, bus and interconnection models. By using a configuration file, users can specify the system configuration, including the coherence protocol, size and speed of the system components, as well as processors and memory banks interconnections. Jaleel *et al.* used a dynamic binary instrumentation tool as an alternative to the trace-driven and execute-driven approaches [16]. They proposed a memory system simulator to characterize memory performance of x86 workloads on multicore systems. Contreras and Martonosi characterized a subset of PARSEC benchmark applications that were compiled with Intel TBB on AMD dual-core processors in order to determine the sources of overhead within the TBB [17]. Bienia *et al.* characterized PARSEC applications and found that they have various types of multithreaded behaviors. Bhattacharjee and Martonosi characterized TLB behavior of the PARSEC [18]. Dey *et al.* also characterized PARSEC and measured the effect of shared resource contention on performance [19]. They classified resource contention into intra-application contention, which is the contention among threads from the same application, and inter-application contention, which is the contention among threads from different applications. Natarajan and Chaudhuri characterized a set of multithreaded applications selected from PARSEC, SPEC OMP and SPLASH-2 to understand last-level cache (LLC) behavior of multithreaded applications [20]. They proposed a generic design that introduces sharing-awareness in LLC replacement policies. They showed that their design could significantly improve the performance of LLC replacement policies.

2.4 Configuration Independent Analysis

The configuration independent characterization technique is a unique technique for characterizing the inherent application characteristics that do not change when changing the system configuration.

Abandah and Davidson developed a configuration independent analysis tool (CIAT) to characterize the configuration independent characteristics, such as memory access instructions, concurrency, communication patterns and sharing behavior of shared-memory applications that run on multiprocessor systems [21]. CIAT analyzes the memory traces of shared-memory applications to find these characteristics. It is faster than detailed simulators, as it only keeps track of accesses to each memory location and does not include detailed models of a specific system's components and protocols [15]. Moreover, its characterization is general; it gives the inherent characteristics of the application that do not depend on a machine configuration. Thus, CIAT gives us a more general understanding of the application behavior.

This work ported CIAT, which was originally developed for RISC multiprocessor systems, to commodity multicore systems. The ported tool was used to characterize the inherent characteristics of representative multicore applications.

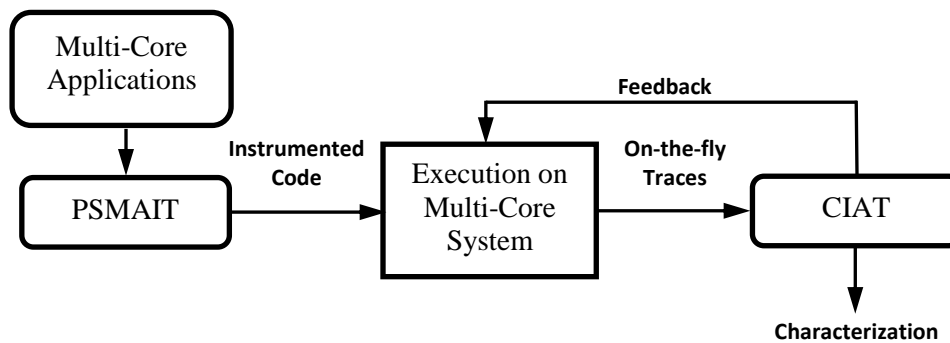


Figure 1. Methodology used to characterize multi-core applications.

3. CHARACTERIZATION METHODOLOGY

This section summarizes the methodology used in this study for monitoring and characterizing multicore applications and describes the tools developed to characterize these applications.

This methodology involves generating detailed memory traces and sending these traces on-the-fly, as the application is being executed, to the analysis tool, as shown in Figure 1. The Pin shared-memory application instrumentation tool (PSMAIT) is used to instrument the multicore application and pipe traces to the ported CIAT that analyzes these traces.

The following sub-sections present more details about the developed tools and the applications being characterized.

3.1 Instrumentation Tool (PSMAIT)

PSMAIT is a tool based on Pin [22], a dynamic binary instrumentation tool for Linux and Windows. Pin is a just in time (JIT)-based dynamic instrumentation tool. It uses dynamic compilation techniques to instrument applications while they are running. Pin instruments single and multithreaded applications and supports Intel IA-32 and x86-64 instruction-set architectures [23]. It has a rich set of API's that can be used to instrument applications without the need to master the underlying instruction set.

PSMAIT is a tool written in C++. It consists of a set of instrumentation and analysis routines as shown in Figure 2. The instrumentation routines determine where instrumentation is inserted and the analysis routines determine what to do when instrumentation is activated. PSMAIT is designed to collect traces of multithreaded parallel applications and to send these traces directly, on-the-fly, to CIAT. PSMAIT is a run-time binary instrumentation tool, which means that it does not need the source code of the parallel application. It instruments both the parallel application's user code and all the user-level libraries that are called during the application execution.

Figure 2 shows the implementation overview of PSMAIT. PSMAIT uses Pin instrumentation routine to capture memory accesses that are performed by user code and user-level libraries only; it does not measure operating system events. Subsequently, it uses `MemRead` or `MemWrite` analysis routine, depending on memory access type, to send a simple trace record to CIAT for every memory access. This trace record contains the type of the access (load or store, integer or floating point), its size and the starting virtual address of the memory location accessed. PSMAIT sends these memory access records to CIAT on-the-fly by using pipes and waits for receiving confirmation feedback from CIAT. On-the-fly analysis enables analyzing large problems fast without needing huge storage medium.

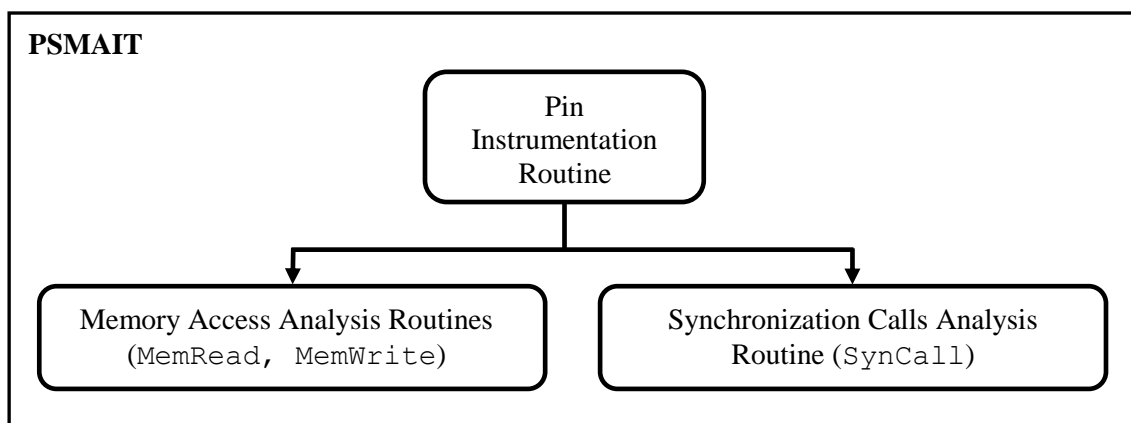


Figure 2. PSMAIT implementation overview.

Additionally, PSMAIT uses Pin instrumentation routine to capture the synchronization calls such as thread spawn and join, mutex lock and unlock, barrier and conditional wait and uses SynCall analysis routine to pipe their trace records to CIAT. Through these records and the response feedback from CIAT, the two tools control the parallel application execution and avoid any non-deterministic behavior of the instrumented application due to the instrumentation overhead.

3.2 Analysis Tool (CIAT)

Our analysis tool is ported from CIAT that was developed for the RISC multiprocessor systems by Abandah [15]. CIAT characterizes the inherent application characteristics, such as memory access instructions, communication patterns and sharing behavior of parallel applications that are independent from one multicore configuration to another. A multicore configuration includes the hierarchy of cores, the interconnection topology, the coherence protocol, the cache configuration, as well as the sizes and speeds of the multicore system components. CIAT does not characterize the application characteristics that depend on configuration parameters, such as cache misses and false sharing. However, CIAT's characterization of the communication characteristics gives a basic understanding of applications execution and helps explain the dependent characteristics such as cache misses; e.g. high number of RAW accesses means that load misses are higher than store misses.

CIAT uses many variables to count the various events by tracking the memory load and store operations. It accepts traces from PSMAIT, which generates n trace pipes for the n executing threads. CIAT supports various execution phases; it assumes that the traces come from a parallel application either in a serial or in a parallel phase. In a serial phase, there is only one thread active, while the other threads are idle. In a parallel phase, more than one thread can be active. CIAT uses the special records of the thread spawn and thread join calls to identify switches between serial and parallel phases. At the end of each phase, CIAT generates statistics and saves them in a report file. At the end of the last phase, CIAT reports the aggregate statistics in the report file.

CIAT assumes that n cores in multicore processor can execute n instructions at the same time and each instruction takes a fixed time. Therefore, a *pseudo clock* in instruction units is used to keep track of the execution time. However, CIAT currently only sees the memory accesses and advances the clock by one for each thread whenever it receives a memory access record. This is an approximation of the instruction stream. CIAT interleaves the analysis of multiple thread traces on the processors according to the thread spawn and join calls and follows the constraints

of the lock, conditional wait and barrier synchronization calls. More details about CIAT are found in Ref. [24].

3.3 Case Study Applications

We have chosen a set of parallel applications that are representative of multicore applications and are widely used in recent multicore research. This set consists of eight applications from two benchmark suits. The first four of these applications are from SPLASH-2 suite [25], which are Radix, FFT, LU and Cholesky. The second four are from PARSEC suite [26], which are Canneal, Blackscholes, Fluidanimate and Swaptions. These eight particular applications are selected, because they represent a wide range of applications and are often used in multicore research. More details about these selected applications are found in Ref. [24].

To study the impact of the application problem size on the communication behavior, we use two problem sizes of each application: Size I and Size II, where the problem solved in Size I is smaller than that of Size II. Table 1 shows these problem sizes and the abbreviations that are used for naming these applications.

We have conducted many analysis experiments of these studied applications with various numbers of threads for the two problem sizes. To validate our results, we repeated these experiments on two machines that have different types of multicore processors. The first machine has dual-core Core i5 2520M processor (3-MB Cache, 3.20 GHz) and the second has quad-core Core i5 2400 processor (6-MB Cache, 3.40 GHz). The characterization results on the two machines are identical. Therefore, this is one validation check that our characterization tools do not depend on the hardware configuration. Moreover, the number of instructions that have been obtained from running the applications with Pin is similar to that obtained by the developed tools.

Table 1. The applications' problem sizes.

Suite	Application	Abbreviation	Size I	Size II
SPLASH-2	Radix	Radix	256K integers	2M integers
	FFT	FFT	64K points	1M points
	LU	LU	256×256	512×512
	Cholesky	Chole	tk15.0 file	tk29.0 file
PARSEC	Canneal	Cann	simsmall	simmedium
	Blackscholes	Black	simsmall	simmedium
	Fluidanimate	Fluid	simsmall	simmedium
	Swaptions	Swap	simsmall	simmedium

4. CHARACTERIZATION RESULTS AND EVALUATION

This section presents the results of the inherent characteristics of the multicore applications that are measured and reported by CIAT. Due to paper length limitations, we present here the results of Size II; interested readers can find the results of Size I in Ref. [24].

4.1 Memory Access Instructions

As mentioned in the previous section, the developed tools capture the user code and user-level libraries operations on the memory and report the number of load and store operations as shown in Table 2.

Table 2 shows the number of memory accesses in billions for the eight studied applications when using one thread. These numbers represent the number of operations on memory and not the memory access instructions, where some memory access instructions may do more than one operation on the memory.

Table 2. The counts and percentages of load and store operations for one thread.

	Radix	FFT	LU	Chole	Cann	Black	Fluid	Swap
No. of Loads (in 10⁹)	0.285 (66.7%)	0.190 (57.1%)	0.109 (68.2%)	0.381 (77.6%)	3.577 (57.7%)	0.213 (61.1%)	1.146 (81.9%)	2.246 (75.2%)
No. of Stores (in 10⁹)	0.143 (33.3%)	0.144 (42.9%)	0.051 (31.8%)	0.110 (22.4%)	2.625 (42.3%)	0.136 (38.9%)	0.253 (18.1%)	0.742 (24.8%)

In all applications, loads are more frequent. The load operations ratio is about twice the store operations in most of the studied applications. Some applications, such as Cholesky, Fluidanimate and Sawptions have even larger percentages of load operations. Cholesky has about four times more of load operations than store operations, because it operates on sparse matrices and needs to find the indices of non-zero elements in these matrices. Fluidanimate has about five times more load operations than store operations. Sawptions has about three times more load operations than store operations. The load and store operations in Canneal are relatively equi-frequent.

Figure 3 shows the percentage of memory accesses for running the eight applications with various numbers of threads. The percentages are normalized to the number of memory accesses when running the respective applications with single thread. Thus, we can notice the parallelization overhead. As obvious, the parallelization overhead is negligible in most of the studied applications. However, there are two of the eight studied applications that have a high percentage of parallelization overhead, which are Cholesky from SPLASH-2 and Fluidanimate from PARSEC. Cholesky has about 50% of memory accesses as overhead when running 16 threads. This overhead is due to Cholesky's work on sparse matrices, which have a larger communication to computation ratio. Fluidanimate has about 33% of memory accesses as overhead when running 16 threads. This overhead is due to Fluidanimate's partitioning of the work among the threads and each thread handles its portion and interacts with other threads to handle the shared data.

Figure 4 shows the percentage of the byte, half-word (2 bytes), word (4 bytes), double-word (8 bytes), float (single-precision floating-point) and double-float (double-precision floating-point) load and store operations when running 16 threads. All the studied applications do not have any quad-word (16 bytes) or extended-float (extended-precision floating-point) memory accesses. All the studied applications are scientific benchmarks, which have a large percentage of floating-point operations except Radix and Canneal, which are integer kernel applications. The percentages of byte and half-word accessed data is insignificant in almost all the studied applications except Radix and FFT that have 25% and 6% half-word load and store operations, respectively. These relatively large percentages are because they have a large portion of integer computation.

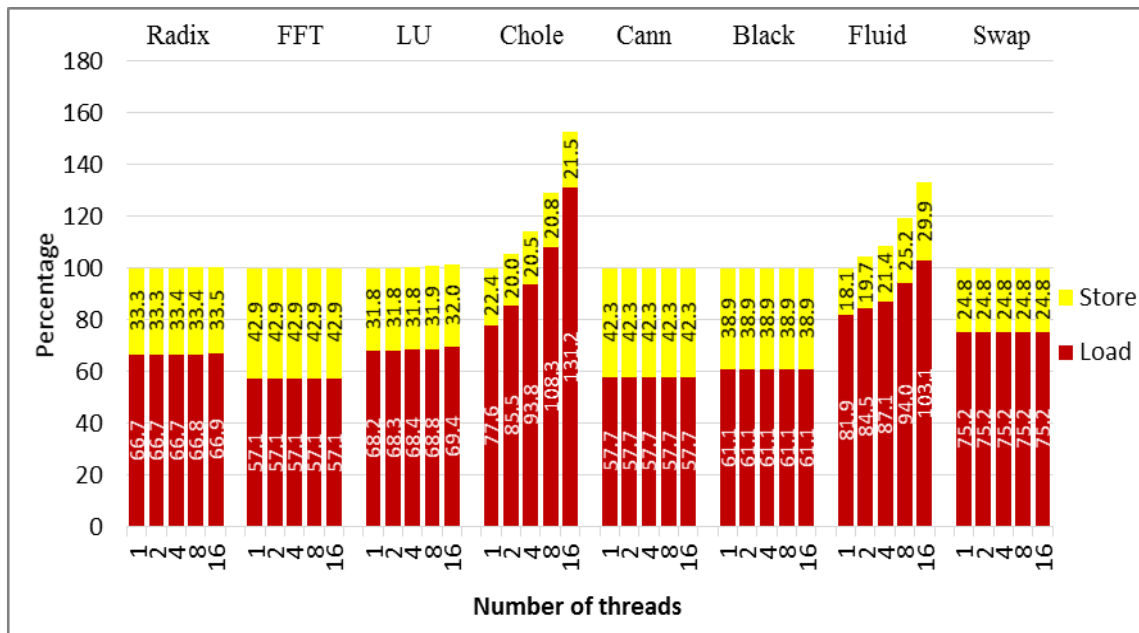


Figure 3. Percentage of memory accesses for 1-16 threads normalized to the memory accesses of one thread.

4.2 Communication Characteristics

This sub-section presents the inherent communication characteristics that are reported by CIAT.

4.2.1 Communication Patterns

The communication among the cores of a multicore processor occurs when those cores access same-shared memory locations. Characterizing the *communication patterns* is important to know which of the communication patterns are common, thus, facilitating the design of system that supports these patterns efficiently in the current applications and facilitating tuning applications to have less expensive patterns. For each memory location, CIAT keeps track of the type of accesses and the cores that perform these accesses. Consequently, CIAT can report the numbers of the following four types of communication patterns:

- Read after write (RAW) occurs when one core writes to a memory location and other core(s) read from this location. This is the main producer/consumer(s) communication pattern and usually involves copying the written data from the producer's cache.
- Write after read (WAR) occurs when a core writes to a memory location that was read by other core(s). This pattern usually involves invalidating the data copies in the other cores' private caches.
- Write after write (WAW) occurs when a core writes to a memory location that was written by another core. This pattern also involves invalidation and occurs when cores take turns on updating some shared locations.
- Read after read (RAR) occurs when a core reads from a memory location that was read by another core and the first visible access to this location is a read. Here, the data is usually replicated in the cores' caches.

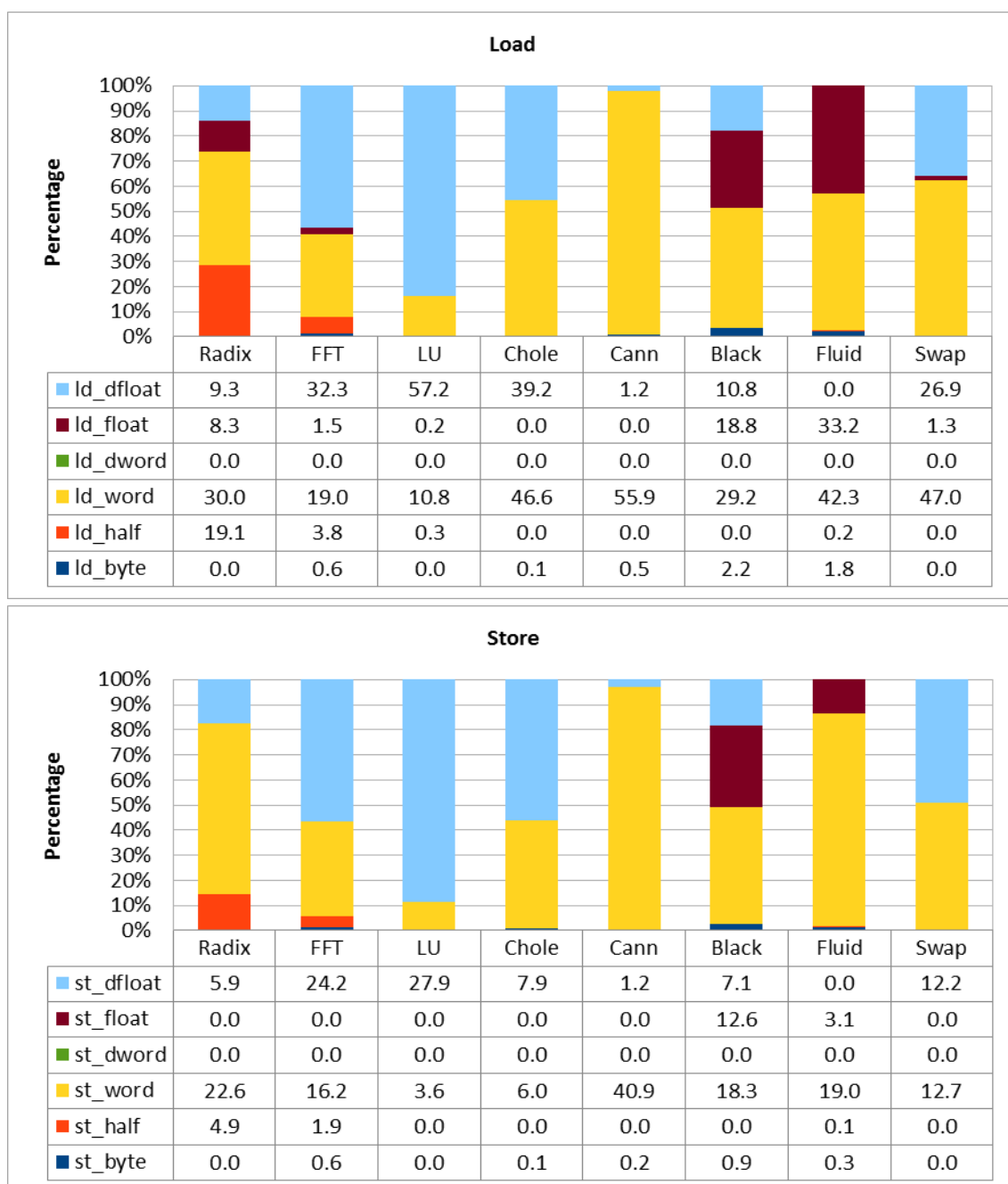


Figure 4. Percentages of the load and store operations according to the type and size of accessed data.

Figure 5 shows the percentages of these four communication patterns of the total number of memory accesses as function of the number of threads used. In the studied applications, PARSEC applications have less communication rates compared to SPLASH-2 (0.5% or lower). The communication rates generally increase as the number of threads increases, except for Blacksholes and Swaptions that have negligible rates.

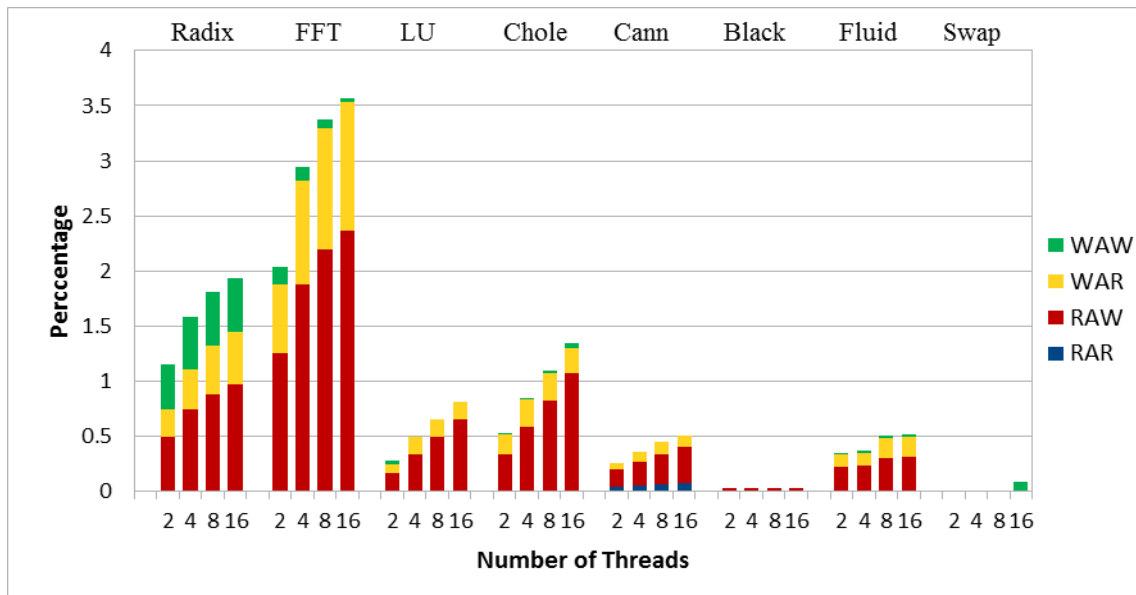


Figure 5. Percentages of the four communication patterns as a function of the number of threads.

Swaptions only has some WAW accesses when using 16 threads due to the reuse of some limited shared locations by these many threads.

Most of the remaining communication accesses are RAW and WAR. FFT has the largest percentage of WAR accesses, which results in large coherence traffic. In addition, almost all the communication accesses of Blacksholes are RAW; the shared memory locations of this application are generally not updated by WAR accesses after the first initialization. The figure shows that only Canneal has a small RAR rate due to reading shared data that is not initialized by the application's user code.

4.2.2 Sharing Degree

The *sharing degree* is the number of threads that read a memory location in the RAW pattern. Figure 6 shows the distributions of sharing degrees for the RAW accesses. It presents the percentages of sharing degrees when using 16 threads. These percentages are calculated by using the following formula:

$$\frac{S[p]}{\sum_{i=1}^{16} S[i]} \times 100\% ; \text{for } p = 1, \dots, 16$$

where $S[p]$ is the number of times that p threads read from a memory location after being previously written. Radix, FFT and Blacksholes have small sharing degrees, where almost all the shared locations are shared with only one thread each. Fluidanimate and Swaptions have two sharing degrees. In Fluidanimate, 76% of shared locations are shared with one thread and 23% with two threads. In Swaptions, 78% of shared locations are shared with one thread and 22% with two threads. LU, Cholesky and Canneal have also some sharing degrees higher than two. In LU, 97% of shared locations are shared with four threads and the remaining shared locations are shared with one, two or three threads. In Cholesky, about 58% of shared locations are shared with one thread and 42% are shared with two or more threads. In Canneal, 76% of shared locations are shared with one thread and 24% with two or more threads.

4.2.3 Invalidation Degree

The *invalidation degree* is the number of threads that have read a shared memory location in the WAR pattern.

Figure 7 shows the distributions of invalidation degrees for the WAR accesses. It presents the percentages of invalidation degrees when using 16 threads. These percentages are calculated by using the following formula:

$$\frac{I[p]}{\sum_{i=1}^{16} I[i]} \times 100\% ; \text{for } p = 1, \dots, 16$$

where $I[p]$ is the number of times that a memory location was updated after being previously read by p threads. The invalidation degrees in Radix, FFT, LU, Fluid and Swaptions are almost similar to their sharing degree, because the memory locations are iteratively shared and updated by RAW and WAR accesses. Choleskey and Canneal's invalidation degrees drop to one, because the locations that are shared with high degree are not updated by WAR accesses. Blackscholes has some invalidations of degree two, however, its WAR accesses are negligible compared to its RAW accesses.

4.2.4 Communication Slack

The *communication slack* is a measure to know how much time is present between writing a value to a memory location and referencing it by either read or write operation. CIAT measures this time by counting the number of instructions from writing the value until referencing it. The communication slack is distributed into eight ranges from less than ten instructions to more than ten million instructions.

Figure 8 shows the percentages of the communication slack distributions using 16 threads. These percentages are the number of instructions in each range over the total number of memory accesses. In all the studied applications, the communication has most of the slack in the range of tens of thousands of instructions and more. These ranges are enough to make use of prefetching.

4.2.5 Communication Locality

The *communication locality* is a measure of how the cores communicate with each other. Characterizing the communication locality helps both software developers in assigning threads to the cores and hardware designers in selecting a suitable system topology.

CIAT characterizes the communication locality by counting the number of communication events for each thread pair. CIAT maintains a 2D matrix for the communication events, where the rows represent the data producer threads and the columns represent the data consumer threads. For example, the value in Row i and Column j is the number of communication events from Thread T_i to T_j . This value is incremented by one whenever T_j reads from a location after T_i write (RAW), T_j writes to a location after T_i write (WAW) or T_i updates a location after T_j read (WAR).

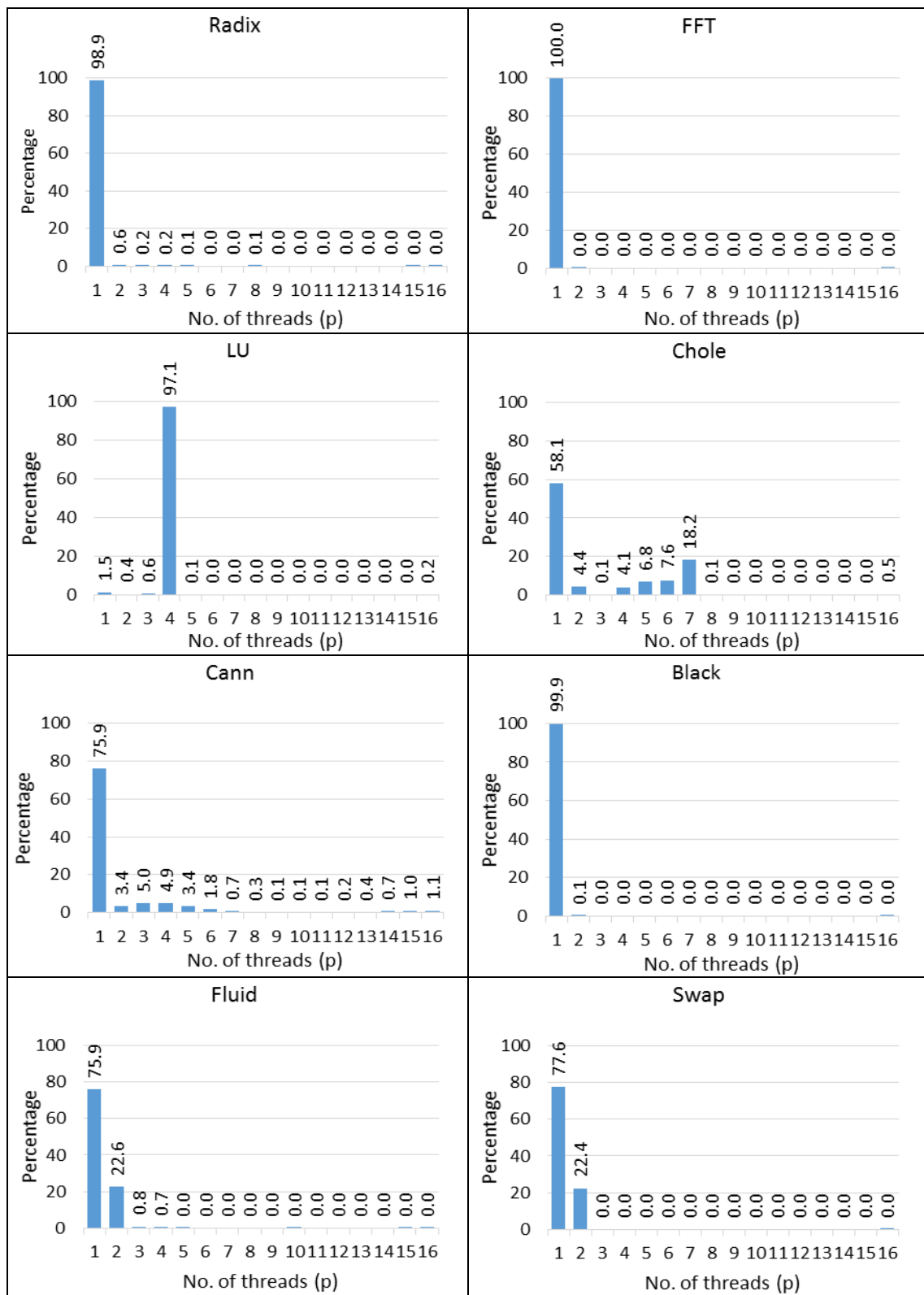


Figure 6. RAW sharing degree for 16 threads.

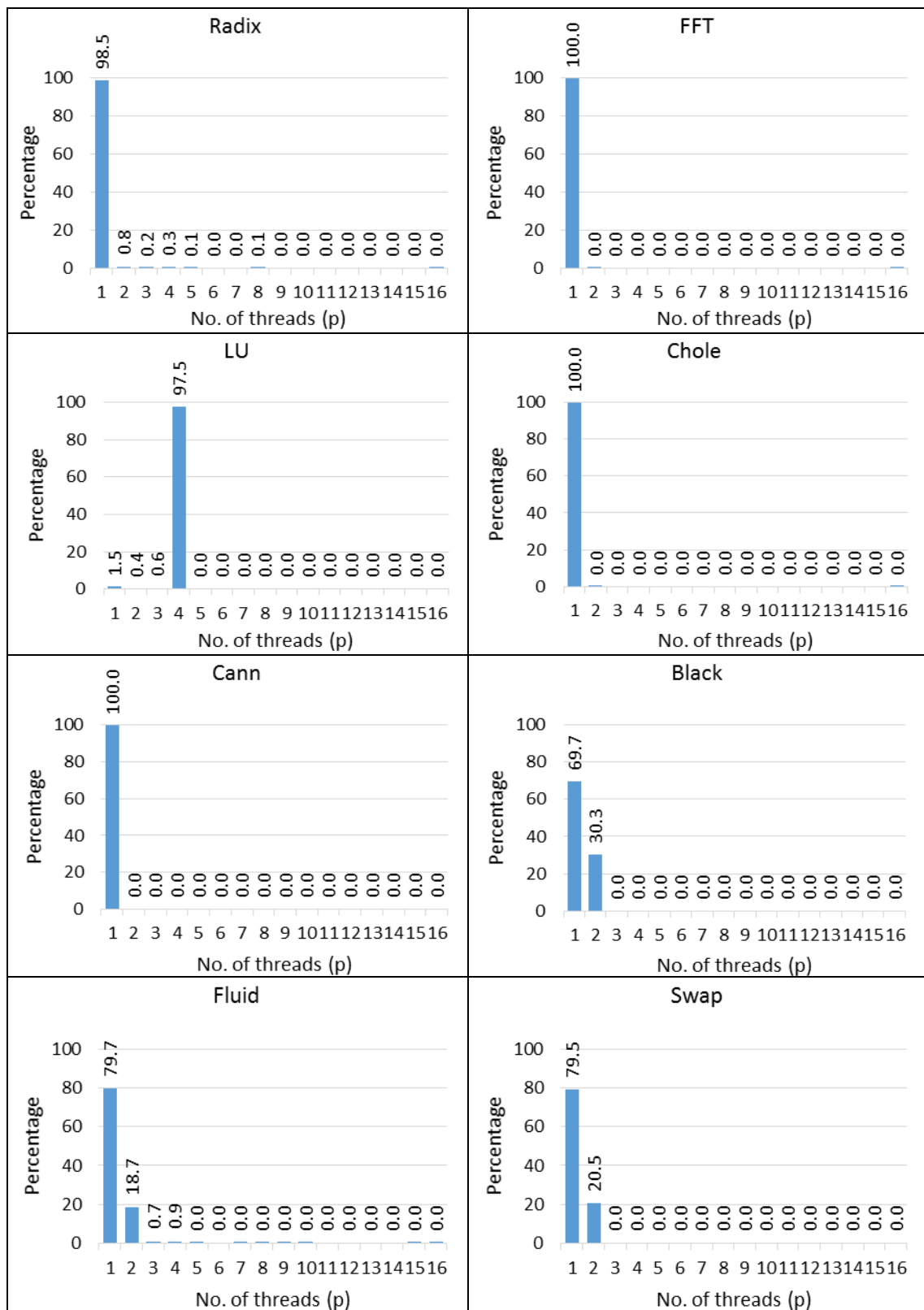


Figure 7. WAR invalidation degree for 16 threads.

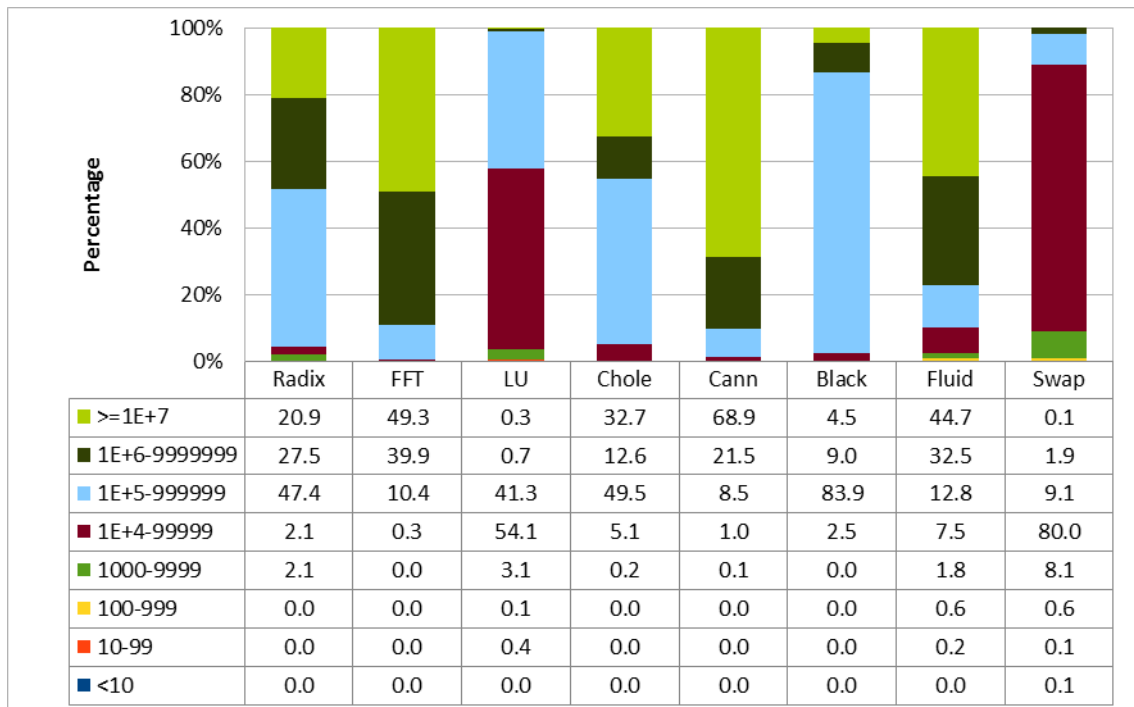


Figure 8. Communication slack distributions for 16 threads.

Figure 9 presents the communication locality when using 16 threads. In Radix, each thread communicates with all other threads. Also, there are some additional communications with the neighbors, where some odd threads communicate with only the next thread and some even threads communicate with more than one thread.

FFT has some uniform communication component and has also a large amount of communication from the initial thread to every other thread. In LU, the communication is clustered within groups of $g = n/4$ threads where n is the number of threads that are used to run the application. For example, when running LU using 16 threads, $g = 16/4 = 4$ threads. Additionally, each thread communicates to and from the thread that is located after multiple of g threads from it. For example, if $g = 4$, Thread 1 communicates with Threads 5, 9 and 13. Moreover, the initial thread communicates to all other threads and from the last g threads.

In Cholesky, the communication is non-uniform and each thread communicates with itself; i.e., each thread reads from or writes to memory locations that it previously wrote to and shared with other threads. The initial thread communicates with all other threads. Canneal has some uniform communication component and each thread communicates with itself and the initial thread communicates with all other threads. In Blackscholes, the communication is only with the initial thread; there is no data sharing among the other threads.

In Fluidanimate, the communication is non-uniform and each thread communicates with itself and the initial thread communicates with all other threads. Swaptions has low communication rates and each thread communicates with itself and there is some additional communication due to WAW accesses.

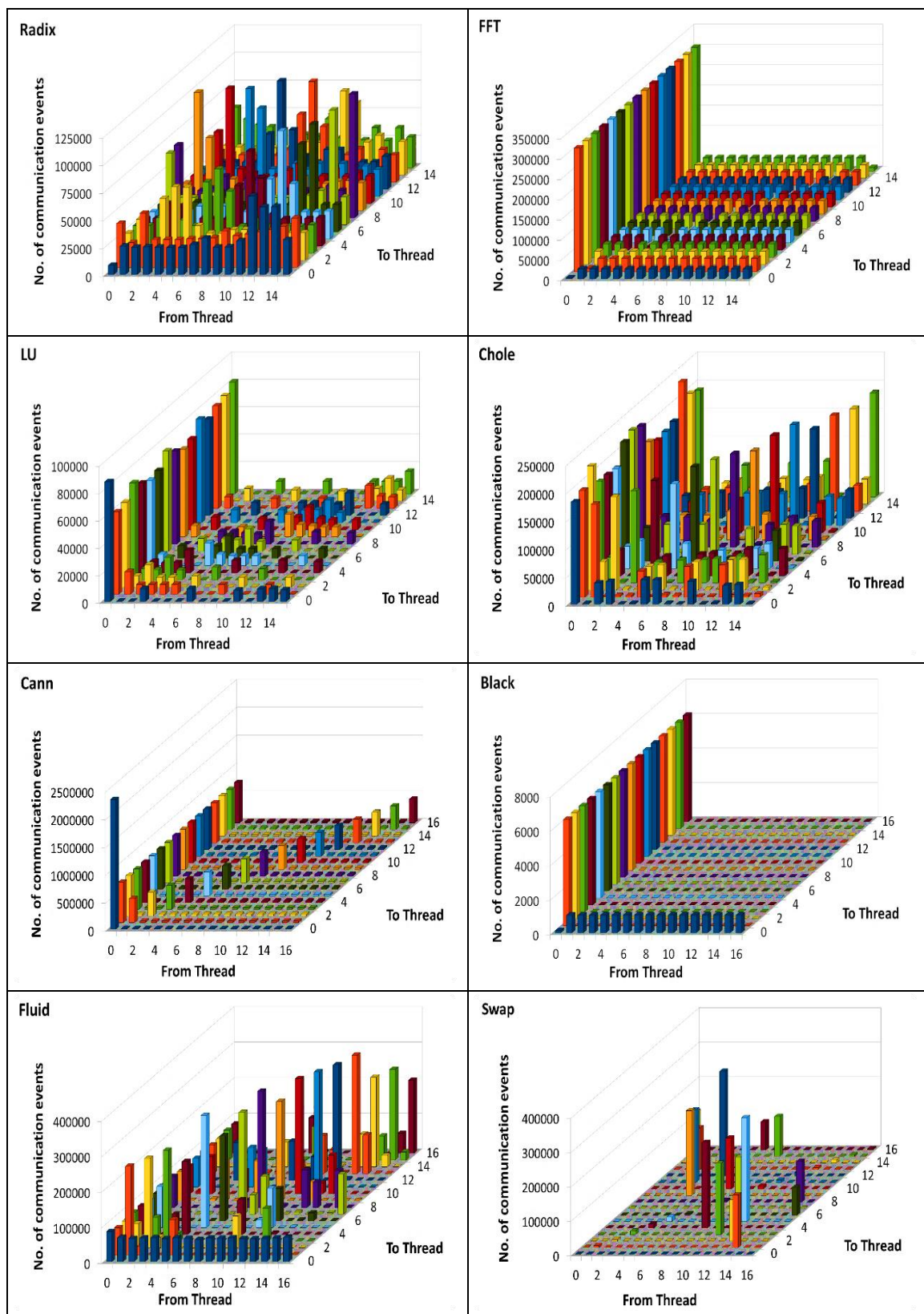


Figure 9. Number of communication events per thread pair for 16 threads.

5. CONCLUSIONS AND FUTURE WORK

Characterizing the inherent characteristics of multicore applications is important to help the programmers in tuning the current applications and developing future parallel applications, as well as to help designers in developing multi-core architectures that efficiently run parallel applications.

In this work, we have used on-the-fly configuration-independent analysis approach to characterize the inherent characteristics of eight multicore applications. Four applications are from SPLASH-2, which are: Radix, FFT, LU and Cholesky, and four from PARSEC, which are: Canneal, Blackscholes, Fluidanimate and Swaptions. The used on-the-fly approach is fast and enables analyzing large problems without needing a huge storage medium.

The obtained results show that the number of memory accesses, in the studied applications, does not change significantly as the number of threads increases. However, some applications such as Cholesky and Fluidanimate show high parallelization overhead, which is about 50% in Cholesky and about 33% in Fluidanimate. This overhead is due to the synchronization operation. Therefore, the speedup of these applications is limited by the increasing parallelization overhead. As expected, the largest percentages of memory accesses in the scientific applications are of floating point accesses.

The most common communication patterns are RAW and WAR except in Radix that has 36% of its communication in the WAW pattern and Swaptions that has 100% of its communication in the WAW pattern when using 16 threads. Therefore, designers must design systems that support these common patterns efficiently. Also, programmers must tune the applications to reduce these patterns. In general, the communication rates increase with more threads and PARSEC applications have rates smaller than SPLASH-2 applications.

Almost all the sharing in Radix, FFT and Blackscholes is with only one thread. In Fluidanimate and Swaptions, there are about 23% of sharing with two threads. In LU, Cholesky and Canneal, there are 97, 42 and 24% of sharing with two or more threads, respectively. The invalidation degrees in most of the applications are similar to their sharing degrees. There is considerable diversity in the communication locality of the studied applications. Some applications show uniform communication components such as FFT, Canneal and Blackscholes. Others show non-uniform communication and almost in all applications, the initial thread communicates with the other threads. Therefore, it is advisable to assign the initial thread to a central core to reduce the communication cost.

As future work, we plan to extend CIAT to capture the instruction stream in addition to capturing the data stream. Moreover, we need to develop CIAT to handle additional parallelization schemes such as the pipeline parallelization scheme that is used in three PARSEC applications: Dedup, Ferret and X264.

REFERENCES

- [1] K. Olukotun, B. A. Nayfeh, L. Hammond, K. Wilson and K. Chang, "The Case for a Single-chip Multiprocessor," *ACM Sigplan Notices*, vol. 31, no. 9, pp. 2–11, 1996.
- [2] D. Geer, "Chip Makers Turn to Multicore Processors," *Computer*, vol. 38, no. 5, pp. 11–13, 2005.
- [3] C. van Berkel, "Multi-core for Mobile Phones," in *Design, Automation Test in Europe Conference Exhibition*, pp. 1260–1265, 2009.
- [4] G. Blake, R. G. Dreslinski and T. Mudge, "A Survey of Multicore Processors," *Signal Processing Magazine, IEEE*, vol. 26, no. 6, pp. 26–37, 2009.

- [5] B. A. Mahafzah, "Performance Assessment of Multithreaded Quicksort Algorithm on Simultaneous Multithreaded Architecture," *The Journal of Supercomputing*, vol. 66, no. 1, pp. 339–363, 2013.
- [6] B. A. Mahafzah, "Parallel Multithreaded IDA* Heuristic Search: Algorithm Design and Performance Evaluation," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 26, no. 1, pp. 61–82, 2011.
- [7] G. A. Abandah and E. S. Davidson, "Origin 2000 Design Enhancements for Communication Intensive Applications," in *Proc. of the International Conference Parallel Architectures and Compilation Techniques (PACT'98)*, pp. 30–39, 1998.
- [8] J. Dongarra, S. Moore, P. Mucci, K. Seymour and H. You, "Accurate Cache and TLB Characterization Using Hardware Counters," in *Computational Science-ICCS 2004*, Springer, pp. 432–439, 2004.
- [9] M. Bhaduria, V. M. Weaver and S. A. McKee, "Understanding PARSEC Performance on Contemporary CMPs," in *IEEE Int'l Symp. Workload Characterization*, pp. 98–107, 2009.
- [10] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafae, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki and B. Falsafi, "Clearing the Clouds: A Study of Emerging Scale-out Workloads on Modern Hardware," *ACM SIGARCH Computer Architecture News*, vol. 40, no. 1, pp. 37–48, 2012.
- [11] Z. Jia, L. Wang, J. Zhan, L. Zhang and C. Luo, "Characterizing Data Analysis Workloads in Data Centers," in *IEEE Int'l Symp. Workload Characterization (IISWC)*, pp. 66–76, 2013.
- [12] W. E. Cohen and B. A. Mahafzah, "Statistical Analysis of Message Passing Programs to Guide Computer Design," in *Proceedings of the IEEE Thirty-First Hawaii International Conference on System Sciences*, vol. 7, pp. 544–553, 1998.
- [13] S. R. Alam, R. F. Barrett, J. A. Kuehn, P. C. Roth and J. S. Vetter, "Characterization of Scientific Workloads on Systems with Multi-core Processors," in *IEEE International Symposium on Workload Characterization*, pp. 225–236, 2006.
- [14] L. Chai, Q. Gao and D. K. Panda, "Understanding the Impact of Multicore Architecture in Cluster Computing: A Case Study with Intel Dual-core System," in *7th IEEE Int'l Symp. Cluster Computing and the Grid*, 2007, pp. 471–478.
- [15] G. A. Abandah, *Reducing Communication Cost in Scalable Shared Memory Systems*, Ph.D. dissertation, The University of Michigan, 1998.
- [16] A. Jaleel, R. S. Cohn, C.-K. Luk and B. Jacob, "CMP\$im: A Pin-based on-the-fly Multi-core Cache Simulator," in *Proc. 4th Annual Workshop on Modeling, Benchmarking and Simulation*, pp. 28–36, 2008.
- [17] G. Contreras and M. Martonosi, "Characterizing and Improving the Performance of Intel Threading Building Blocks," *Proc. of the IEEE International Symposium on Workload Characterization (IISWC 2008)*, pp. 57–66, 2008.
- [18] A. Bhattacharjee and M. Martonosi, "Characterizing the TLB Behavior of Emerging Parallel Workloads on Chip Multiprocessors," in *18th Int'l Conf. Parallel Architectures and Compilation Techniques*, pp. 29–40, 2009.
- [19] T. Dey, W. Wang, J. W. Davidson and M. L. Soffa, "Characterizing Multi-threaded Applications Based on Shared-resource Contention," in *IEEE Int'l Symp. Performance Analysis of Systems and Software*, pp. 76–86, 2011.
- [20] R. Natarajan and M. Chaudhuri, "Characterizing Multi-threaded Applications for Designing Sharing-aware Last-level Cache Replacement Policies," in *IEEE International Symposium on Workload Characterization*, pp. 1–10, 2013.
- [21] G. A. Abandah and E. S. Davidson, "Configuration Independent Analysis for Characterizing Shared-memory Applications," in *Proc. of the 12th International Parallel Processing Symp. (IPPS)*, pp. 485–491, 1998.

- [22] C.-K. Luk, R. Cohn, R. Muth, H. Patil, A. Klauser, G. Lowney, S. Wallace, V. J. Reddi and K. Hazelwood, "Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation," SIGPLAN Not., vol. 40, no. 6, pp. 190–200, 2005.
- [23] Intel, "Pin-A Dynamic Binary Instrumentation Tool," <https://software.intel.com/en-us/articles/pin-a-dynamic-binaryinstrumentation-tool/>, 2015, [Online; accessed 22-March-2015].
- [24] M. S. Mohammed, Hardware Configuration-independent Characterization of Multi-core Applications, Master's Thesis, The University of Jordan, Amman, 2015.
- [25] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," in ACM SIGARCH Computer Architecture News, vol. 23, no. 2, pp. 24–36, 1995.
- [26] C. Bienia, S. Kumar, J. P. Singh and K. Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in Proc. of the 17th Int'l Conf. Parallel Architectures and Compilation Techniques, pp. 72–81, 2008.

ملخص البحث:

اكتسبت معماريات المعالج متعدد النوى انتشاراً متزايداً في السنوات الأخيرة. إلا أن العديد من التطبيقات المتاحة لا تستفيد تمام الاستفادة من هذه المعماريات. لذا، فقد طوّر باحثون كُثُر تقنيات عديدة لوصف الخصائص من أجل مساعدة المبرمجين على فهم سلوك هذه التطبيقات على المنصات متعددة النوى وضبطها للحصول على فاعلية أفضل. تقترح هذه الورقة منحى لوصف الخصائص، مباشرة، من دون الاعتماد على التكوين؛ لوصف الخصائص الكامنة للتطبيقات متعددة النوى. وهذا المنحى يمتاز بالسرعة؛ لأنه لا يعتمد على تفاصيل أي تكوين آلة بعينه، ولا يتطلب إعادة وصف الخصائص لكل تكوين مستهدف. فهو يتتبع فقط إمكانيات الوصول إلى الذاكرة والنوى التي تؤدي تلك الإمكانيات من خلال إيصال بيانات تتبع الذاكرة على نحو فوري إلى أداة التحليل.

لقد تم تطبيق هذا المنحى على ثمانية تطبيقات لوصف خصائصها، مأخوذة من مجموعتي المقارنة (SPLASH-2) سبلاش ٢ و (PARSEC) بارسيك. تعرض هذه الورقة الخصائص الكامنة لهذه التطبيقات، بما في ذلك تعليمات الوصول إلى الذاكرة، وأنماط خصائص الاتصال، ودرجة التشارك، ودرجة الإبطال، وتراخي الاتصال، ومحلية الاتصال. وتُظهر النتائج أن إثنتين من التطبيقات المدروسة لهما سقف تواز عال، وهما: تشولسكي و (Fluidanimate) فلويدانيميت. كما تشير النتائج إلى أن التطبيقات المدروسة من مجموعة "سبلاش ٢" تمتلك معدّلات اتصال أعلى مقارنة بالتطبيقات المدروسة من مجموعة "بارسيك"، وأن هذه المعدّلات تزداد بشكل عام كلما ازداد عدد المسارات (Threads) المستخدمة. ويحدث التشارك والإبطال في معظمه بدرجات قليلة. ومع ذلك، كان لإثنتين من تطبيقات مجموعة "سبلاش ٢" جزء مهم من الاتصال بدرجات تشارك عالية باستخدام أربعة مسارات أو أكثر. وكانت لمعظم التطبيقات مركبة اتصال موحّدة، وكان المسار الابتدائي بشكل عام منخرطاً في قدر أكبر من الاتصال مقارنة بالمسارات الأخرى.

A LOW COMPLEXITY DIRECTION FINDING SYSTEM BASED ON A SIX-PORT INTEGRATED MIMO ANTENNA SYSTEM

Rifaqat Hussain¹, Ali H. Muqaibel², Wajih Abu-Al-Saud³
and Mohammad S. Sharawi⁴

Electrical Engineering Department, King Fahd University for Petroleum and Minerals
(KFUPM), Dhahran 31261 Saudi Arabia
{rifaqat¹, muqaibel², wajih³, msharawi⁴}@kfupm.edu.sa

(Received: 15-Dec.-2015, Revised: 28-Jan.-2016, Accepted: 03-Feb.-2016)

ABSTRACT

In this paper, a low complexity microwave based direction finding (DF) system is presented. The proposed system consists of a single six-port (SP) circuit integrated with a reconfigurable multiple-input-multiple-output (MIMO) antenna system. The SP circuit covers a wide frequency band (1.68-2.25 GHz). The SP circuit is also characterized for phase error compensation caused by the slight asymmetry of SP and power detectors. The reconfigurable MIMO antenna system used is a compact design and covers several well-known wireless standards in the frequency bands from 0.7 GHz to 3 GHz. The SP circuit is integrated with the reconfigurable MIMO antenna system to form a complete beam forming mode for second generation cognitive radio (CR) platforms. The proposed design is a complete integrated solution with DF capabilities for CR platforms. The design is suitable to be used in compact wireless handheld and mobile communication devices. The fabricated integrated system achieves $\pm 16^\circ$ accuracy in its direction of arrival estimates.

KEYWORDS

Microwave DF, Six-port circuit, Beamforming mode, Reconfigurable MIMO antenna, Cognitive radio platforms.

1. INTRODUCTION

Low complexity direction finding (DF) systems in wireless communication devices have attracted increasing attention over the past decade. Radio frequency (RF) based DF schemes are of particular interest, because of their low profile RF structure and their minimal data processing requirements. RF DF schemes are gaining popularity in wireless communication devices and in military services. Wireless handheld devices complying with 3G/4G wireless standards integrated with RF DF systems would be an attractive feature for next generation cognitive radio (CR) platforms [1].

The basic concept of RF DF is an angle-of-arrival (AoA) estimation of an incoming RF signal from a distant source. Classical techniques of digital signal processing (DSP) algorithms, such as Multiple Signal Classification (MUSIC) and Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT), use an array of antennas followed by multiple receivers to estimate the AoA [2]-[3]. With the use of such single or multiple receivers, computationally intensive DF algorithms and techniques are limited to be used in practical wireless handheld and mobile devices [4]. RF DF systems using the six-port (SP) circuit have gained popularity over the past few years, because of their low cost and simple microwave structure [5].

Most of the existing 3G/4G wireless standards cover low frequency bands, while most of the existing SP structures cover frequency bands above 2GHz [5]–[7]. In addition to high frequency operation, the given SP circuit dimensions are not suitable to be used in wireless handheld devices. A low frequency compact SP circuit was presented in [8]. Although the SP design was compact and covered low frequency bands, it was not presented in a complete integrated solution for AoA estimation. Also, a dual SP design was presented in [9] with low frequency operation. In [10], an ultra-wideband (UWB) six-port network was presented with operating bands of 2~8 GHz. It consisted of a Wilkinson power divider and three 3-dB quadrature couplers. The fabricated SP phase measurement system with calibration technique based on support vector regression (SVR) was introduced. Results show that when the SVR model was efficiently utilized, a phase error of $\pm 1.5274^\circ$ was obtained.

In this work, we propose a complete integrated solution complying with the second generation CR standards in mobile devices. The system integrates a compact SP circuit based on [8] with a reconfigurable MIMO antenna system based on [11]. The antenna system used covers several frequency standards in the frequency bands from 0.7 to 3 GHz, while the SP circuit used covers frequency bands from 1.68 to 2.25 GHz. The unique feature of the proposed design is the integration of the SP circuit with the reconfigurable MIMO antenna system for lower frequency bands of operation. The complete system was tested and its AoA determination capability was studied in a complete experimental setup.

Moreover, the integrated design is easily distinguished from all other contemporary designs, as it could be utilized in wireless handheld devices and mobile terminals. Additionally, the complete system can be used in data Tx/Rx mode and beam forming mode complying with second generation CR standards.

2. DF IN COGNITIVE RADIO PLATFORMS

The revolutionary technique of CR is defined as a system with efficient utilization of frequency spectrum along with direction finding capabilities [1]. In this section, the classification of CR antennas, their use in beamforming mode and operating principles are discussed.

2.1 Beamforming in CR Platforms

Front-end antennas for CR platforms are categorized as shown in Figure 1. CR antennas consist of two types of antennas: (1) an ultra-wide-band (UWB) sensing antenna and (2) a reconfigurable communication antenna. Reconfigurable antennas can be utilized to change their operating fundamental characteristics; i.e., resonance frequency, radiation pattern, polarization and impedance bandwidth. Reconfigurable antennas can be simple frequency reconfigurable ones or reconfigurable MIMO antennas that can be utilized to enhance the data rate capability. The MIMO antenna system can be used in two modes of operation in a CR platform: (1) Data Tx/Rx mode and (2) Beamforming mode. Beamforming mode in CR platforms can be utilized for RF DF.

2.2 DF Operating Principle

The block diagram of the operating principle of AoA estimation based on RF DF using the SP circuit is shown in Figure 2. The detailed description of the operating principle of the SP circuit for RF DF is given in detail in [8]. The receiving antennas are separated by distance d with a path difference Δd , while the received signals experience a phase difference $\Delta\phi$. AoA (ϕ) of a distant object can be calculated using Δd and $\Delta\phi$ information using Equation (1), where λ is the wave length of operating frequency.

$$\Delta d = \lambda \frac{\Delta\phi}{2\pi}, \quad \sin \phi = \frac{\lambda \Delta\phi}{a 2\pi} \quad (1)$$

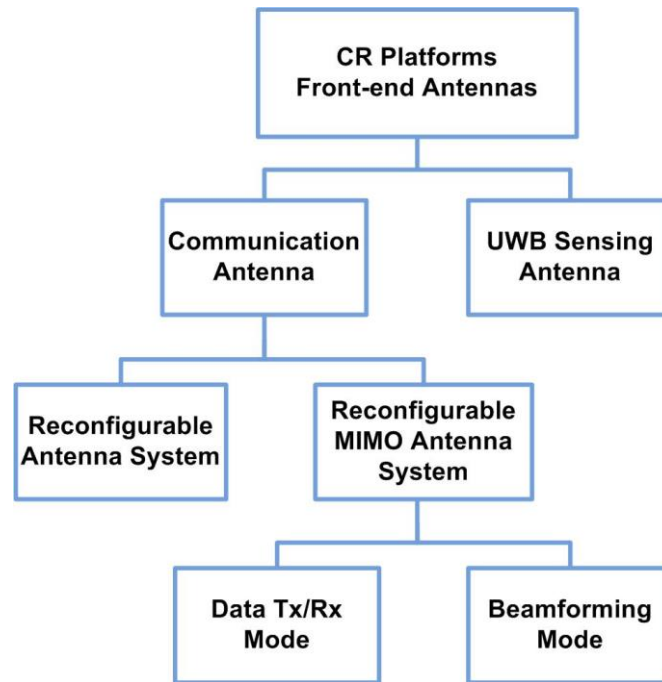


Figure 1. Classification of CR antennas.

Equation (1) can be used to determine the AoA of the incoming RF signal based on the information of phase difference. The SP circuit is fed with incoming wave signals from the antennas and used to find the phase difference. Once the phase difference becomes known, it could be utilized to find the AoA of distant target objects.

2.3 SP Operating Principle

In this sub-section, the operating principle of RF DF using SP circuit is described. Two input RF signals a_5 and a_6 are received by two antennas separated by distance d . The two signals are impinging on the SP circuit with phase difference $\Delta\phi$ owing to path difference Δd . The phase difference can be written as $\Delta\phi = \phi_6 - \phi_5$. The complete details of the SP circuit for RF DF are discussed in [9]. The SP circuit used in this experimentation is shown in Figure 3. The SP circuit output scan be written as:

$$b_i = a_5 \cdot S_{5i} + a_6 \cdot S_{6i} \quad (2)$$

The resultant signal at output port-1 can be written as:

$$b_1 = \frac{a}{2} \cdot e^{j[\phi_5 - \frac{\pi}{2}]} \{1 + \alpha e^{j[\Delta\phi + \pi]}\} \quad (3)$$

For port-2, the resultant signal is given by:

$$b_2 = \frac{-a}{2} \cdot e^{j[\phi_5]} \{1 + \alpha e^{j[\Delta\phi + \frac{\pi}{2}]}\} \quad (4)$$

Similarly, for port-3, the resultant signal is:

$$b_3 = \frac{-a}{2} \cdot e^{j[\phi_5]} \{1 + \alpha e^{j[\Delta\phi]}\} \quad (5)$$

Similarly, for port 4:

$$b_4 = \frac{-a}{2} \cdot e^{j[\phi_5 + \frac{\pi}{2}]} \{1 + \alpha e^{j[\Delta\phi - \frac{\pi}{2}]}\} \quad (6)$$

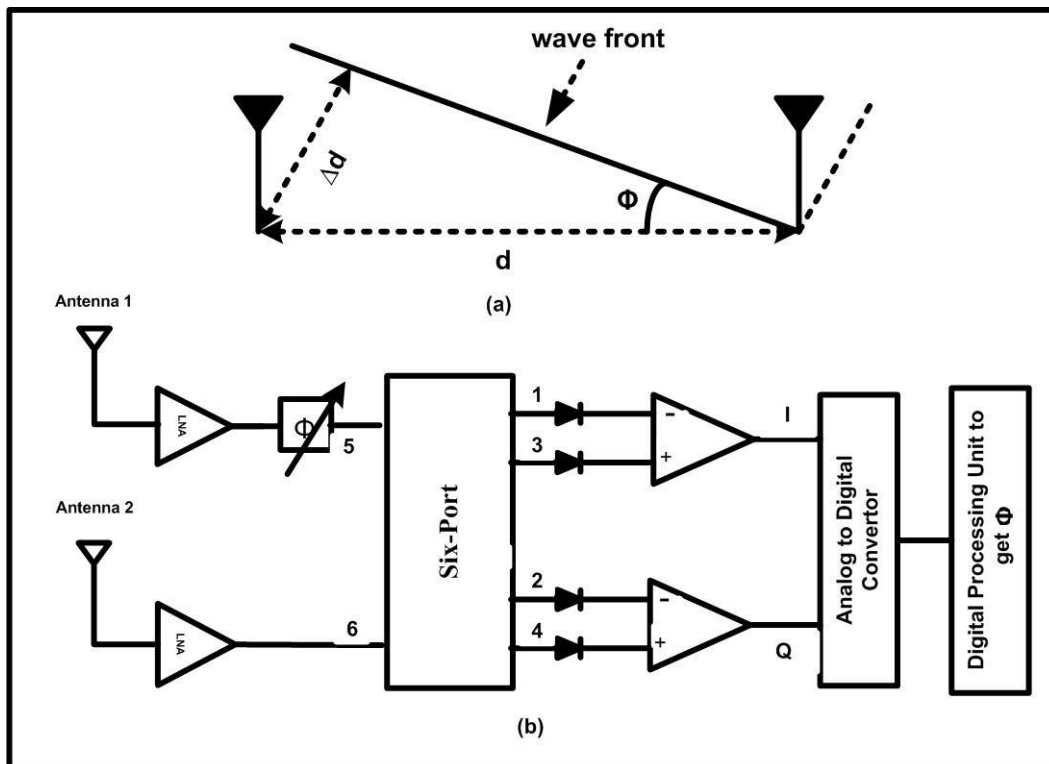


Figure 2. Six-port based direction finding system.

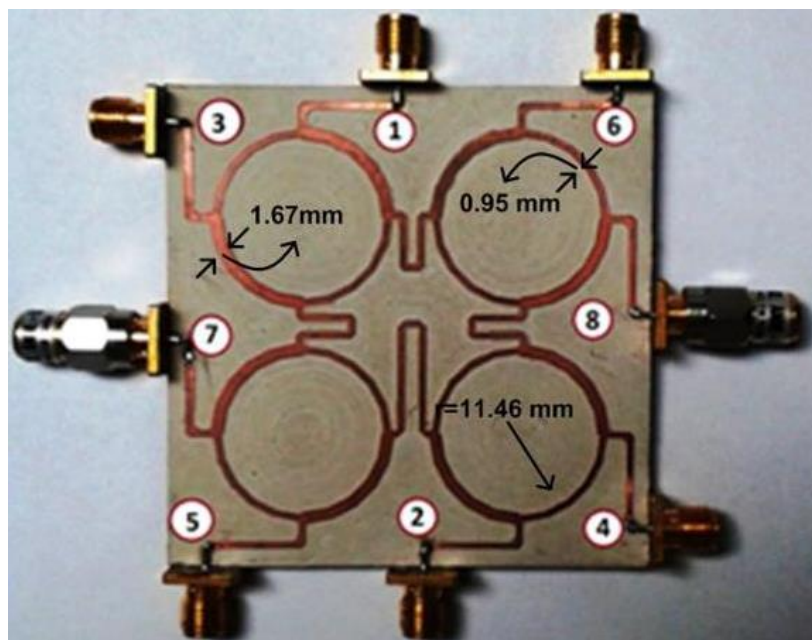


Figure 3. Single SP circuit [8].

The SP output RF signals are passed through power detectors and can be written as:

$$V_i = K_i |b_i|^2 \quad (7)$$

$$i=1,2,3,4$$

where K_i constants are measured in V/W. The constants are specified for each power detector showing the relationship between output voltage and input power. Each output of the power detector circuit is passed through a differential amplifier circuit to get the two components in-phase and quadrature (I and Q) [9]. For the SP, these are I and Q .

$$I = V_3 - V_1 = \alpha K a^2 \cdot \cos(\Delta\phi) \quad (8)$$

$$Q = V_4 - V_2 = \alpha K a^2 \cdot \sin(\Delta\phi) \quad (9)$$

The vector Γ is defined for the dual SP in terms of the in-phase and quadrature components as:

$$\Gamma = I + jQ = K \cdot a^2 \cdot \exp(j\Delta\phi) \quad (10)$$

where a is the amplitude of the incoming RF signal and α is the ratio a_6/a_5 . Equation (10) can be used to determine the phase relationship between the input RF signals. The measured $\Delta\phi$ can be used in Equation (1) to find the AoA (ϕ) of the distant target object.

3. CHARACTERIZATION OF THE SIX-PORT OUTPUTS

The direction of the incoming RF waves can be easily determined using Equation (1) based on the determination of the phase differences measured by the SP given in Equation (10). But this calculation overlooks the phase error caused by the slight asymmetry of the various SP paths, asymmetry of the power detectors, as well as frequency measurement errors. These errors in the phases were characterized via simulations and by laboratory measurements. In this section, a mathematical analysis is carried out to develop an analytical model to compensate for these errors in order to get high accuracy in phase measurement at the output of each port.

$$\begin{aligned} \phi'_{5i} &= \text{phase error at output port } i \text{ due to input port 5, where } i = 1,2,3,4; \\ \phi'_{6i} &= \text{phase error at output port } i \text{ due to input port 6, where } i = 1,2,3,4 \end{aligned} \quad (11)$$

So, the output signal can be written as a linear combination of input signals a_5 and a_6 .

$$\begin{aligned} b_1 &= a_5 \cdot S_{51} + a_6 \cdot S_{61} \\ b_1 &= a \cdot e^{j\phi_5} \cdot e^{j\phi'_{51}} - \frac{j}{2} + \alpha a_5 \cdot \frac{j}{2} e^{j\Delta\phi} e^{j\phi'_{61}} \\ b_1 &= \frac{a}{2} \cdot e^{j[\phi_5 - \frac{\pi}{2}]} \cdot e^{j\phi'_{51}} + \alpha \cdot a \frac{j}{2} e^{j\phi_5} e^{j\Delta\phi} e^{j\phi'_{61}} \\ b_1 &= \frac{a}{2} \cdot e^{(j[\phi_5 - \frac{\pi}{2}])} \left\{ e^{j\phi'_{51}} + \alpha e^{[j(\Delta\phi + \pi)]} e^{j\phi'_{61}} \right\} \\ b_1 &= \frac{a}{2} \cdot e^{(j[\phi_5 - \frac{\pi}{2}])} \left\{ 1 + \alpha e^{[j(\Delta\phi + \pi)]} \frac{e^{j\phi'_{61}}}{e^{j\phi'_{51}}} \right\} \\ b_1 &= \frac{a}{2} \cdot e^{(j[\phi_5 - \frac{\pi}{2}])} e^{j\phi'_{51}} \left\{ 1 + \alpha e^{[j(\Delta\phi + \pi)]} e^{j\Delta\phi'_1} \right\} \end{aligned} \quad (12)$$

$$\begin{aligned} b_2 &= a_5 \cdot S_{52} + a_6 \cdot S_{62} \\ b_2 &= \frac{a}{2} \cdot e^{j\phi_5} \cdot e^{j\phi'_{52}} - 1 + \alpha a_5 \cdot \frac{-j}{2} e^{j\Delta\phi} \cdot e^{j\phi'_{62}} \\ b_2 &= \frac{-a}{2} \cdot e^{(j[\phi_5])} e^{j\phi'_{52}} \left\{ 1 + \alpha e^{[j(\Delta\phi + \frac{\pi}{2})]} e^{j\Delta\phi'_2} \right\} \end{aligned} \quad (13)$$

$$\begin{aligned}
b_3 &= a_5 \cdot S_{53} + a_6 \cdot S_{63} \\
b_3 &= \frac{-a}{2} \cdot e^{j\phi_5} \cdot e^{j\phi'_{53}} - 1 + \alpha a_5 \cdot \frac{-1}{2} e^{j\Delta\phi} \cdot e^{j\phi'_{63}} \\
b_3 &= \frac{-a}{2} e^{j[\phi_5]} + \alpha a \frac{-1}{2} e^{j\phi_5} e^{j\Delta\phi} \\
b_3 &= \frac{-a}{2} \cdot e^{(j[\phi_5])} e^{j\phi'_{53}} \{1 + \alpha e^{[j(\Delta\phi)]} e^{j\Delta\phi'}\}
\end{aligned} \tag{14}$$

$$\begin{aligned}
b_4 &= a_5 \cdot S_{54} + a_6 \cdot S_{64} \\
b_4 &= \frac{-j}{2} a \cdot e^{(j[\phi_5])} \cdot e^{j\phi'_{64}} + \alpha a \frac{-1}{2} e^{j\phi_5} e^{j\Delta\phi} \cdot e^{j\phi'_{64}} \\
b_4 &= \frac{-a}{2} e^{(j[\phi_5 + \frac{\pi}{2}])} \cdot e^{j\phi'_{64}} + \alpha a \frac{1}{2} e^{j\phi_5} e^{j\Delta\phi} - j \cdot -j \cdot e^{j\phi'_{64}} \\
b_4 &= \frac{-a}{2} e^{(j[\phi_5 + \frac{\pi}{2}])} \cdot e^{j\phi'_{64}} \left\{1 + \alpha e^{[j(\Delta\phi - \frac{\pi}{2})]} e^{j\Delta\phi'}\right\}
\end{aligned} \tag{15}$$

Supposing that four identical detectors ($K_i = K$) are used, the dc output voltages including the error terms become:

$$V_1 = K_1 |b_1|^2 = K \frac{a^2}{4} [1 + \alpha^2 - 2\alpha \cdot \cos(\Delta\phi + \Delta\phi'_1)] \tag{16}$$

$$V_2 = K_2 |b_2|^2 = K \frac{a^2}{4} [1 + \alpha^2 - 2\alpha \cdot \sin(\Delta\phi + \Delta\phi'_2)] \tag{17}$$

$$V_3 = K_3 |b_3|^2 = K \frac{a^2}{4} [1 + \alpha^2 + 2\alpha \cdot \cos(\Delta\phi + \Delta\phi'_3)] \tag{18}$$

$$V_4 = K_4 |b_4|^2 = K \frac{a^2}{4} [1 + \alpha^2 + 2\alpha \cdot \sin(\Delta\phi + \Delta\phi'_4)] \tag{19}$$

where

$$\Delta\phi'_1 = \phi'_{61} - \phi'_{51}$$

$$\Delta\phi'_2 = \phi'_{62} - \phi'_{52}$$

$$\Delta\phi'_3 = \phi'_{63} - \phi'_{53}$$

$$\Delta\phi'_4 = \phi'_{64} - \phi'_{54}$$

In the I/Q complex plane, a Γ vector can be defined using the four six-port DC output voltages with error terms. The error terms are known a priori and can be used to adjust or compensate for errors to give high precision in phase measurement.

Once the error in each phase of the SP is characterized, the error in the incoming RF wave can be analyzed based on Equations (16)–(19). This phase error in each path can be compensated for using an additional phase compensation block to have more accurate DF results. This block could be implemented in hardware using a field programmable gate array (FPGA) or software based-phase compensation can produce satisfactory results.

4. SP CIRCUIT INTEGRATION WITH THE RECONFIGURABLE MIMO ANTENNA SYSTEM

The complete measurement setup of an RF DF consists of a transmitting antenna as a source and a receiving MIMO antenna system integrated with SP circuit for AoA estimation.

4.1 Reconfigurable MIMO Antenna System

The receiving antenna used was a reconfigurable modified PIFA MIMO antenna system along with its UWB sensing antenna as shown in Figure 4. The complete antenna system consists of two printed circuit boards with main board dimensions of $65 \times 120 \times 1.56 \text{ mm}^3$. The bottom layer consisted of the UWB sensing antenna. The secondary elevated board contained the reconfigurable antenna that was short-circuited with GND plane. The complete antenna system was realized on a commercial FR-4 substrate with $\epsilon_r=4.4$ [11].

Each MIMO antenna element was embedded with two PIN diodes. The diodes were used for discontinuity of the antenna structure. The diodes were used to provide more flexibility by their ON/OFF operation for the various antenna radiating structures. The two PIN diodes in each antenna element resulted in four distinct operating modes for each MIMO antenna element. The various modes were used to resonate the antenna at various bands to cover particular wireless standards [11].

4.2 Measurements Setup

To find the AoA from a distant source, a reconfigurable MIMO antenna system integrated with a single SP circuit was used. The source used was an UWB antenna. A reconfigurable MIMO antenna system was used as the receiving system. Figure 5 shows an overview of the implemented system. The receiving antennas (within the MIMO antenna system) were separated by a distance of 53 mm (centre to centre). Owing to the path difference between the two receiving antennas, the propagated signal had a shift that was utilized to find the AoA.

4.3 Details of the Receiver Setup

A detailed view of the measurement setup at the receiving side is shown in Figure 6. The receiving antenna used is a two-element reconfigurable MIMO antenna system. The antenna operates at four distinct modes covering several frequency bands. The details of each mode and its operating frequencies were completely described in [11]. In the current scenario, we have used it in mode-1 and mode-4 at frequency bands 2020 MHz and 1690 MHz. The proposed SP circuit covered frequency ranges starting from 1.68 GHz to 2.25 GHz. The two reconfigurable antennas were connected with low noise amplifiers (LNA). The LNA used was the ZX60-33LN-S+, with a wide bandwidth covering frequency bands from 50~3000MHz. Its gain was between 13 ~ 14.4 dB in the frequency range between 1690~2020MHz.

The output of the LNA was fed to the SP circuit at port-5 and port-6. Two ports of the SP circuit were terminated with 50Ω loads. The four output ports of the SP structure were connected with the power detectors (ZX47-40LN-S+) followed by a difference amplifier based on the LM 741 IC. The output of the SP was passed through a power detector to get the DC signal. The power detector covered a wide range with low noise DC output and needed a single 5V supply. The outputs of the difference amplifiers were the in-phase (I) and quadrature (Q) components. Both (I) and (Q) were DC and were acquired by a data acquisition card (LabJack u6). Two channels were used for acquiring this data. The data acquired by the data acquisition system were processed in MATLAB for AoA estimation using Equation (1). Figure 7 shows all the components of the measurement setup. Figure 7(a) shows the source, while Figure 7(b) shows the setup used for the direction of the source. Figure 7(c) shows the SP circuit integrated with the MIMO antenna system, while Figure 7(d) shows the LabJack u6 interface for data acquisition.

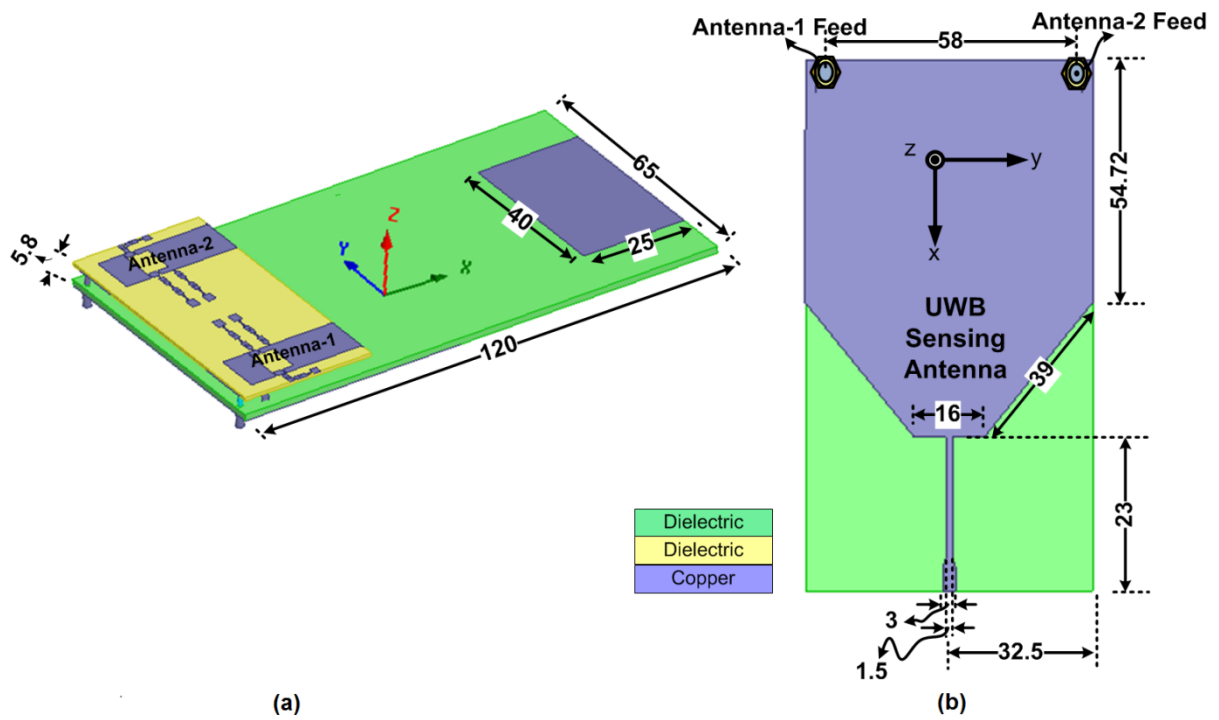


Figure 4. Reconfigurable MIMO antenna system (a) Top view (b) Bottom view. [11] - All dimensions are in mm.

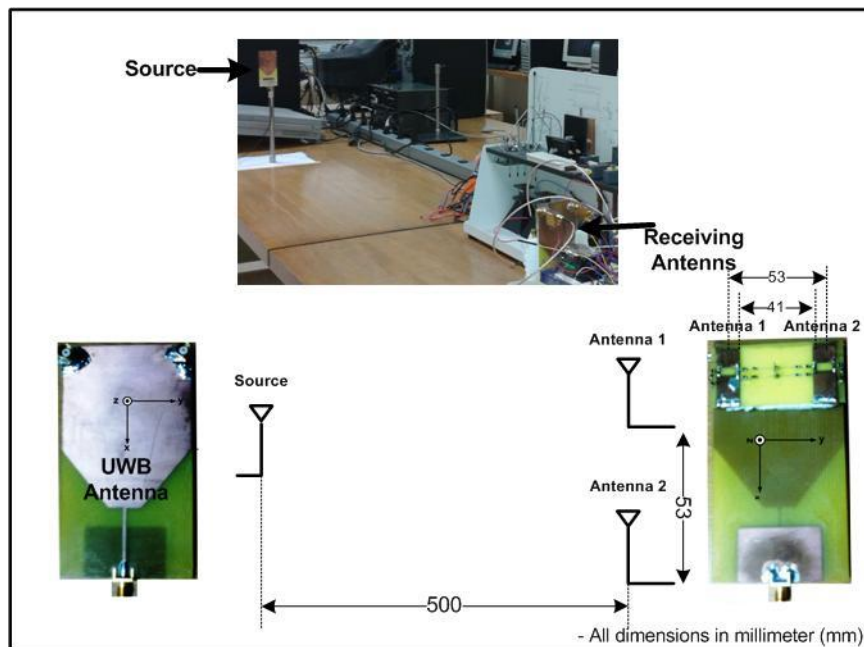


Figure 5. Block diagram of the DF measurement setup.

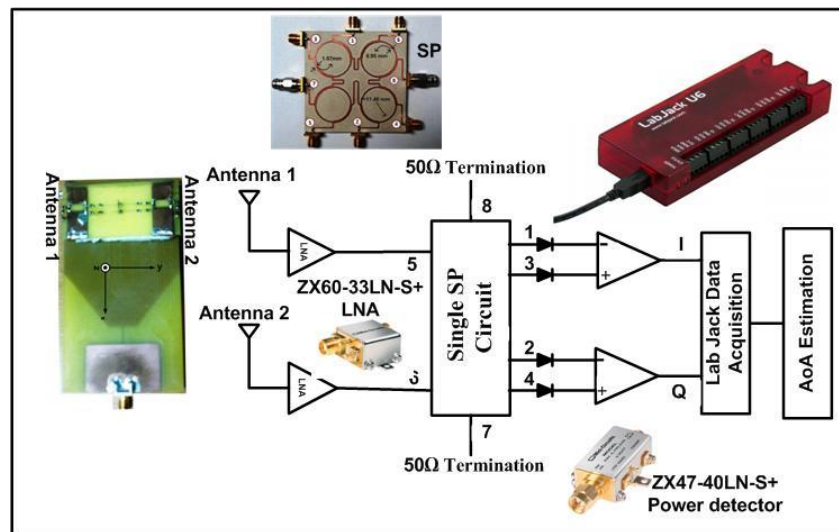


Figure 6. Detailed view of SP integration with reconfigurable MIMO antenna system for RF DF.

5. MEASUREMENT RESULTS

In the setup shown in Figure 7, a single SP circuit was used to determine the AoA in 2D. Using a single SP with a two-element antenna setup can be used to determine the AoA in a single plane. For a complete 3D (i.e., θ and ϕ) DF, a dual SP circuit with four antenna elements is required. Due to hardware limitations, in this work we have determined the AoA in 2D only.

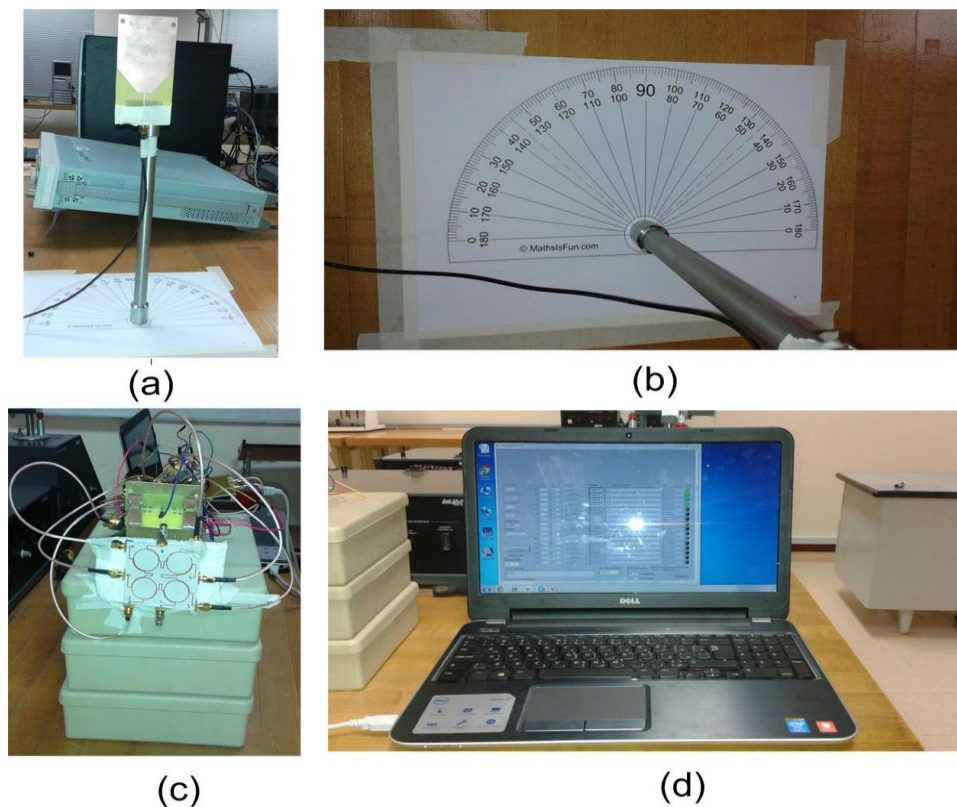


Figure 7. (a) RF source (b) Setup for angle adjustment (c) SP circuit integrated with a two-MIMO antenna system (d) LabJack interface for data acquisition.

5.1 Description of AoA Measurements

The objective of this work was to find the AoA of an RF distant source using the SP circuit integrated with reconfigurable MIMO antenna system. In this experiment, AoA measurements were made under known conditions. The receiving antenna was located with known orientation. The AoA of the incoming signal wave was known beforehand. A single antenna was positioned at known angles with respect to the receiver. The receiving antenna was operated with a single tone signal at 1690 MHz and 2020 MHz for two different measurements of AoA determination. This experiment was conducted at the Microwave Lab at KFUPM. In this AoA experiment, the transmitting and receiving antennas were placed at a distance of 500 mm. The distance was made to ensure the minimum level of power to be received for accurate AoA estimation. The orientation of the transmitting antenna was changed for azimuth angles between $\pm 80^\circ$. It has been found that the error was becoming drastic for angles above $\pm 60^\circ$. The maximum errors found in the AoA measurements for azimuth angle between $\pm 60^\circ$ were 16° and 13° for frequency bands 1690 MHz and 2020 MHz, respectively. The error introduced in the phase by the SP was subtracted from the final error plots (the phase response of the SP was obtained from the measured parameters). Figures 8 and 9 show the error in the estimated AoA ($\tilde{\phi}$) based on the measured values of (I) and (Q) using Equation (1) for the frequency bands 1690 MHz and 2020 MHz. The figures show the error in estimated AoA ($\tilde{\phi}$). Practically, with the current setup of SP and two-element MIMO antenna, the feasible range of scanning angle is from -60° to 60° with a maximum phase error in estimated AoA ($\tilde{\phi}$) of 16° in the given two bands of operation.

5.2 Sources of Error in AoA Estimation

The proposed design was able to estimate the AoA with a maximum error of $\pm 16^\circ$. Although, the error was high, it helps in understanding the problem and its implementation. The possible sources of error are:

1. Ideally, the antenna elements are supposed to be 0.5λ apart. The accuracy of the results drops for closely spaced antenna elements. In the current scenario, the two antennas are separated by a distance of 0.23λ and 0.27λ , for the two frequency bands of 1690 MHz and 2020 MHz, respectively.
2. Although all the circulators in the SP were designed to be symmetrical in the SP design, due to fabrication tolerances, some phase error was observed at the output of SP circuit and this contributes to the final estimated error.
3. Asymmetry and non-linearities resulting from the power detectors and difference amplifiers contributed to phase errors as well.
4. Different circuit modules were connected using wires. The slight difference in the lengths of wires might have added extra phase, thus contributing to this final error.

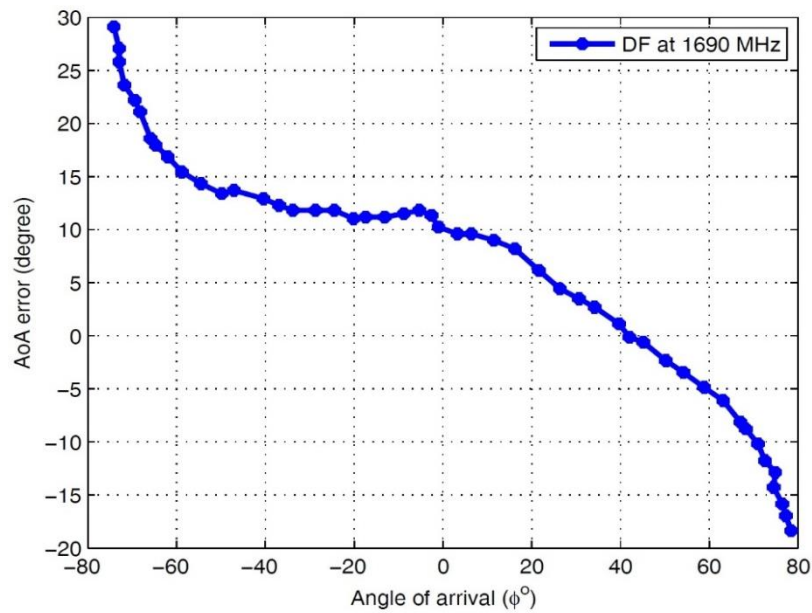


Figure 8. Angle of arrival at 1690 MHz.

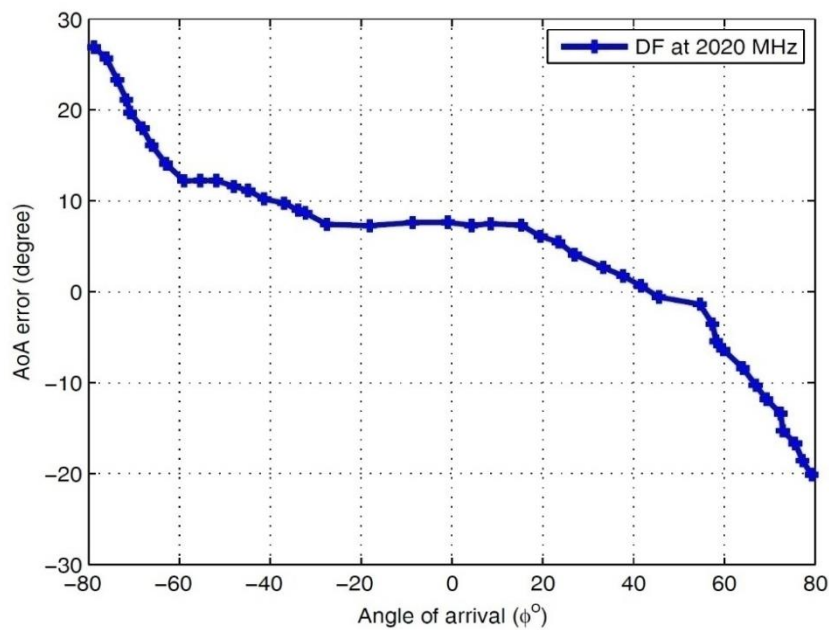


Figure 9. Angle of arrival at 2020 MHz.

6. CONCLUSIONS

In this paper, a low profile RF DF system is proposed for second generation CR applications. The complete integrated system is of low cost, targeting lower frequency bands of practical wireless devices and could be utilized for low processing DF systems in wireless handheld devices and mobile terminals. The compact single SP circuit integration with the multi-band reconfigurable MIMO antenna system is unique due to its contemporary design for RF DF. The complete system is versatile, as it could be utilized to enhance the data throughput as well as in beam forming mode for AoA estimation. The maximum error observed using this complete system was $\pm 16^\circ$.

ACKNOWLEDGEMENTS

This project was funded by the National Plan for Science, Technology and Innovation (Maarifah) - King Abdul Aziz City for Technology- through the Science and Technology Unit at King Fahd University of Petroleum and Minerals (KFUPM), Kingdom of Saudi Arabia; award number 12-ELE3001-04.

REFERENCES

- [1] J. Bernhard, J. Reed, J. Park, A. Clegg, A. Weisshaar and A. Abouzeid, "Final Report of the National Science Foundation Workshop on Enhancing Access to the Radio Spectrum (EARS)," Arlington, Virginia, 4-6 August 2010.
- [2] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp.276–280, 1986.
- [3] R. Roy and T. Kailath, "Esprit-estimation of Signal Parameters via Rotational Invariance Techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [4] D. Peavey and T. Ogumfunmi, "The Single Channel Interferometer Using a Pseudodoppler Direction Finding System," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 4129–4132, 1997.
- [5] G. Vinci, F. Barbon, B. Laemmle, R. Weigel and A. Koelpin, "Wide Range Dual Six-Port Based Direction-of-arrival Detector," in *IEEE 7th German Microwave Conference (GeMiC)*, pp. 1–4, 2012.
- [6] S. Tatu, K. Wu and T. Denidni, "Direction-of-arrival Estimation Method Based on Six-port Technology," *IEE Proceedings-Microwaves, Antennas and Propagation*, vol. 153, no. 3, pp. 263–269, 2006.
- [7] H. Peng, Z. Yang and T. Yang, "Design and Implementation of a Practical Direction Finding Receiver," *Progress in Electromagnetic Research Letters*, vol. 32, pp. 157–167, 2012.
- [8] R. Hussain and M. S. Sharawi, "Compact Low Frequency Six-port Design for Wireless Communication Devices," *17th IEEE Mediterranean Electrotechnical Conference (MELECON)*, pp. 29-32, 13-16 April 2014.
- [9] R. Hussain and M. S. Sharawi, "A Dual Six-port with Two-angle Resolution and Compact Size for Mobile Terminals," in *IEEE Radio and Wireless Symposium (RWS)*, pp. 226–228, 2014.
- [10] H. Peng, Y. Ziqiang Yang and T. Yang, "Design and Implementation of an Ultra-wideband Six-port Network," *Progress in Electromagnetics Research* 131, pp. 293-310, 2012.
- [11] R. Hussain and M. S. Sharawi, "Integrated Reconfigurable Multiple-input–multiple-output (MIMO) Antenna System with an Ultra-wideband Sensing Antenna for Cognitive Radio Platforms," *IET Microwaves, Antennas and Propagation*, vol. 9, no. 9, pp. 940-947, 2015.
- [12] S. O. Tatu, E. Moldovan, K. Wu, R. G. Bosisio and T. A. Denidni, "Ka Band Analog Front-end for Software-defined Direct Conversion Receiver," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 9, pp. 2768–2776, 2005.

ملخص البحث:

في هذا البحث، يتم تقديم نظام منخفض التعقيد لإيجاد الاتجاه في نطاق الميكرووييف. النظام المقترح يتكون من دائرة مفردة سداسية المنافذ، مدمجة مع نظام هوائيات متعدد المداخل- متعدد المخارج قابل لإعادة التشكيل. تغطي الدائرة سداسية المنافذ نطاقاً ترددياً واسعاً (١,٦٨ - ٢,٢٥ جيجا هيرتز). كذلك تم وصف خصائص الدائرة سداسية المنافذ لتعويض خطأ الطور الناجم عن عدم التجانس الطيف للمنافذ الستة وكواشف القدرة. نظام الهوائيات متعدد المداخل- متعدد المخارج القابل لإعادة التشكيل المستخدم في هذا البحث ذو تصميم مدمج، ويغطي عدة مقاييس معروفة جيداً في مجال الاتصالات اللاسلكية في النطاقات الترددية من ٠,٧ جيجا هيرتز إلى ٣ جيجا هيرتز.

لقد تم دمج الدائرة سداسية المنافذ مع نظام الهوائيات متعدد المداخل- متعدد المخارج القابل لإعادة التشكيل لتشكيل نسق كامل لتكوين الشعاع للجيل الثاني من منصّات الرّاديو المدركة. والتصميم المقترح هو حلّ متكامل يمتلك القدرة على إيجاد الاتجاه لمنصّات الرّاديو المدركة.

والجدير بالذكر أنّ التصميم المقترح مناسب للاستخدام في أجهزة الاتصال اللاسلكية المدمجة المحمولة باليد و النقالية. ويحقق النظام المدمج المُصنّع دقّة في تقدير اتجاه الإشارات المستقبلية في حدود $\pm ١٦^\circ$.

PERFORMANCE EVALUATION OF META-HEURISTICS IN ENERGY AWARE REAL-TIME SCHEDULING PROBLEMS

Ashraf Suyyagh¹, Jason G. Tong² and Zeljko Zilic³

Department of Electrical and Computer Engineering,
McGill University, Montreal, Canada
ashraf.suyyagh@mail.mcgill.ca¹, jason.tong@mail.mcgill.ca²,
zeljko.zilic@mcgill.ca³

(Received: 15-Dec.-2015, Revised: 03-Feb.-2016, Accepted: 10-Feb.-2016)

ABSTRACT

Energy efficient real-time systems have been a prime concern in the past few years. Techniques at all levels of system design are being developed to reduce energy consumption. At the physical level, new fabrication technologies attempt to minimize overall chipset power. At the system design level, technologies such as Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Power Management (DPM) allow for changing the processor frequency on-the-fly or go into sleep modes to minimize operational power. At the operating system level, energy-efficient scheduling utilizes DVFS and DPM at the task level to achieve further energy savings. Most energy-efficient scheduling research efforts focused on reducing processor power. Recently, system-wide solutions have been investigated. In this work, we extend on the previous work by adapting two evolutionary algorithms for system-wide energy minimization. We analyse the performance of our algorithms under variable initial conditions. We further show that our meta-heuristics statistically provide energy minimizations that are closer to the optimum 85% of the time compared to about 30% of those achieved by simulated annealing over 500 unique test sets. Our results further demonstrate that in over 95% of the cases, meta-heuristics provide more minimizations than the CS-DVS static method.

KEYWORDS

Real-time systems, Embedded systems, Energy-aware scheduling, Meta-heuristics, DVFS, DPM.

1. INTRODUCTION

Embedded systems are evolving into cyber-physical systems; highly interconnected, tightly-coupled systems with the physical world. The consolidation of the physical world into the interconnected virtual world requires the use of many devices. Cyber-physical systems heavily rely on sensors, communication devices and even the cloud. Such devices claim significant portion of the system power profile and their share can no longer be excluded in energy minimization. As a result, system-wide power reduction becomes a significant challenge.

Traditionally, research efforts have focused on processor power optimizations. The processor dynamic power consumption highly depends on operational voltage and frequency as given by Equation 1:

$$P_{dynamic} = C_{eff} \cdot V_{dd}^2 \cdot f \quad (1)$$

where C_{eff} is the effective switching capacitance, V_{dd} is the supply voltage and f is the operational frequency [21]. With the advancement of fabrication technologies and the shrinking

of transistors' size into the nanometer scale, effects of sub-threshold leakage current become more prominent. Transistor miniaturization allows for the reduction of the supply voltage. However, sub-threshold leakage current I_{sub} is inversely proportional to V_{dd} . This becomes more evident in Equation 2:

$$I_{sub} = K_1 e^{\frac{-V_{th}}{nV_\theta}} (1 - e^{\frac{-V_{dd}}{V_\theta}}) \quad (2)$$

where K_1 and n are experimentally derived, V_θ is the thermal voltage which is 25mV at room temperature and increases linearly with temperature. Increasing V_{th} to reduce I_{sub} is not a viable option, as any increase in V_{th} will reduce the maximum processor frequency and therefore affect the performance. The relationship between threshold voltage V_{th} and frequency f is given by Equation 3:

$$f \propto \frac{(V_{dd} - V_{th})^\alpha}{V_{dd}} \quad (3)$$

where α is a technology parameter. To reduce processor dynamic power, Dynamic Voltage and Frequency Scaling (DVFS) was introduced [1]-[2]. DVFS allows for the run-time change of operating voltages/frequencies. Modern operating systems can access and operate processor DVFS hardware. On the other hand, to reduce leakage current, Dynamic Power Management (DPM) techniques are employed to turn off the processor [3] or inactive devices [4]-[5] when idle.

Previous research has targeted system-level optimizations. It was shown that when system devices are involved, the interplay between processor DVFS/DPM and device DPM techniques is complex [6]-[8]. DVFS techniques could lower processor energy but increase device energy. Many devices are expected to remain in the active state for the entire duration the task is running on the processor [9]. Devices are allowed to switch into their low power states if they are no longer used by successive scheduled tasks and only if it is energy efficient to do so. As task execution times are scaled using DVFS techniques, the time the associated devices are expected to be powered on is scaled as well, and more energy is consumed [9]-[10]. Moreover, even though employing DVFS reduces dynamic power, the reduction of V_{dd} increases leakage current, and the prolonged execution of the task due to frequency scale down increases the overall leakage power consumption. Therefore, an optimal frequency scaling assignment needs to balance and minimize the overall system energy; that is of the processor and the devices combined. This problem is known to be an NP-Hard problem [11]. As such, design time algorithms developed specialized heuristics [11]-[12] or were based on mathematical optimizations such as integer linear programming [9].

This work extends on previous research by adapting different meta-heuristics to minimize system-wide energy and evaluating their performance. Meta-heuristics have been used in scheduling problems related to energy minimization, load balancing [13], [22] or makespan minimization. The genetic algorithm (GA) is one the earliest meta-heuristic evolutionary algorithms employed for task scheduling and partitioning and for test set generation [14]-[16]. Differential Evolution (DE) is one of the newer evolutionary algorithms which differs in that it is not biologically inspired, but rather relies on stochastic approaches [17]. Simulated annealing (SA) is another meta-heuristic which approximates a global optimum of an NP-hard problem. Simulated annealing was employed in [10] to minimize power consumption of a periodic hard-real time system. We summarize our contributions in this work as follows:

- We propose and adapt two meta-heuristic evolutionary algorithms to find the DVFS configurations which minimize system energy. We analyse and compare the performance proposed algorithms against each other and against previous work.
- We investigate the optimal parameters for these algorithms and attempt to establish confidence in the ability of such algorithms in producing near-optimal energy savings.

The organization of the paper is as follows: Section 2 introduces the system's task and energy models. Section 3 describes the proposed algorithms as well as the reference algorithms. The simulation methodology is presented in Section 4 and our analysis and results are summarized in Section 5.

2. SYSTEM MODELS

In this section, the power and task models used in this paper are presented. The notations used to describe task and power parameters are listed in Table 1.

Table 1. List of notations used in the paper.

τ_i	Task i , $i = 1, 2, \dots, N$
D_i	Deadline of Task i
C_i	Worst Case Execution Time (WCET) of i under maximum system frequency F_1
T_i	Period of task τ_i
F_i	Normalized discrete CPU frequency levels, $F_i \in F_1, F_2 \dots F_Q$ and $F_1 > F_2 > \dots > F_Q$
d_k	Device k , $k = 1, 2, \dots, K$
s	The deep sleep mode where the least power is consumed
t_{sw}^0	Total switching time overhead when the processor or device d_k is switched down from active state to the low power state t_{sd}^0 and up from the low power state to the active state t_{su}^0
E_{sw}^0	Total energy dissipated when the processor (CPU) or device d_k is switched down from active state to the low power state E_{sd}^0 and back up from the low power state to the active state E_{su}^0 , respectively
P_s^0	Device d_k or processor (CPU) power consumed while the device is in the low power state s
$P_a^{d_k}$	Device d_k power consumed in the active state
$P_{F_i}^{CPU}$	Processor power consumed in the active state while running at frequency level F_i
t_{BET}^0	Break even time of device d_k or the processor

2.1 Task Model

The system is a hard real-time system based on a set of N independent, periodic and fully preemptible tasks with implicit deadline model (deadline equals the period). Tasks are assumed schedulable under EDF policy when executed with no frequency or voltage scaling techniques employed. Each task τ_i is represented by the tuple (C_i, D_i, T_i) denoting task worst case execution time, deadline and period, respectively. Each task τ_i is assigned a number of devices d_k and a frequency scaling factor F_i . Similar to previous work [10], [23], we assume an inter-task device scheduling model. That is, devices are available, active and running throughout the associated task run time. Devices can be switched into an inactive state only at the end of the associated task run-time. Switching the device to low power state takes place if the device is no longer needed for a subsequent task and if it is energy-efficient to go to low-power state. The hyper-period (HP) is the least common multiple of all task periods and represents the period in which the task scheduling pattern repeats. Utilization is measured by $\sum_1^N \frac{C_i}{T_i}$ and must be less than or equal to one for a feasible EDF schedule. In accordance with previous research [9]-[10], [7], the relationship between task execution time and frequency scaling is assumed to be linear and the execution time for task τ_i after scaling is measured by C_i/F_i . Slack time (processor / device idle time) is only utilized for the possibility of switching the processor / device to low-power mode.

2.2 System Power Model

We consider a processor and devices which have one active state and one low power (deep sleep) state s . The deep sleep state s is the state where most of the processor/device components are turned off. This model is used to keep the design space exploration for the algorithms proposed in this paper manageable. Yet, the system could be readily extended to support multiple low power states s_i . In the active state, the processor is capable of executing tasks at one of Q -discrete frequencies. F_i is the normalized frequency corresponding to frequency f_i ; where $f_i > f_{i+1} > \dots > f_Q$. The frequencies are normalized to the highest system frequency f_1 . Switching from the processor active state to the lower power states s entails switching time and energy overheads, defined as t_{sw}^{CPU} and E_{sw}^{CPU} , respectively. The switching overheads include both the switching overheads from active to low power state and vice versa. The power consumed while the processor is in the active state depends on the currently selected frequency and is denoted $P_{F_i}^{CPU}$. Power consumed while the processor is in deep sleep state s is denoted as P_s^{CPU} .

Similar to the processor model, the device power consumed in active and low power states are denoted as $P_a^{d_k}$ and $P_s^{d_k}$, respectively. And device time and energy switching overheads between the active and low power state are represented by $t_{sw}^{d_k}$ and $E_{sw}^{d_k}$, respectively.

Processors' and devices' transition from their active states to a lower power state occurs only when the transition is power-efficient. Switching states is considered power-efficient if the total switching power between active, low-power to active states is less than the power consumed if the processor/device is kept idle in the active state. The decision to switch to a lower power processor or device state is based on the break-even time t_{BET} . t_{BET} represents the minimum idle time threshold required to switch to a lower power state to satisfy the power-efficiency condition. Equation 4 computes the break-even time for the processor [18].

$$t_{BET}(CPU) = \max \left(t_{sw}^{CPU}, \frac{E_{sw}^{CPU} - P_s^{CPU} \times t_{sw}^{CPU}}{P_{F_i}^{CPU} - P_s^{CPU}} \right) \quad (4)$$

Similarly, Equation 5 computes the break-even time for any device d_k .

$$t_{BET}(d_k) = \max \left(t_{sw}^{d_k}, \frac{E_{sw}^{d_k} - P_s^{d_k} \times t_{sw}^{d_k}}{P_a^{d_k} - P_s^{d_k}} \right) \quad (5)$$

3. ENERGY AWARE SCHEDULING ALGORITHMS

In this section, we propose and adapt two meta-heuristics for system wide energy minimization. One is based on a discrete implementation of genetic algorithm for frequency scale assignment. The other is based on the newer differential evolution algorithm. In section 5, we compare these algorithms against each other and against the simulated annealing meta-heuristic based on the recent work of [10]. We also compare the results to the famed heuristic algorithm CS-DVS [7] which is one of the most powerful and regularly cited algorithms. We summarize the simulated annealing algorithm and present the CS-DVS for completeness at the end of this section.

3.1 Definitions

Before introducing the proposed energy aware scheduling algorithms, a few definitions need to be presented, as they are frequently encountered in the subsequent sections.

Definition 1: A power configuration is defined as a permutation of DVFS frequency assignments of dimension N (mapped to each task τ_i), a power cost variable and a set of status flags. Flags convey information on the feasibility of the configuration or control decision paths within the algorithm.

Definition 2: The quality of a configuration (and algorithm) is defined as how well it is able to minimize system energy to near optimal values (within 1% of optimal).

Definition 3: A feasible power configuration is one which satisfies EDF scheduling feasibility condition when DVFS is employed; that is $\sum_1^N \frac{C_i}{T_i \times F_i} \leq 1$.

3.2 Genetic Algorithm Frequency Scaling (GAFS)

The genetic algorithm is an optimization algorithm based on the principles of genetics. The genetic algorithm mimics the process of natural selection and the survival of the fittest. The algorithm starts with a population of random solutions of a certain size that is allowed to evolve through time towards global optimum. Each member of this population is called a chromosome. Each chromosome consists of a set of variables which are called genes.

The genetic algorithm passes through multiple iterations. In each iteration, the cost of each chromosome is evaluated by applying the chromosome genes (variables) into the objective function. The objective function represents the problem which we aim to minimize or maximize. The chromosomes are then sorted in terms of their cost. Half of the chromosomes which have costs closer to the optimum are maintained, while the other half is discarded. This set of preserved chromosomes is called parents. Parents are used to generate the other missing half of the population which is termed offspring or children. Parent chromosomes are paired amongst each other. Genes are exchanged between paired chromosomes in an operation called crossover. There exist many techniques and strategies for pairing parents and gene crossover [20]. The current population of parent and child chromosomes is subjected to a mutation operation. A certain percentage of genes across all chromosomes is randomly selected and altered. This is to mimic genetic mutations which occur in nature. This new population is now processed in the very same manner in the next iteration. The algorithm iterates until it converges to a solution.

In our work, a discrete non-binary version of the genetic algorithm is implemented to find a power configuration that minimizes system power. The power configuration used in GAFS is comprised of a chromosome c of N genes and a set of configuration flags, where N is the system tasks count. Each gene represents a possible task frequency assignment F_i for task τ_i and is initialized by the index i of the frequency scale level F_i assigned to the task. This is due to the fact that we are using a discrete and integer genetic algorithm. The flags specify whether the configuration is feasible, a parent chromosome or in mutated state. Mutated chromosomes are the ones with one or more genes randomly changed through the mutation operation. Crossover and mutation operations are detailed below. The genetic algorithm for frequency assignment (GAFS) is listed in Algorithm 1.

Initially, a set of power configurations of size NP are initialized with random frequency scaling factors such that each configuration is feasible. All configurations are set to be parents and in an unmodified state. Each initial configuration is run for one hyper-period and the system power cost is computed and assigned to the configuration. The lowest power $NP/2$ chromosomes are selected as parents and sorted from lowest to highest power. A top-bottom pairing approach is used to pair parents from the parent pool. A one point crossover operation is performed, where genes are exchanged between the paired chromosomes. No minimum less than 25% and no maximum more than 40% of chromosome genes take part in the crossover process. After the exchange, a mutation operator is applied on the whole population except for the elite chromosome, which is the one that yielded the overall minimum system energy. The elite chromosome carries from one generation to another and is only replaced if another chromosome yields lower system power. The number of mutations applied is calculated according to Equation 6.

Algorithm 1 GAFS

```

1: Initialization:
2: iteration  $\leftarrow$  0
3: for  $i \leftarrow 1, NP$  do ▷ NP: Population Size
4:    $c_i \leftarrow$  Random and feasible frequency scales
5:   Set  $c_i$  as parent, feasible, and unmodified
6:   Run  $c_i$  in next HP - measure and store power
7:   iteration  $\leftarrow$  iteration + 1
8: end for
9: while iteration <  $max\_hp$  do
10:  Produce Next Generation:
11:  Sort NP configurations from min to max power
12:  Select best  $\frac{NP}{2}$  configurations for crossover and mutation
13:  Pair best configurations from lowest to highest power
14:  Perform one point cross over and replace discarded configurations
15:  Set state for new configurations as child
16:  for  $n \leftarrow 1, \#mutations$  do
17:    Randomly choose a configuration from new generation (exclude best/elite)
18:    Randomly mutate frequency assignment to new value
19:    if chosen configuration is parent then
20:      Change parent state to modified
21:    end if
22:  end for
23:  for  $j \leftarrow 1, NP$  do
24:    if new  $c_i$  is unfeasible then
25:      Set power of unfeasible configuration to  $\infty$ 
26:    end if
27:  end for
28:  for  $n \leftarrow 1, NP$  do
29:    if new  $c_i$  is feasible configuration then
30:      if new  $c_i$  is child or modified parent then
31:        Load new configuration  $c_i$ , measure power in HP
32:        iteration  $\leftarrow$  iteration + 1
33:      end if
34:    end if
35:  end for
36:  if No configuration is feasible in current generation then
37:    Terminate. Choose elite chromosome as solution
38:  end if
39: end while

```

$$Mutations = \mu \times (NP - 1) \times N; \quad (6)$$

where NP and μ are the population size and the mutation factor, respectively.

Finally, each new chromosome configuration is checked for feasibility. If it fails, then its cost is set to ∞ . Parent chromosomes which have undergone mutation are marked as such.

Only feasible child and modified parent configurations are run in subsequent hyper-periods. This eliminates redundant computation for the elite and unmodified parents. The whole process repeats for each generation of power configurations until the maximum hyper-periods specified are reached or the algorithm converges. Ideally, if each generated chromosome is feasible, then the lower bound of the number of generations produced is $\left\lceil \frac{HP}{NP} \right\rceil$, where HP is the number of test hyper-periods. The upper bound case will be when the algorithm is only able to produce one feasible configuration per generation. The upper bound will be equal to $1 + HP - NP$. If the

algorithm returns no feasible chromosomes to be run, then it stops and the elite chromosome is used as a solution.

3.3 Differential Evolution Frequency Scaling (DEFS)

Differential evolution is an optimization algorithm which belongs to the same group of evolutionary algorithms as that of the genetic algorithm. Differential evolution is founded on stochastic principles to find a solution. The algorithm maintains a set of solutions called candidate vectors which evolve through iterative operations. Each vector is comprised of a set of variables which are applied to the objective function to calculate the cost of a solution.

The differential evolution algorithm passes through multiple iterations. In each iteration, for each vector in the population (called base or parent vector), a new vector is created. Each new vector is generated from the addition of a scaled difference of two different candidates to another third candidate. The set of new candidates are called donor vectors. A new vector called the trial vector is generated from each base vector and its corresponding donor vector based on a certain probability. The cost of the trial vector is measured. It replaces its corresponding base vector only if it is closer to the optimum. Otherwise, it is discarded.

In our adaptation of the differential evolution algorithm which is listed in Algorithm 2, a configuration in DEFS is comprised of an N -dimensional vector v , a feasibility flag and a cost variable. N is the number of system tasks. In a population of a size of NP configurations, each configuration is initialised with random and feasible frequency scales (i.e., the index i of the frequency scale level F_i assigned to the task) and run in subsequent hyper-periods. System power consumption of initial configurations is recorded for each configuration in each hyper-period.

To produce the next set of candidate configurations, for each base vector in the population, three different vectors are randomly chosen. A donor vector t_v is computed from these three vectors on an element-by-element basis using a mutation formula as shown in Equation 7:

$$t_{v_i} = \text{round}(t_{v_1} + \phi \cdot (t_{v_2} - t_{v_3})); \quad (7)$$

where $i \neq i_1 \neq i_2 \neq i_3$ and Φ is the vector difference scaling factor (mutation factor) and should not be confused with frequency scaling factors. Rounding to integer is one form of discretising the continuous version of DE algorithm. A boundary check follows to constrain the frequency scales to fall within the supported processor frequency levels according to Equation 8:

$$t_{v_i} = \min(f_{max}, \max(f_{min}, t_{v_i})); \quad (8)$$

where f_{min} and f_{max} are the lowest and highest frequency scales supported by the processor, respectively. Finally, the donor vector t_{v_i} is crossed over on an element-by-element basis with its parent (base) v_i , the i^{th} vector of the population using Equation 9:

$$u_i[j] = \begin{cases} t_{v_i}[j] & \text{if } r_j > CR \\ v_i[j] & \text{otherwise} \end{cases} \quad (9)$$

where j is the j^{th} element of vectors v_i , t_{v_i} and $j \in [1, N]$. r_j is a randomly generated number for each element j , where $r_j \in [0, 1]$. CR is the crossover probability used as a control element for the differential evolution algorithm, $CR \in [0, 1]$.

Algorithm 2 DEFS

```

1: Initialization:
2: iteration  $\leftarrow$  0
3: for  $i \leftarrow 1, NP$  do ▷ NP: Population Size
4:    $v_i \leftarrow$  Random and feasible frequency scaling
     configuration
5:   Set  $v_i$  as feasible
6:   Run  $v_i$  in next HP - measure and store power
7:   iteration  $\leftarrow$  iteration + 1
8: end for
9: while iteration <  $max\_hp$  do
10:  Produce Next Generation:
11:  for  $i \leftarrow 1, NP$  do
12:    Get donor vector  $t_{v_i}$  for parent (base)  $v_i$ 
13:    Perform boundary checking and correction on  $t_{v_i}$ 
14:    Trial vector  $u_i \leftarrow$  crossover between donor vector  $t_{v_i}$  and base vector  $v_i$ 
15:    if  $u_i$  is feasible then
16:      Run trial  $u_i$  in next HP and measure power
17:      iteration  $\leftarrow$  iteration + 1
18:    else
19:      Set power of  $u_i$  to  $\infty$ 
20:    end if
21:    if  $Power(u_i) < Power(v_i)$  then
22:      Replace  $v_i$  with new candidate configuration  $u_i$ 
23:    end if
24:  end for
25: end while

```

Each of the trial vectors undergoes a schedulability check and its feasibility flag is set accordingly. If it is unfeasible, then its cost is set to ∞ ensuring that it will never replace its parent (base vector). Only feasible configurations are allowed to execute in the next hyper-period. Once the trial vector power is measured, a replacement check is conducted according to Equation 10:

$$v_i = \begin{cases} u_i & \text{if } Power(u_i) < Power(v_i) \\ v_i & \text{otherwise} \end{cases} \quad (10)$$

where $Power(u_i)$ is the trial configuration power within the hyper-period and $Power(v_i)$ is the power of its parent (base) configuration. Candidate configurations will be generated until the number of iterations hits the specified maximum or the algorithm converges.

3.4 Critical Speed – Dynamic Voltage Scaling (CS-DVS)

Each running task consumes both processor dynamic and leakage power, as well as power that is related to all associated system devices. Increasing the voltage/frequency scale of the processor leads for the task to have a shorter execution period on the expense of consuming higher dynamic power. Since the task execution time has been reduced, the leakage power and all associated devices' power is also reduced. This is due to the fact that associated devices are kept in an ON/wait state for a shorter time. The converse is equally true when the voltage/frequency scales are reduced.

As such, using the lowest frequency/voltage scales does not necessarily lead to minimum system power due to increased leakage power effects and prolonged device execution/wait time. Critical speed is defined as the speed which minimizes the overall dynamic, leakage and device power. Due to the different set of associated devices that a task uses, each task may have a different critical speed which minimises its power consumption. To find the critical speed of a

task τ_i , the power consumed by task τ_i is measured at each frequency scale F_i . The frequency scale F_i which results in the minimum task power is thus selected as the one corresponding to its critical speed.

Setting all system tasks to run at their critical speeds does not ensure feasibility, as the critical speed scales might cause system tasks to miss their deadlines. To maintain a schedulable system, it is necessary to select and run certain tasks at higher speeds than the critical speed.

Algorithm 3 shows the description of the CS-DVS algorithm which was presented in [7]. The algorithm runs in two stages. Initially, it computes the critical speed of each system task. Then, to maintain feasibility, it determines which tasks and by which factor their frequency/voltage scales need be increased while maintaining minimum power consumption. As long as the system is not in a feasible state, all tasks which are not already assigned the maximum system speed are candidates to have their frequency scales increased to the next scale. The power difference between having a task run at its current scale and the next scale is computed. The task with the minimum power consumption penalty is chosen and its scale is adjusted to the next one. The process repeats until a feasible schedule is achieved.

Algorithm 3 CS-DVS [7]

- 1: **Initialisation:**
 - 2: Compute critical speed of each task τ_i
 - 3: Set frequency scale F_i of task τ_i to that of the critical speed
 - 4: **while** (configuration not feasible) **do**
 - 5: **for** All tasks not running at max processor speed F_1 **do**
 - 6: Compute task associated power at next higher frequency scale
 - 7: Compute task power consumption difference between current
 and next higher frequency scale
 - 8: **end for**
 - 9: Choose task with lowest increase in power
 - 10: Set chosen task frequency scale F_i to F_{i-1}
 - 11: **end while**
-

3.5 Simulated Annealing (SA)

Simulated annealing is a meta-heuristic that approximates a global optimum of an NP-hard problem. The algorithm starts with a solution and explores neighbouring solutions that move towards a global optimum. A neighbouring configuration is defined as that which differs from the current configuration by one value in the set of solutions (in our case one frequency scale). A neighbouring solution that is closer to the global optimum always replaces the current solution. To avoid falling into a local minimum, a worse solution could be accepted based on a certain acceptance probability. The rationale behind this is that even though a worse solution is accepted, the neighbouring solutions of the worse solution could potentially move us toward the global optimum. In our work, we follow the adaptation of simulated annealing meta-heuristic for system energy minimization as presented in [10] with minor modifications.

Similar to our proposed algorithms, SA starts with a feasible configuration J which is a vector of size N tasks. The frequency scales assigned to the initial vector are set to those resulting from applying the CS-DVS algorithm. The initial configuration is executed in one hyper-period and its power consumption cost is recorded. One frequency scale in the current configuration J is randomly changed to another supported scale to generate a neighbouring configuration J^* . J^* is checked for feasibility. If feasible, the cost of this neighbour configuration J^* is measured and recorded in a subsequent hyper-period; otherwise, another neighbour J^* is generated from J until a feasible neighbour is found. If the newly found neighbour configuration minimizes the

system energy more than the current configuration, the optimized configuration replaces the current configuration J . However, if the neighbour configuration results in more system power consumption, it still can replace the current configuration J . For this case, a random probability $\rho \in [0, 1]$ is generated and acceptance probability α is computed according to Equation 11:

$$\alpha = e^{-\frac{Power(J) - Power(J^*)}{Power(J) * K}} \quad (11)$$

where K is the annealing factor to be decided through experimentation. If $\alpha > \rho$, then the worst solution replaces the current solution. The algorithm stops when the number of test hyper-periods is reached. The SA algorithm is listed in Algorithm 4.

Algorithm 4 Simulated Annealing

```

1: Initialisation:
2: Set initial configuration  $J$  to the output of CS-DVS [7]
3: Run  $J$  in next HP - measure and store power
4: iteration  $\leftarrow$  iteration + 1
5: Produce Next Neighbour:
6: while iteration <  $max\_hp$  do
7:   Generate neighbouring configuration  $J^*$ 
8:   if  $J^*$  is feasible then
9:     Run  $J$  in next HP - measure and store power
10:    if Power ( $J^*$ ) < Power ( $J$ ) then
11:       $J = J^*$ 
12:    else
13:      Generate random probability  $\rho$ 
14:      Compute acceptance probability  $\alpha$ 
15:      if  $\alpha > \rho$  then
16:         $J = J^*$ 
17:      end if
18:    end if
19:    iteration  $\leftarrow$  iteration + 1
20:  end if
21: end while

```

4. SIMULATION

To analyse the performance of the proposed algorithms for frequency scaling and system wide power reduction, we devised a set of experiments. We developed an event-driven simulator using SystemC 2.3.0 and TLM. Processor and device power models are consistent with previous work [10], [18]. They are based on the Intel XScale processor power profile and the device set shown in Tables 2 and 3, respectively.

Table 2. Intel XScale processor power model.

Frequency Steps (MHz)	1000	800	600	400	150
$P_{F_i}^{CPU}$ (Watt)	1.6	0.9	0.4	0.17	0.08
Voltage (V)	1.55	1.45	1.35	1.25	1.15
$E_{sw}^{CPU} = 0.5$ mJ		$t_{sw}^{CPU} = 85$ ms			

For each task set of size N , where $N \in [1 - 9]$, a total of 500 random and unique task sets are generated. The upper limit for task set size is limited by the exhaustive search time for an optimal solution. Each task τ_i is randomly assigned a unique device set, where the number of different devices per task $\in [0 - 2]$. Each device is randomly chosen from the device set shown

in Table 3. The periods of the tasks are randomly and uniformly chosen from the range of [0.5 - 100] ms. We assume that the periods of tasks in real-time systems are harmonic [10], so that they can help in reducing the simulation time. The algorithm in [19] is employed for this purpose. Tasks WCETs are randomly selected to be between 2% and 40% of the original unmodified task period. Simulations are run ten times per test case for a total of 5000 simulations for each task set size.

Table 3. Devices power model.

Device	$P_a^{d_k}$ (W)	$P_s^{d_k}$ (W)	$P_{su}^{d_k}$ (W)	$P_{sd}^{d_k}$ (W)	$t_{su}^{d_k}$ (W)	$t_{sd}^{d_k}$ (W)
Realtek Ethernet Chip	0.187	0.085	0.125	0.125	0.01	0.01
IBM Microdrive	1.3	0.1	0.5	0.5	0.12	0.12
SST Flash SST39LF020	0.125	0.001	0.05	0.05	0.001	0.001
SimpleTech Flash Card	0.225	0.02	0.1	0.1	0.002	0.002
MaxStream Wireless Module	0.75	0.005	0.1	0.1	0.04	0.04
	$P_{su}^{d_k} = E_{su}^{d_k} \times t_{su}^{d_k}$		$P_{sd}^{d_k} = E_{sd}^{d_k} \times t_{sd}^{d_k}$			

The next step is to find the best tuning parameters to initialize the meta-heuristics algorithms; namely, population size, mutation factors and crossover probabilities. Population size for both GAFS and DEFS is set to 8, 16 and 24 with an additional 32 test case for GAFS. GAFS mutation rates are set to 0.1 and 0.2. Larger mutation rates could in theory make it harder to converge as the algorithm will keep jumping between search points. Lower values could possibly lead to premature convergence and produce non-optimal results [20]. In DEFS, we chose crossover probabilities CR of 0.3, 0.5 and 0.7. We also chose the same range of mutation factors (scaling factors). We chose these values as uniform probability samples in the range [0-1]. All simulations are investigated over hyper-period sizes of [50, 100, 200 and 400]. We use two additional hyper-periods 1000 and 2500 with task sizes of 7 and 9. We assume the scheduling overhead to be low and therefore neglected.

An exhaustive search with all possible task DVFS permutations is carried out to obtain the optimal value with minimum system power. The optimal configuration serves as a reference for testing the quality of the configuration found by the algorithms under investigation. The performance of the algorithm is measured by how often the near optimal results are produced in every single case. This gives confidence in the ability of the algorithm performance to minimize system power. Finally, we simulate the CS-DVS [7] algorithm and the SA algorithm from [10] for comparison purposes.

5. RESULTS AND DISCUSSION

In this section, we report the sensitivity results of the proposed algorithms as well as the simulated annealing algorithm to the different tuning parameters. We also compare the proposed algorithms to previous work in terms of how close they are to optimal energy savings, and how much they yield better results than the well-established CS-DVS algorithm. The base energy savings from DVFS assignments are shown for the static CS-DVS and optimal search in Figure 1. These results serve as a baseline for comparing the quality of the proposed and previous algorithms.

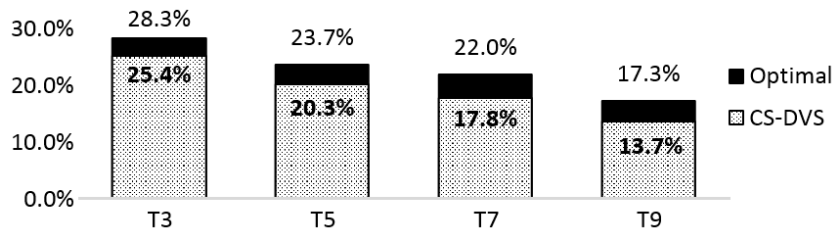


Figure 1. Average CS-DVS and optimal DVFS power savings over 500 unique sets. Tx denotes a task set with x tasks.

5.1 Sensitivity Analysis

In this sub-section, we present the performance of the proposed algorithms when their initial parameters are changed. Mainly, we vary the parameter under investigation, while the values of the other parameters remain fixed. Our experiments include varying the number of hyper-period iterations for which the algorithm is simulated. This is to analyse the convergence of the algorithms and their possible early termination effects (i.e., GAFS no longer has a pool of feasible chromosomes). We also vary mutation rates and crossover probabilities and report our findings. The results in Table 4 through Table 11 are shown for the cases where one variable is studied, while the others are fixed at the values which gave the best overall results.

The effects of running the algorithms over more hyper-periods (generations) is shown in Table 4 and Table 5 for a sample of the tasks for both GAFS and DEFS algorithms. Since the assignment space for $N = 3$ is small, an exhaustive search will always guarantee an optimal result in fewer HPs than running either GAFS or DEFS (5^3 compared to 400). The base five corresponds to the number of frequency levels supported by our model processor as shown in Table 2. However, for larger task sets, meta-heuristics deliver near-optimal energy savings in much less time (within 2500 HPs compared to 5^9 HPs for $N = 9$). The majority of the results for GAFS are near-optimal. Our observations show that setting the HP test limit to around 3% of the search space 5^N yields good results. The more hyper-periods the algorithm runs through, the better the overall results. This allows for more time for the algorithm to converge towards a near optimal solution as it is testing more chromosomes as potential solutions.

Table 4. GAFS near-optimal power savings sensitivity to hyper-period.

Tasks	Hyper-period					
	50	100	200	400	1000	2500
T3	43.3%	67.1%	92.8%	97.5%	-	-
T5	22.9%	36.2%	67.7%	84.6%	-	-
T7	13.2%	17.7%	32.8%	57.5%	70.1%	78%
T9	13.9%	17%	27%	46.9%	68.6%	85.3%
GAFS population size = 32						

Table 5. DEFS near-optimal power savings sensitivity to hyper-period.

Tasks	Hyper-period					
	50	100	200	400	1000	2500
T3	35.8%	42.2%	69.4%	93.6%	-	-
T5	19.5%	22.6%	33.4%	67.1%	-	-
T7	12.2%	13.1%	17.1%	36%	70.9%	82.8%
T9	12.5%	13.1%	16.5%	28.8%	62.4%	86.5%
DEFS population size = 24, CR = 0.3 and $\varphi = 0.5$						

The next step is conducting analysis on varying the initial population size. That is, we change the number of initial feasible chromosomes for the GAFS algorithm, as well as the initial candidate vector pool for DEFS. In Table 6, we observe that larger population sizes in GAFS yield better results with wider margins for larger task sets. Larger population sizes make it possible to have more pairings and crossover possibilities that in turn allow for better exploration of the search space. A population size of eight only has two pairs of parents to generate an offspring compared to 16 pairs in a population of 32. The difference between a population size of 24 and 32 is insignificant in most of the cases.

Table 6. GAFS near-optimal power savings sensitivity to population size, $\mu = 0.1$.

Population size		8	16	24	32
Task No.	HP	Percentage of near-optimal power saving configurations			
T3	400	95.9%	96.9%	96.9%	97.5%
T5		78.3%	82.1%	84.6%	84.6%
T7	2500	67.7%	75.3%	78.0%	78.0%
T9		77.1%	82.4%	84.6%	85.3%
Task No.	HP	Percentage of configurations better than CS-DVS			
T3	400	94.9%	95.0%	95.0%	95.0%
T5		94.8%	96.7%	97.6%	98.0%
T7	2500	95.8%	97.5%	98.2%	98.0%
T9		97.4%	98.3%	98.4%	98.7%

In GAFS, we limited our study of mutation rates to rates of $\mu = 0.1$ and $\mu = 0.2$. This is to keep by the suggestion in [20] of using low mutation rates to ensure algorithm convergence. Table 7 shows power savings sensitivity to GAFS mutation rates and task set size when the population size is fixed at 32. As seen from the table, we observe that lower $\mu = 0.1$ gives overall better results and only in few cases $\mu = 0.2$ results in marginal gains. Given the population size and the chromosome size of our problem, lower mutation rates were expected to give better results. Higher mutation rates would entail exploring further away from our current best results. As μ increases, the closer the genetic algorithm gets to a random search. GAFS outperforms CS-DVS in most cases, especially with larger task set sizes.

Table 7. GAFS near-optimal power savings sensitivity to mutation factors, population size = 32.

Task No.	HP	Percentage of near-optimal power saving configurations		Percentage of operations better than CS-DVS	
		$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.1$	$\mu = 0.2$
T3	400	97.5%	96.7%	95.0%	94.9%
T5		84.6%	78.6%	98.0%	95.7%
T7	2500	78.0%	81.8%	98.0%	99.0%
T9		85.3%	73.2%	98.7%	97.3%

Table 8 shows the results of varying the population size for the DEFS algorithm when $CR = 0.3$ and the mutation rate (scaling factor) $\phi = 0.5$. The reported results are best when each of the values of CR is tuned over the set $[0.3, 0.5, 0.7]$. We see that a population size of 16 provides better results for smaller task sets; whereas a population size of 24 gives better results for larger task sets. Similar to GAFS, large population sizes allow for richer selection of candidate vectors, as well as for more variance in the crossover and mutation operations. Since larger task sizes entail larger dimensions, larger initial population is expected to achieve convergence compared to smaller task sizes.

Table 8. DEFS near-optimal power savings sensitivity to population size, CR = 0.3 and $\varphi = 0.5$.

Task No.	HP	Percentage of near-optimal power saving configurations			Percentage of configurations better than CS-DVS		
		Population Size			Population Size		
		8	16	24	8	16	24
T3	400	90.8%	95.8%	93.6%	91.0%	94.6%	94.2%
T5		71.2%	80.4%	67.1%	90.5%	96.7%	94.6%
T7	2500	34.3%	77.9%	82.8%	86.8%	97.4%	98.8%
T9		40.6%	81.4%	86.5%	88.1%	98%	98.6%

The sensitivity analysis findings of DEFS crossover probability (CR) and mutation (scaling) factor φ parameters is summarized in Table 9. For larger task sizes of 7 and 9, we find that crossover probability and mutation factor carry no statistical differences in yielding better results across different combinations of CR and φ . However, for smaller task sizes, a CR of 0.3 and $\varphi = 0.5$ provide better results by a wide margin (i.e., up to 14% better results than those at CR = 0.7 and $\varphi = 0.7$ for a system with five tasks).

Table 9. DEFS sensitivity to crossover probability (CR) and mutation factor φ at best results of fixed population size and hyper-period.

Population size		16		24	
Hyper-period		400		2500	
CR	φ	T3	T5	T7	T8
0.3	0.3	94.4%	76.1%	77.1%	81.8%
	0.5	95.8%	80.4%	82.8%	86.5%
	0.7	95.3%	78.2%	83.3%	85.6%
0.5	0.3	92.9%	75.2%	80.7%	84.9%
	0.5	94.3%	78.2%	84%	87%
	0.7	93.7%	76.3%	83.7%	87.0%
0.7	0.3	87.9%	68.0%	80.1%	83.8%
	0.5	89.7%	69.2%	82.5%	84.8%
	0.7	87.6%	66.4%	82.6%	83.9%

Table 10. Simulated Annealing (SA) near-optimal power savings sensitivity to hyper-period.

Tasks	Hyper-period					
	50	100	200	400	1000	2500
T3	36.8%	43.5%	51.5%	58.8%	-	-
T5	21.5%	23.3%	26.5%	32.3%	-	-
T7	13.9%	14.7%	15.6%	17.3%	19.6%	31.5%
T9	20.6%	23.0%	25.3%	27.9%	17.3%	25.6%

Table 11. Simulated Annealing (SA) percentage of configurations better than CS-DVS.

Tasks	Hyper-period					
	50	100	200	400	1000	2500
T3	63.3%	70.4%	74.9%	75.7%	-	-
T5	54.7%	60.8%	65.9%	72.8%	-	-
T7	48.1%	54.4%	59.8%	66.5%	70.3%	78.6%
T9	40.5%	44.9%	48.1%	52.0%	63.4%	72.6%

The reference values for the simulated annealing (SA) algorithm implementation are summarized in Table 10 and Table 11. One major observation is that the results of the SA algorithm do not provide substantial gains as the number of hyper-periods is increased when it comes to near-optimal results. This is more obvious at the larger task set size of T9. In fact, lower number of hyper-periods could provide better results. This is due to the algorithm design as implemented by [10], where even though the algorithm can escape a local minimum, there is no guarantee that it will converge to a better solution and the best-yet found values are not preserved.

5.2 Algorithm Comparison and Discussion

Figure 2 provides a performance summary of the proposed and reference algorithms. We observe that the proposed algorithms outperform the SA algorithm in terms of their ability to consistently provide near-optimal power savings. As the task set size increases, the quality performance of SA decreases, while the proposed algorithms consistently maintain their quality performance. Figure 3 shows that the SA algorithm fails 25% of the time to yield quality configurations better than CS-DVS. In effect, this could lead to high system-wide energy consumption. Both GAFS and DEFS are superior to SA, as they almost always deliver better power configurations than CS-DVS.

GAFS slightly outperforms DEFS for smaller task sets and the converse is true for larger task sets. The weakness point of GAFS is that a new generation of test quality configurations can only be generated when the current population has been fully examined. DEFS does not suffer from this issue, as candidate test configurations are generated randomly from a list of the so-far best found quality configurations which are readily available. The SA algorithm suffers from the possibility of replacing an elite solution by a non-optimal one. This is due to the inherent design of the algorithm, where it stochastically accepts worse solutions as means to escape a local minimum.

Finally, even though GAFS maintains an elite solution through the generations (which only gets updated if better solution is found), GAFS suffers from the possibility of producing a whole generation of non-feasible solutions aside from the elite. DEFS, on the other hand, does not suffer from these issues as it maintains a population of best found feasible solutions at any time.

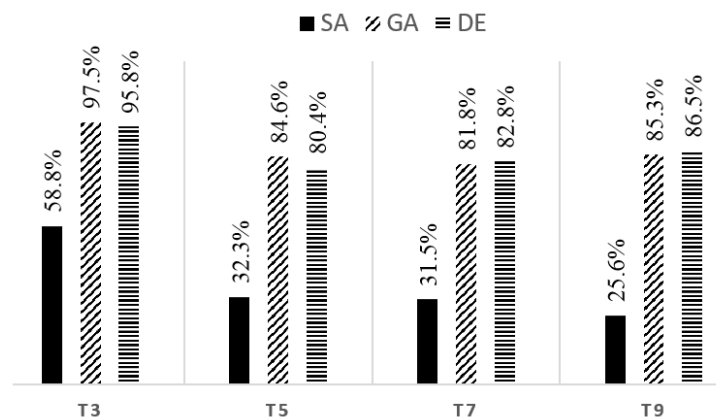


Figure 2. Percentage of near-optimal results of the three meta-heuristics over 500 unique sets.

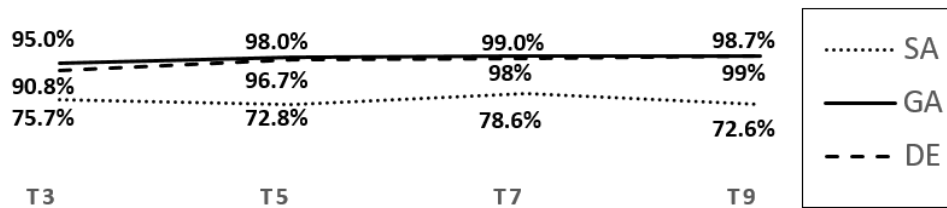


Figure 3. Percentage of the three meta-heuristics that are better than the CS-DVS heuristic over 500 unique sets.

Furthermore, since each configuration is randomly produced from this set based on mutation probabilities, even if a pass generates a set of unfeasible power configurations, new feasible power configurations can still be produced in subsequent hyper-periods.

6. CONCLUSIONS

System wide energy minimization is of paramount importance in modern embedded system design. We specifically adapted the use of genetic (GA) and evolutionary (DE) algorithms with the goal of reducing the overall energy consumption. We have investigated the performance of our developed meta-heuristic algorithms that assign frequency scaling (DVFS) to tasks in a hard-real-time system. We measure energy consumption at the system level; that is that of the processor and the devices. We have conducted a sensitivity analysis over a wide range of initial values of the proposed algorithms. We have found that setting the algorithm search space to 12% of the available search space for small task sizes (i.e., T5) yields a majority of near-optimal results. A much smaller search space of < 3% of the total exploration space works well for larger task sizes of 7 and 9. For the genetic algorithm, an initial chromosome set of 32 performs better than all other initial set sizes of 8, 16 or 24. There are marginal result differences in changing the mutation factor μ from 0.1 to 0.2. In DE, the size of the initial vector set affects smaller task sets differently than larger task ones. An initial size of 24 vectors provides a majority of near-optimal results for task set size of 7 and 9. The same results are obtained for smaller task sets of 3 and 5 with a smaller initial vector set size of 16. Finally, a crossover probability of 0.3 and mutation (scaling) factor $\varphi = 0.5$ provide the best overall results regardless of the task set size.

The proposed algorithms outperformed the simulating annealing (SA) algorithm by an approximate factor of 2.75 to 1 for finding a near-optimal configuration when the system task set is comprised of 5 to 9 tasks. Furthermore, based on 500 unique sets of tasks, our proposed algorithms deliver near-optimal results in over 95% of the cases compared to the CS-DVS algorithm. Simulated annealing is better than CS-DVS by an average of 75% of the time. The proposed techniques we put forth have allowed for additional energy optimizations, which is favourable for the quest in the design of low power embedded systems.

REFERENCES

- [1] P. Pillai and K. G. Shin, "Real-time Dynamic Voltage Scaling for Low-power Embedded Operating Systems," in Proc. of the 18th ACM Symp. on Operating Systems Principles (SOSP '01), New York, NY, USA, pp. 89-102, 2001.
- [2] S. Saewong and R. Rajkumar, "Practical Voltage-scaling for Fixed-priority RT-Systems," in the 9th IEEE Proc. on Real-Time and Embedded Technology and Applications, pp. 106-114, 2003.
- [3] L. Benini et al. "A Survey of Design Techniques for System-level Dynamic Power Management," Proc. in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 3: pp. 299-316, 2000.
- [4] H. Cheng and S. Goddard, "Online Energy-aware I/O Device Scheduling for Hard Real-time

- Systems," in Proc. of Design, Automation and Test in Europe (DATE '06), volume 1, pp. 1055-1060, Munich, 6-10 March 2006.
- [5] V. Swaminathan and K. Chakrabarty, "Pruning-based Energy-optimal Device Scheduling for Hard Real-time Systems," in Proc. of the 10th International Symp. on Hardware/Software Co-design (CODES 2002), pp. 175–180, 2002.
- [6] V. Devadas and H. Aydin, "On the Interplay of Voltage/frequency Scaling and Device Power Management for Frame-based Real-time Embedded Applications," Proc. of the IEEE Transaction on Comput., vol. 61, no. 1, pp. 31–44, 2012.
- [7] R. Jejurikar and R. Gupta, "Dynamic Voltage Scaling for System Wide Energy Minimization in Real-time Embedded Systems," Proc. of the 2004 International Symp. on Low Power Electronics and Design (ISLPED '04), pp. 78–81, 2004.
- [8] W. Wang et al. "System-wide Energy Optimization with DVS and DCR," Proc. of Dynamic Reconfiguration in Real-Time Systems, no. 4, pp. 129–163, Springer, New York, 2013.
- [9] F. Kong et al. "Minimizing Multi-resource Energy for Real-time Systems with Discrete Operation Modes," in Proc. of the 22nd Euromicro Conference on Real-Time Systems (ECRTS '10), Washington, DC, USA, pp. 113–122, 2010
- [10] D. He and W. Mueller, "Online Energy-efficient Hard Real-time Scheduling for Component Oriented Systems," Proc. of the IEEE 15th International Symp. on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), pp. 56–63, 2012.
- [11] V. Devadas and H. Aydin, "DFR-EDF: A Unified Energy Management Framework for Real-time Systems," Proc. of the 16th IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS), pp. 121–130, 2010.
- [12] L. Niu, "System-level Energy-efficient Scheduling for Hard Real-time Embedded Systems," in Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1–4, 2011.
- [13] B. A. Mahafzah and B. A. Jaradat, "The Hybrid Dynamic Parallel Scheduling Algorithm for Load Balancing on Chained-cubic Tree Interconnection Networks," in Journal of Supercomputing, vol. 52, no. 3, pp.224–252, 2010.
- [14] J. Zhao and H. Qiu, "Genetic Algorithm and Ant Colony Algorithm Based Energy-efficient Task Scheduling," in International Conference on Inform. Science and Technology (ICIST), pp. 946–950, 2013.
- [15] S. G. Ahmad et al. "PEGA: A Performance Effective Genetic Algorithm for Task Scheduling in Heterogeneous Systems," in IEEE 14th Intl. Conference on High Performance Computing and Commun. IEEE 9th Intl. Conference on Embedded Software and Systems (HPCC-ICISS), pp. 1082–1087, 2012.
- [16] M. A Alshraideh et al. "Using Genetic Algorithm as Test Data Generator for Stored pl/sql Program Units," in Journal of Software Eng. and Applicat, vol. 6, no. 2, p. 65, 2013.
- [17] D. Simon, Evolutionary Optimization Algorithms, Wiley, 2013.
- [18] G. Chen et al. "Effective Online Power Management with Adaptive Interplay of DVS and DPM for Embedded Real-time System," in Euromicro Conference on Digital System Design (DSD), pp. 881–889, 2013.
- [19] J. Xu, "A Method for Adjusting the Periods of Periodic Processes to Reduce the Least Common Multiple of the Period Lengths in Real-time Embedded Systems," in IEEE/ASME Intl. Conference on Mechatronics and Embedded Syst. and Applicat. (MESA), pp. 288–294, 2010.
- [20] R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms, Wiley InterScience Electronic Collection, Wiley, 2004.
- [21] R. Jejurikar and R. Gupta, "Optimized Slowdown in Real-time Task Systems," Proc. in IEEE Computer Trans., vol. 55, no. 12, pp. 1588–1598, 2006.

- [22] B. A. Mahafzah and B. A. Jaradat, "The Load Balancing Problem in OTIS-Hypercube Interconnection Networks," in Journal of Supercomputing, vol. 46, no. 3, pp. 276-297, 2008.
- [23] V. Devadas and H. Aydin, "On the Interplay of Voltage/Frequency Scaling and Device Power Management for Frame-Based Real-Time Embedded Applications," Proc. in IEEE Transactions on Computers, vol. 61, no. 1, pp. 31-44, 2012.

ملخص البحث:

تصاعدت أهمية أنظمة الزمن الحقيقي الفعّالة من حيث الطاقة كثيراً في السنوات القليلة الماضية. فقد تم تطوير تقنيات على جميع مستويات تصميم الأنظمة للتقليل من استهلاك الطاقة. فعلى المستوى المادي، تحاول تقنيات التصنيع التقليل من الطاقة التي تستهلكها مجموعة الرقائق على وجه الإجمال. وعلى مستوى تصميم النظام، تسمح تقنيات مثل التدرج الديناميكي للفولتية والتردد (DVFS) الإدارة الديناميكية للطاقة (DPM) بتغيير تردد المعالج على نحو فوريّ أو الانتقال إلى أنماط "النوم" لتقليل القدرة التشغيلية إلى أدنى حدّ ممكن. وعلى مستوى أنظمة التشغيل، تستفيد الجدولة الفعّالة من حيث الطاقة من خلال البحوث والدراسات التي أجريت في هذا المجال من التدرج الديناميكي للفولتية والتردد، ومن الإدارة الديناميكية للطاقة على مستوى المهام لتحقيق وفورات إضافية في الطاقة. وقد تركز معظم جهود البحث في مجال الجدولة الفعّالة من حيث الطاقة على تقليل قدرة المعالج. وحديثاً، تم استنقضاء حلول على مستوى النظام بمفهومه الواسع. في هذه الورقة البحثية، نتوسع في أعمال سابقة عبر تكييف خوارزميتين تطوّريتين لتقليل طاقة النظام ككل إلى أدنى حدّ ممكن، ونحلّل أداء خوارزمياتنا تحت ظروف ابتدائية متغيرة. إضافة إلى ذلك، نبيّن أن تقنياتنا وراء الموجهة تعطي وفورات في الطاقة أقرب إلى الوضع المثالي لما نسبته ٨٥% من الوقت مقارنةً بنسبة ٣٠% تقريباً، التي تم تحقيقها باستخدام التقوية بالمحاكاة لما يزيد على ٥٠٠ مجموعة فريدة من الفحوصات.

كذلك، تُبين نتائجنا أنّ ما يزيد على ٩٥% من الحالات، تُعطي التقنيات ما وراء الموجهة وفورات أكثر في الطاقة المستهلكة مقارنةً بالطريقة الستاتيكية (CS-DVS).



المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

العدد ١

المجلد ٢

نيسان ٢٠١٦

JJCIT

www.jjcit.org

jjcit@psut.edu.jo

مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي