

**SIMULATED KALMAN FILTER
ALGORITHMS FOR ADAPTIVE
BEAMFORMING**



UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : _____

Date of Birth : _____

Title : _____

Academic Session : _____

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserve the right as follows:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

(Supervisor's Signature)

New IC/Passport Number
Date:

Name of Supervisor
Date:

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter page 2 from the organization with the period and reasons for confidentiality or restriction.

SUPERVISOR'S DECLARATION

We hereby declare that We have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science.

(Supervisor's Signature)

Full Name : DR. NURUL HAZLINA BINTI NOORDIN

Position : LECTURER

Date : 21 DECEMBER 2017

(Co-supervisor's Signature)

Full Name : ASSOC. PROF. DR. ZUWAIRIE BIN IBRAHIM

Position : LECTURER

Date : 21 DECEMBER 2017

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : KELVIN LAZARUS A/L LAZARUS

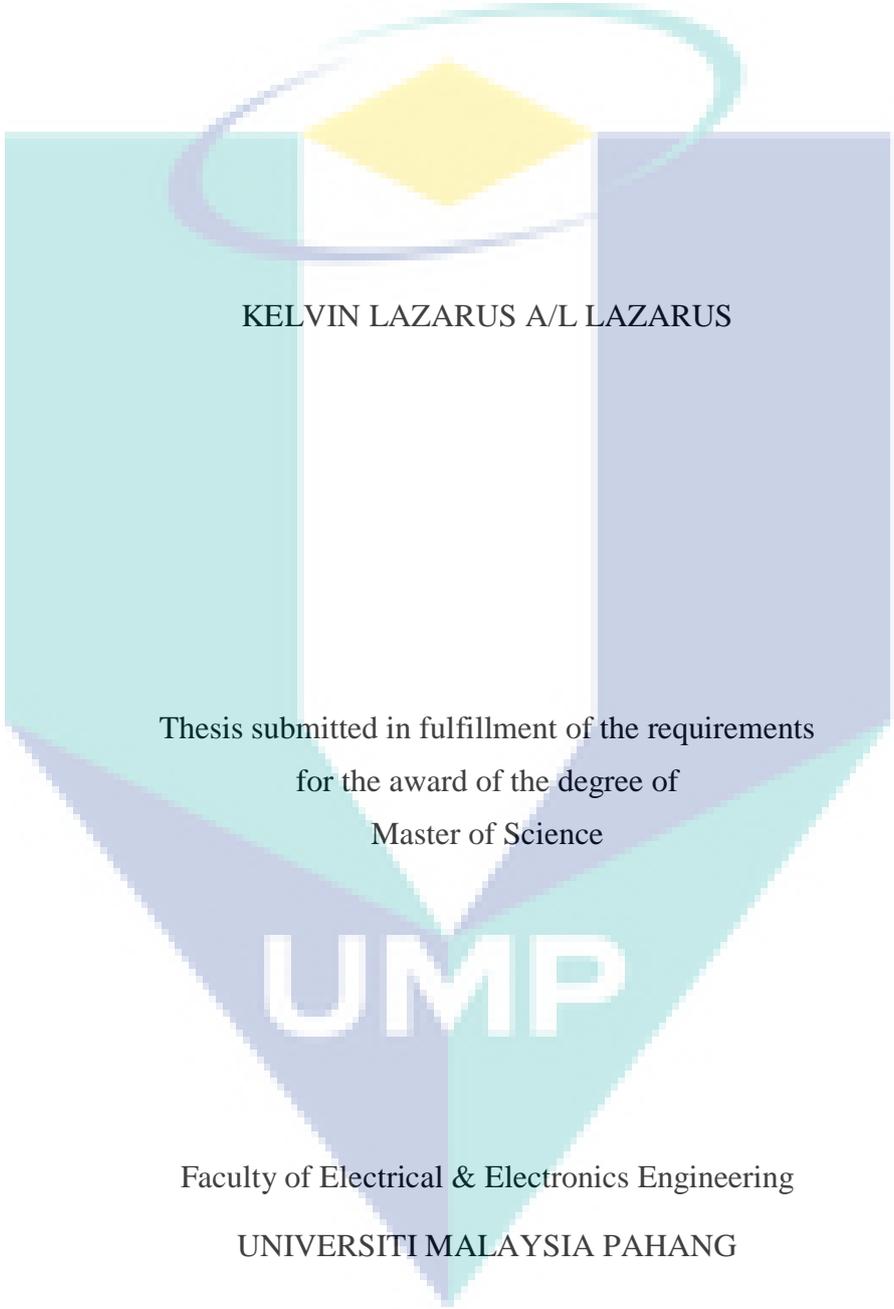
ID Number : MEL15005

Date : 21 December 2017



UMP

ADAPTIVE BEAMFORMING ALGORITHM BASED ON SIMULATED
KALMAN FILTER



KELVIN LAZARUS A/L LAZARUS

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Master of Science

UMP

Faculty of Electrical & Electronics Engineering

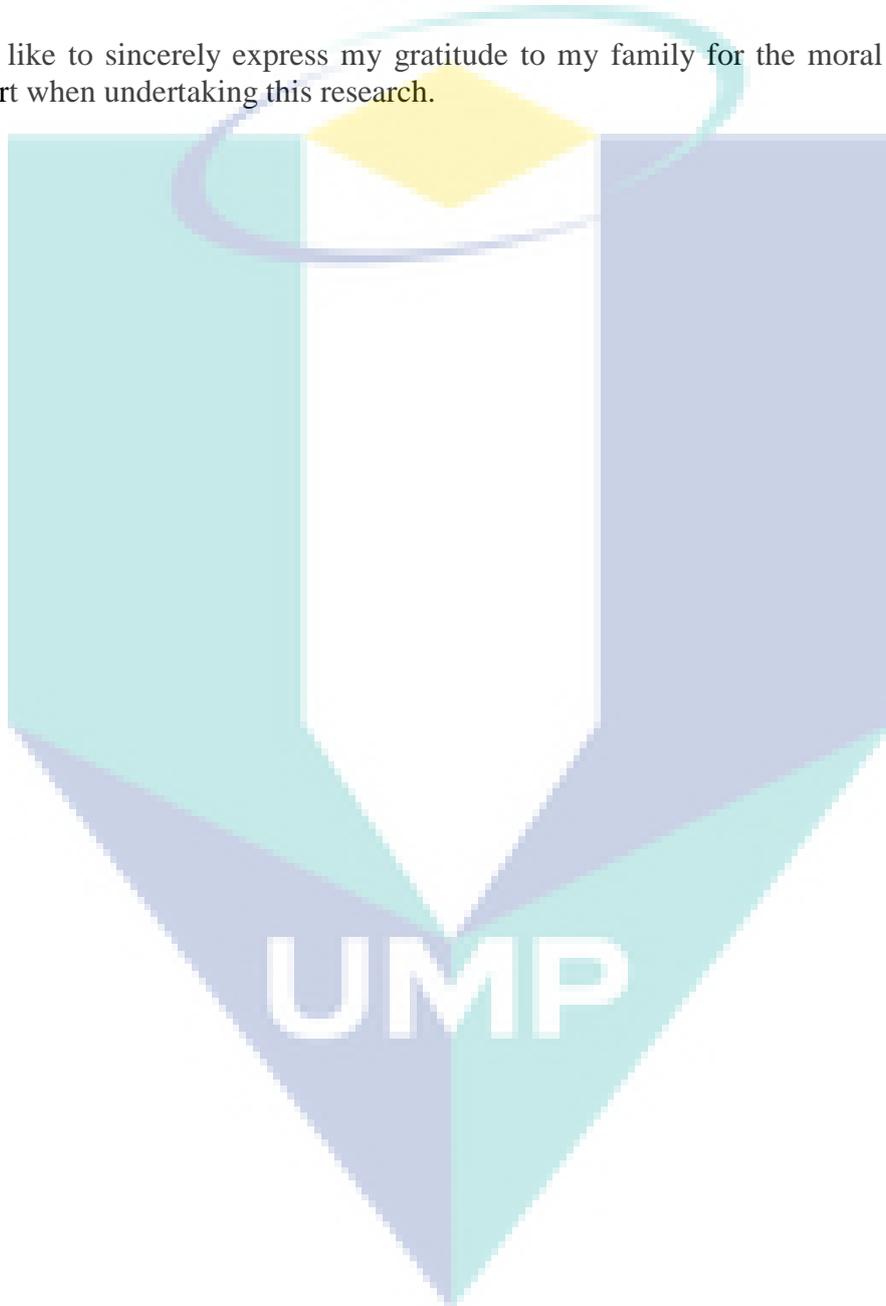
UNIVERSITI MALAYSIA PAHANG

DECEMBER 2017

ACKNOWLEDGEMENT

I would like to take this opportunity to sincerely express my highest gratitude to my supervisor, Dr. Nurul Hazlina and Associate Professor Dr. Zuwairie Ibrahim for the guidance and the help given during the progress of the project. The help and guidance is the sole reason for this research to achieve success.

I also like to sincerely express my gratitude to my family for the moral and spiritual support when undertaking this research.



ABSTRAK

Adaptive beamforming adalah teknik yang digunakan untuk mengarahkan pola sinaran ke arah isyarat yang diinginkan dan batalkan isyarat yang tidak diinginkan dengan mencari medan total (magnitud dan fasa) yang sesuai untuk setiap elemen dalam susunan antenna, untuk mencapai nilai nisbah isyarat dikehendaki kepada isyarat tidak dikehendaki (SINR) yang tinggi. Terdapat banyak kaedah untuk melakukan *adaptive beamforming* dan salah satu cara ialah menggunakan algoritma metaheuristik, untuk menganggarkan medan total bagi elemen dalam susunan antenna. Pelbagai jenis algoritma metaheuristik telah digunakan untuk *adaptive beamforming*. Ada algoritma metaheuristik yang telah diubahsuai dari algoritma asal untuk meningkatkan prestasi algoritma dalam aplikasi *adaptive beamforming*. Algoritma metaheuristik baru bernama *Simulated Kalman Filter* (SKF), diperkenalkan melalui inspirasi daripada keupayaan anggaran *Kalman Filter*, tidak pernah digunakan untuk aplikasi *adaptive beamforming*. Oleh itu, kajian ini membentangkan aplikasi pertama algoritma SKF untuk *adaptive beamforming*. Walau bagaimanapun, algoritma SKF sering menumpuk secara awal pada *local optimum* kerana kekurangan penerokaan untuk mencari penyelesaian yang lebih baik. Versi algoritma SKF lain, yang dinamakan *Opposition-Based SKF*, diperkenalkan oleh K. Zakwan, menggunakan kaedah *Opposition-Based Learning* (OBL) untuk meningkatkan keupayaan penerokaan algoritma SKF. Selain itu, Versi algoritma SKF lain, yang dinamakan *SKF with Modified Measurement* (SKFMM) diperkenalkan untuk meningkatkan keupayaan penerokaan algoritma SKF dengan mengubahsuai *measurement-update* dalam algoritma SKF. Algoritma SKF, OBSKF dan SKFMM diaplikasikan kepada susunan antenna 10 elemen dengan jarak 0.5λ antara elemen. Sudut isyarat yang dikehendaki ditetapkan pada 30° dan sudut isyarat gangguan ditetapkan pada $-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ$. Eksperimen diulang sebanyak 100 kali untuk beberapa input nilai nisbah isyarat dikehendaki kepada isyarat hingar (SNR) untuk mendapat nilai statistic maksimum, minimum, min dan sisihan piawai bagi nilai nisbah isyarat dikehendaki kepada isyarat tidak dikehendaki (SINR). Hasil daripada kajian SKF, OBSKF dan SKFMM pada *adaptive beamforming* dibandingkan dengan kajian yang diterbitkan sebelumnya, iaitu, *Adaptive Mutated Boolean Particle Swarm Optimization* (AMBPSO). Hasil eksperimen menunjukkan bahawa ketiga-tiga algoritma SKF dapat menghasilkan min SINR yang lebih tinggi dan nilai sisihan piawai yang lebih rendah. Nilai min SINR yang tinggi dan nilai sisihan piawai yang rendah membuktikan bahawa ketiga-tiga algoritma SKF adalah konsisten dalam mencari penyelesaian yang bagus. Kesemua algoritma SKF menghasilkan kosistensi lebih daripada 70 % berbanding dengan algoritma AMBPSO dalam aplikasi *adaptive beamforming*. Antara ketiga-tiga algoritma SKF, SKFMM dapat menghasilkan nilai min SINR yang tertinggi dan juga yang paling konsisten. Algoritma SKFMM lebih konsisten berbanding dengan algoritma SKF sebanyak 25.20 % dan SKFMM lebih konsisten berbanding dengan algoritma OBSKF sebanyak 17.50 %.

ABSTRACT

Adaptive beamforming is a technique used to steer the radiation pattern towards the desired signal and cancel out any interference signal by finding the appropriate weights for every element in an array antenna, to achieve maximum signal to interference plus noise ratio (SINR). There are many methods to perform adaptive beamforming and one of the method is to use metaheuristic algorithm, to estimate the weights for individual elements in an array. Over the years, various metaheuristic algorithms have been applied to adaptive beamforming. Some of the metaheuristic algorithms have been modified from the original algorithms to improve the algorithms performance in adaptive beamforming application. A new metaheuristic algorithm named Simulated Kalman Filter (SKF), is inspired by the estimation capabilities of Kalman filter, has not been applied to adaptive beamforming application. Therefore, this research presents the first-time application of SKF algorithm to adaptive beamforming. The SKF algorithm, however, often converge prematurely at local optimum due to lack of exploration, preventing it from finding better solution. A modified version of the SKF algorithm, named Opposition-Based SKF (OBSKF), introduced by K. Zakwan, applies Opposition-Based Learning method to improve the exploration capabilities of SKF algorithm. Moreover, a new modified version of the SKF algorithm named SKF with Modified Measurement (SKFMM) is introduced to further improve the exploration capabilities of SKF algorithm by modifying the measurement-update equation. The SKF, OBSKF and SKFMM is applied to an array antenna with 10 elements arranged linearly with 0.5λ distance between elements. The desired signal angle is set 30° and the interference signal angle is set to $-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ$. The experiment is repeated for 100 times for various signal to noise ratio (SNR) values to obtain statistical results for best, worst, mean and standard deviation of the signal to interference plus noise ratio (SINR). The results obtained using SKF, OBSKF and SKFMM is compared to previously published work, Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO). The results show that all three SKF algorithms can produce higher mean SINR and lower standard deviation values compared to AMBPSO. The high mean SINR and low standard deviation value proves that the SKF algorithms are accurate and consistent in finding better solution. All the SKF algorithms produces consistency above 70% compared to existing AMBPSO for adaptive beamforming. Among the three SKF algorithms, the SKFMM produces the highest mean SINR values and is also the most consistent. The SKFMM is more consistent than the SKF algorithm by 25.20% and the SKFMM is more consistent than the OBSKF algorithm by 17.50%.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	i
ACKNOWLEDGEMENT	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	ix
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Problem Statement	3
1.3 Research Motivation	4
1.4 Research Objective	4
1.5 Research Scope	4
1.6 Research Contributions	5
1.7 Thesis Organization	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Beamforming Techniques	7
2.3 Statistically Optimum Beamforming	8

2.3.1	Maximum Signal to Interference Plus Noise Ratio	8
2.3.2	Minimum Mean Square Error	9
2.3.3	Linear Constrained Minimum Variance	9
2.3.4	Minimum Variance Distortionless Response	9
2.4	Adaptive Algorithms for Beamforming	10
2.4.1	Least Mean-Square	10
2.4.2	Recursive Least Squares	11
2.4.3	Constant Modulus	11
2.5	Beamforming Using Metaheuristic Algorithm	11
2.5.1	Particle Swarm Optimization	13
2.5.2	Gravitational Search Algorithm	13
2.5.3	Firefly Algorithm	15
2.5.4	Grey Wolf Optimization	15
2.5.5	Adaptive Social Behaviour Optimization	16
2.5.6	Backtracking Search Algorithm	16
2.5.7	Mean Variance Mapping Optimization	17
2.5.8	Differential Search Algorithm	17
2.5.9	Cat Swarm Optimization	17
2.5.10	Taguchi's Optimization Method	18
2.5.11	Modified Cuckoo Search Algorithm	19
2.5.12	Harmony Search Algorithm	19
2.5.13	Self-Adaptive Differential Evolution	19
2.5.14	Bees Algorithm	20
2.5.15	Adaptive Invasive Weed Optimization	20
2.5.16	Dynamic Mutated Artificial Immune System	21
2.6	Simulated Kalman Filter	21

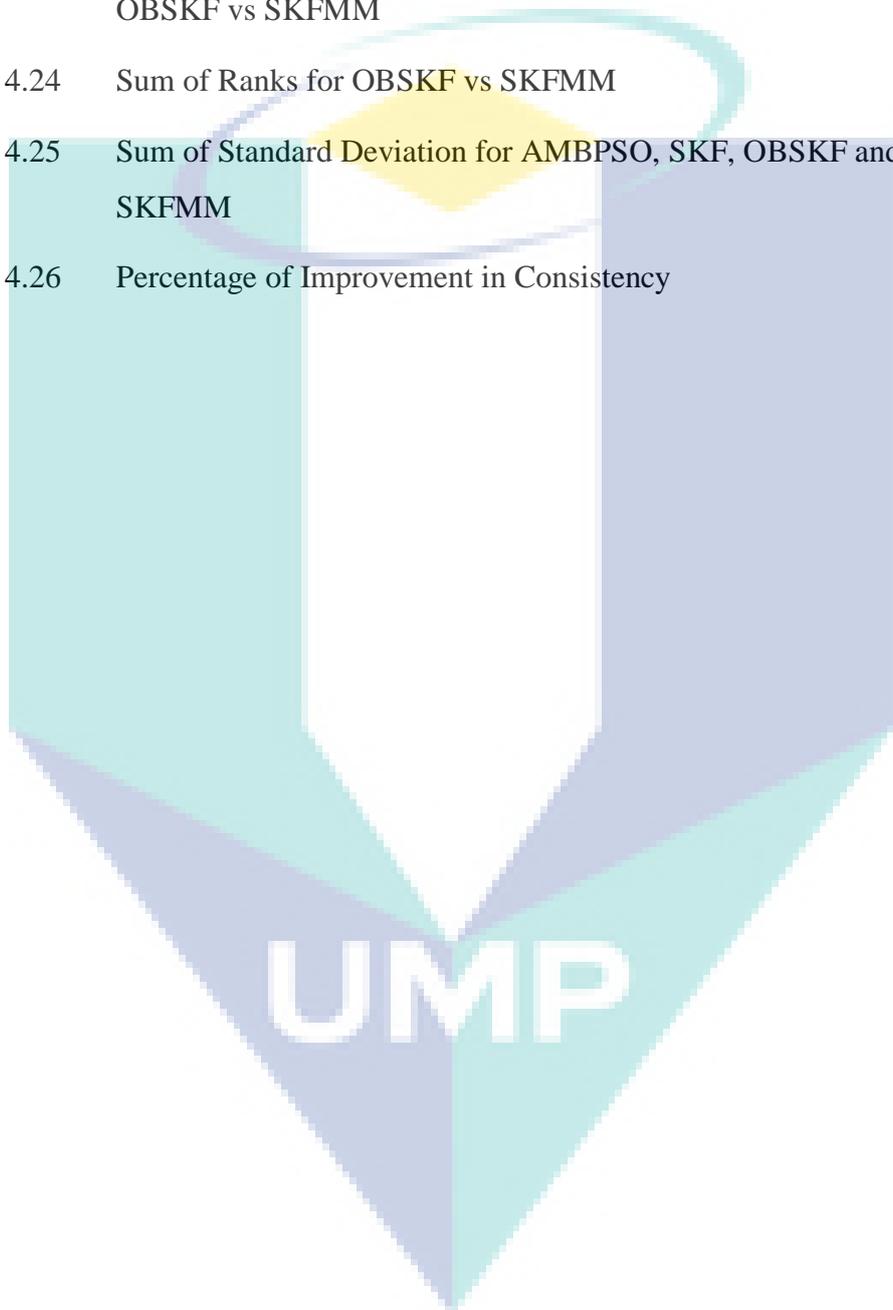
2.7	Opposition-Based Simulated Kalman Filter	23
2.8	Adaptive Mutated Boolean Particle Swarm Optimization	23
2.9	Research Gap Analysis	24
2.10	Summary	25
CHAPTER 3 METHODOLOGY		27
3.1	Introduction	27
3.2	Kalman Filter	27
3.3	Simulated Kalman Filter	29
3.4	Opposition-Based Learning	32
3.5	Opposition Based Simulated Kalman Filter	33
3.6	Simulated Kalman Filter with Modified Measurement	35
3.7	Array System Model and Fitness Function Formulation	38
3.8	Experimental Setup	41
3.8.1	Application of SKF to Adaptive Beamforming	45
3.8.2	Application of OBSKF to Adaptive Beamforming	47
3.8.3	Application of SKFMM to Adaptive Beamforming	50
3.9	Summary	51
CHAPTER 4 RESULT & DISCUSSION		52
4.1	Introduction	52
4.2	Radiation Pattern	52
4.2.1	Signal to Noise Ratio, <i>SNR = 15dB</i>	53
4.2.2	Signal to Noise Ratio, <i>SNR = 30dB</i>	54
4.2.3	Signal to Noise Ratio, <i>SNR = 50dB</i>	56
4.3	Convergence Curve	58

4.3.1	Signal to Noise Ratio, <i>SNR = 15dB</i>	58
4.3.2	Signal to Noise Ratio, <i>SNR = 30dB</i>	60
4.3.3	Signal to Noise Ratio, <i>SNR = 50dB</i>	61
4.4	Statistical Results	62
4.4.1	AMBPSO vs SKF	62
4.4.2	AMBPSO vs OBSKF	64
4.4.3	AMBPSO vs SKFMM	65
4.4.4	SKF vs OBSKF	67
4.4.5	SKF vs SKFMM	69
4.4.6	OBSKF vs SKFMM	71
4.5	Discussion	73
4.6	Summary	78
CHAPTER 5 CONCLUSION AND RECOMMENDATIONS		80
5.1	Conclusion	80
5.2	Recommendations	80
5.3	Significance of Research	81
REFERENCES		83
APPENDIX A Critical Values of T_0 in Wilcoxon Signed Rank Test		90
APPENDIX B List of publication		91

LIST OF TABLES

Table 3.1	Parameters of SKF, OBSKF and SKFMM	45
Table 4.1	Null Depth (dB) for $SNR = 15dB$	53
Table 4.2	Accuracy of the Main Beam for $SNR = 15dB$	54
Table 4.3	Maximum Sidelobe Level for $SNR = 15dB$	54
Table 4.4	Null Depth (dB) for $SNR = 30dB$	55
Table 4.5	Accuracy of Main Beam for $SNR = 30dB$	56
Table 4.6	Maximum Sidelobe Level for $SNR = 30dB$	56
Table 4.7	Null Depth (dB) for $SNR = 50dB$	57
Table 4.8	Accuracy of Main Beam for $SNR = 50dB$	58
Table 4.9	Maximum Sidelobe Level for $SNR = 50dB$	58
Table 4.10	Maximum SINR Value After 10000 Iteration for $SNR = 15dB$	59
Table 4.11	Maximum SINR Value After 10000 Iteration for $SNR = 30dB$	60
Table 4.12	Maximum SINR Value After 10000 Iteration for $SNR = 50dB$	61
Table 4.13	Best, Worst, Mean and Standard Deviation(STD) of SINR(dB) for AMBPSO vs SKF	62
Table 4.14	Sum of Ranks for AMBPSO vs SKF	63
Table 4.15	Best, Worst, Mean and Standard Deviation(STD) of SINR(dB) for AMBPSO vs OBSKF	64
Table 4.16	Sum of Ranks for AMBPSO vs OBSKF	65
Table 4.17	Best, Worst, Mean and Standard Deviation(STD) of SINR(dB) for AMBPSO vs SKFMM	66
Table 4.18	Sum of Ranks for AMBPSO vs SKFMM	67
Table 4.19	Best, Worst, Mean and Standard Deviation(STD) of SINR for SKF vs OBSKF	68
Table 4.20	Sum of Ranks for SKF vs OBSKF	69

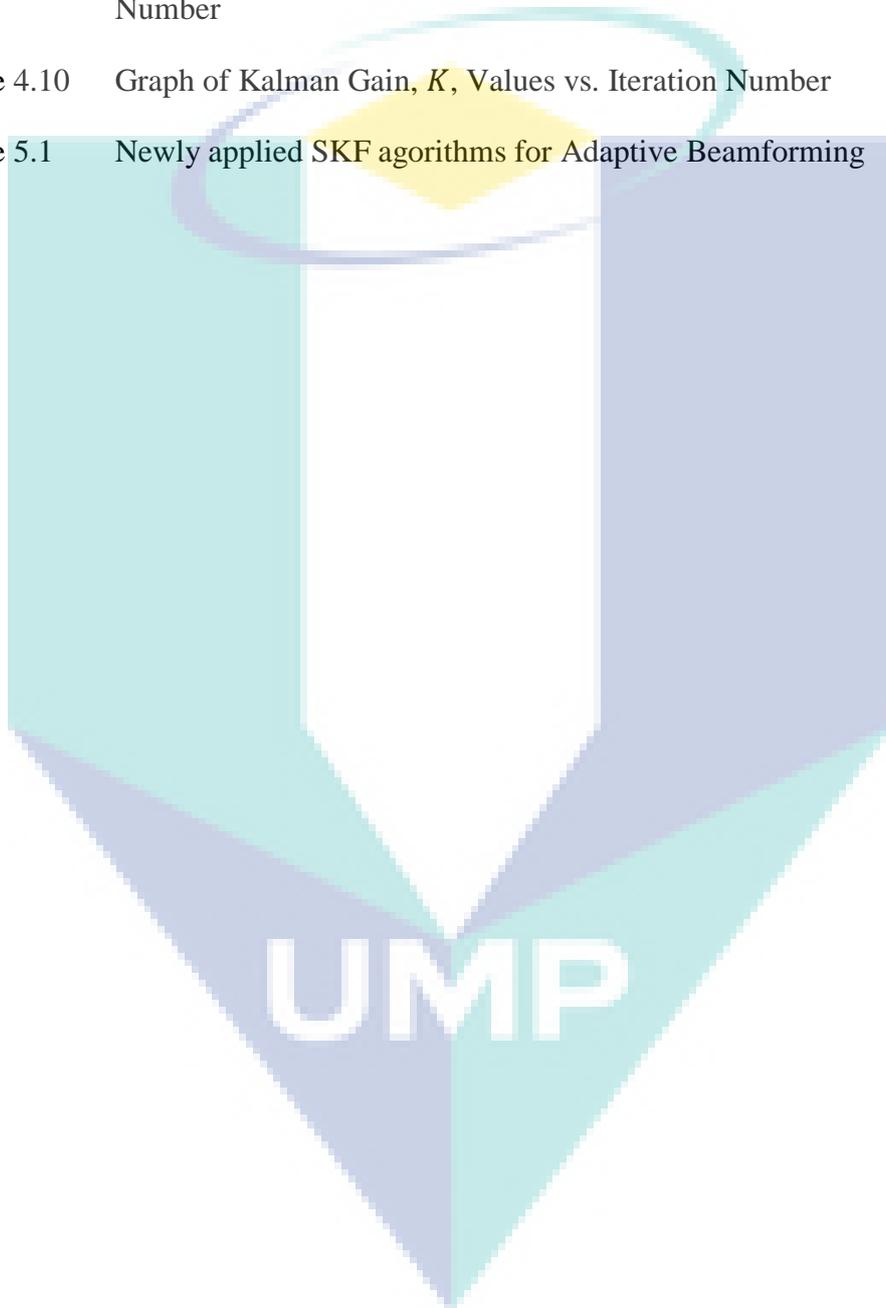
Table 4.21	Best, Worst, Mean and Standard Deviation(STD) of SINR for SKF vs SKFMM	70
Table 4.22	Sum of Ranks for SKF vs SKFMM	71
Table 4.23	Best, Worst, Mean and Standard Deviation(STD) of SINR for OBSKF vs SKFMM	72
Table 4.24	Sum of Ranks for OBSKF vs SKFMM	73
Table 4.25	Sum of Standard Deviation for AMBPSO, SKF, OBSKF and SKFMM	78
Table 4.26	Percentage of Improvement in Consistency	78



LIST OF FIGURES

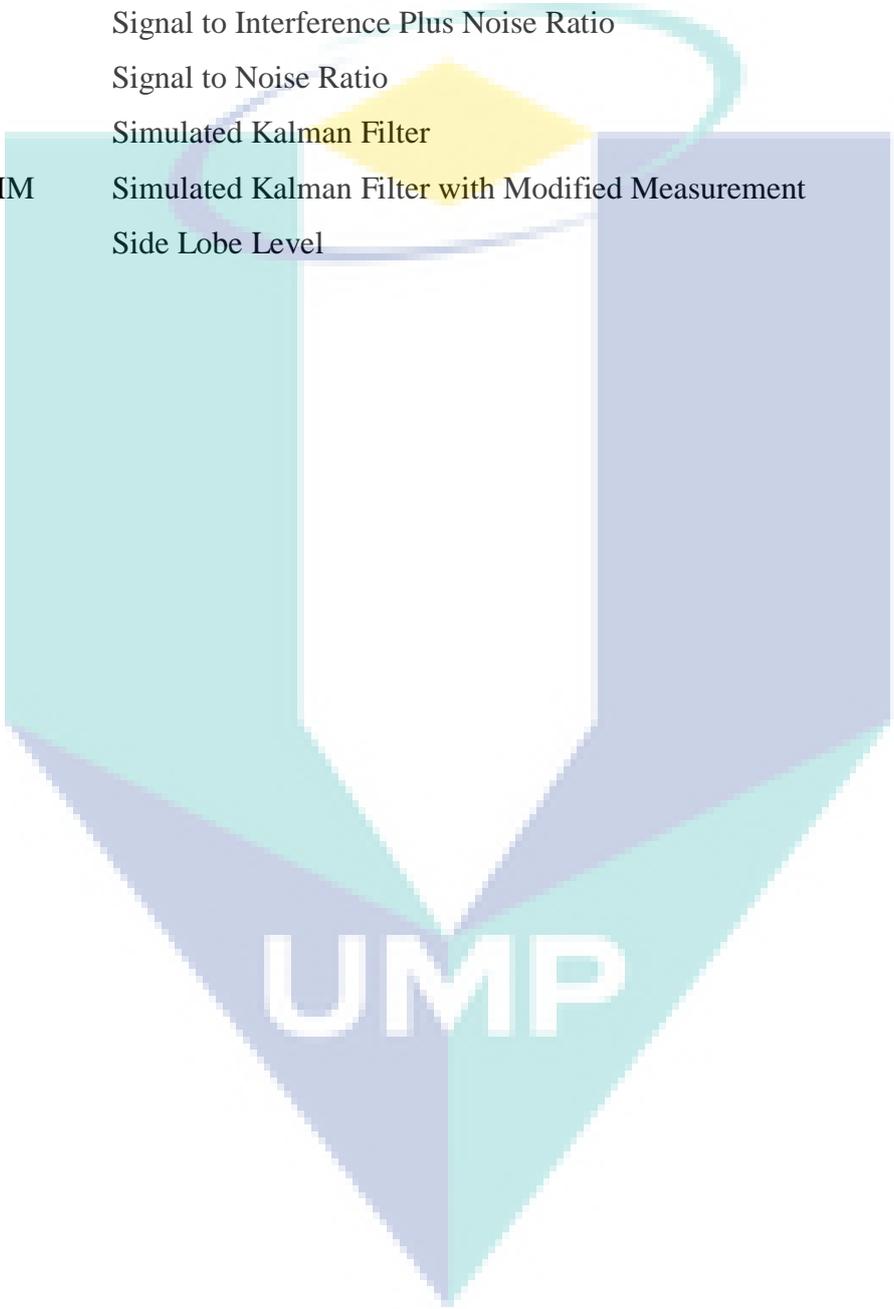
Figure 1.1	Model of an Array Antenna	2
Figure 1.2	General Idea of Beamforming	2
Figure 2.1	Classification of Beamforming Technique	7
Figure 2.2	Classification of Statistically Optimum Beamforming	8
Figure 2.3	Classification of Adaptive Algorithms for Beamforming	10
Figure 2.4	Summary of Optimization Algorithm Applied to Beamforming	12
Figure 2.5	The Principle of SKF algorithm	22
Figure 3.1	Ongoing Cycle of Time Update and Measurement Update in Kalman Filter	28
Figure 3.2	Flowchart of SKF algorithm	30
Figure 3.3	Current Point, x and Opposite Point, ox in Domain $[a, b]$	33
Figure 3.4	Flowchart of Opposition-Based SKF Algorithm	34
Figure 3.5	Flowchart of SKF with Modified Measurement (SKFMM)	37
Figure 3.6	N -Element Linear Array	38
Figure 3.7	Array Model with Arriving Signal	39
Figure 3.8	Isotropic Linear Array Model in MATLAB	42
Figure 3.9	Flowchart of SKF for Adaptive Beamforming	47
Figure 3.10	Flowchart of OBSKF for Adaptive Beamforming	49
Figure 3.11	Flowchart of SKFMM for Adaptive Beamforming	51
Figure 4.1	Radiation Pattern for $SNR = 15dB$	53
Figure 4.2	Radiation Pattern for $SNR = 30dB$	55
Figure 4.3	Radiation Pattern for $SNR = 50dB$	57
Figure 4.4	Convergence Curve for $SNR = 15dB$	59
Figure 4.5	Convergence Curve for $SNR = 30dB$	60
Figure 4.6	Convergence Curve for $SNR = 50dB$	61

Figure 4.7	Graph of Standard Deviation of AMBPSO, SKF, OBSKF and SKFMM	74
Figure 4.8	Difference Between Best and Worst SINR	74
Figure 4.9	Graph of Error Covariance Estimate, P , Values vs. Iteration Number	76
Figure 4.10	Graph of Kalman Gain, K , Values vs. Iteration Number	76
Figure 5.1	Newly applied SKF algorithms for Adaptive Beamforming	82



LIST OF ABBREVIATIONS

AMBPSO	Adaptive Mutated Boolean Particle Swarm Optimization
OBL	Opposition-Based Learning
OBSKF	Opposition Based Simulated Kalman Filter
SINR	Signal to Interference Plus Noise Ratio
SNR	Signal to Noise Ratio
SKF	Simulated Kalman Filter
SKFMM	Simulated Kalman Filter with Modified Measurement
SLL	Side Lobe Level



UMP

CHAPTER 1

INTRODUCTION

1.1 Overview

Omnidirectional antennas radiate in all direction equally, therefore, making it less directive. An antenna system with low directivity often produce lower gain and is not suitable for long distance communications. To solve this problem, a group of elements is assembled in electrical and geometrical configuration to form an array antenna. Array antenna can produce much focused and narrow radiation pattern. This means array antennas are more directive, producing higher gain in one direction and is suitable for reducing transmission power in long distance communication.

In an array antenna, each individual element has its own amplitude control and phase control. Figure 1.1 shows the model of an array antenna with amplitude control and phase control. The amplitude control and phase control of individual elements are known as weights. Another advantage of an array antenna is that it can adjust the direction of the radiation pattern by controlling the weights of individual elements in an array. The ability of an array antenna to adjust the weights makes the array antenna adaptive to the signal environment which is full of noise and interference signal. Therefore, adaptive array antenna is introduced.

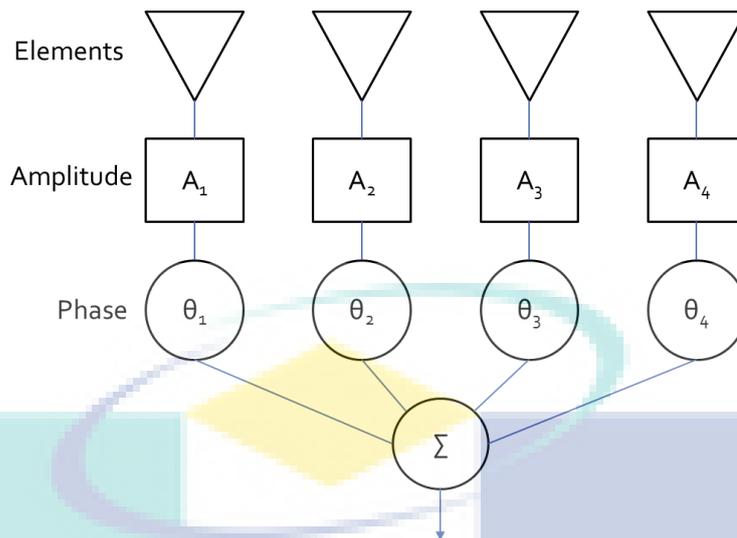


Figure 1.1 Model of an Array Antenna

An adaptive array antenna can steer the radiation pattern by directing the main beam towards the desired signal and placing null at the interference signal. Figure 1.2 shows the general idea of beamforming where the main beam point towards the desired signal and the null is placed at the interference. To change the direction of the radiation pattern, suitable weights for individual elements need to be determined.

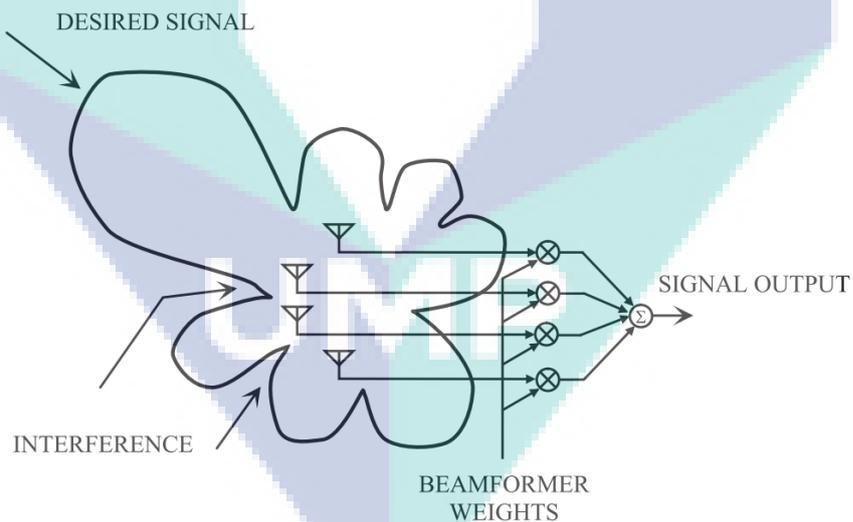


Figure 1.2 General Idea of Beamforming

Source: Constantine A. Balanis (2012)

Adaptive array antenna uses adaptive beamforming technique to find the suitable weights for individual elements. There are many methods in which beamforming can be performed. These beamforming methods are classified into three categories: statistical

optimum algorithms, adaptive algorithms and metaheuristic algorithms. These classifications will be further discussed in Chapter 2.

Over the years, many metaheuristic algorithms have been applied to adaptive beamforming application. However, a new metaheuristic algorithm named Simulated Kalman Filter (SKF), has not been applied to adaptive beamforming. The SKF algorithm, introduced by Ibrahim *et al.*, is inspired by the estimation capabilities of Kalman Filter. The SKF algorithm is a consistent algorithm and has a high convergence rate (Ibrahim et al., 2015). The SKF algorithm estimates the weights of individual elements in an array antenna to achieve maximum signal to interference plus noise ratio (SINR).

1.2 Problem Statement

The main aim for adaptive beamforming is to maximize the signal towards the desired signal and place deep null the signal at the interference direction, to achieve maximum signal to interference plus noise ratio (SINR). Metaheuristic algorithm applied to adaptive beamforming estimates the weights of individual elements to achieve maximum SINR. The problem with adaptive beamforming using metaheuristic algorithm is that sometimes the algorithm might converge prematurely, preventing from achieving maximum SINR. Furthermore, when the metaheuristic algorithm converges prematurely, the consistency of the beamformer in achieving maximum SINR is affected.

Previously published work, uses Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO), a modified version of the Particle Swarm Optimization (PSO), is used to increase the exploration of the particles in the search space (Zaharis & Yioultsis, 2011). After 100 runs, the results show that AMBPSO can produce high mean SINR values compared to existing beamforming technique and low standard deviation of SINR values for various Signal to Noise Ratio (SNR). However, the AMBPSO algorithm results shows a huge difference between the best SINR values and the worst SINR values, showing that AMBPSO is less consistent.

Simulated Kalman Filter (SKF) algorithm is a new algorithm, inspired by the estimation capabilities of Kalman Filter, is proven to be the most consistent algorithm using CEC2014 benchmark function (Ibrahim et al., 2015). However, SKF algorithm has not been applied to adaptive beamforming. With high consistency, the SKF algorithm can be suitable to improve the performance of adaptive beamforming.

The SKF algorithm is a good algorithm and there is always room to improve the algorithm's performance. One known method to improve the optimization algorithm is the Opposition-based Learning (OBL) method. The OBL method helps increase the exploration capabilities of the metaheuristic algorithm, to find better solution in the search space (Tizhoosh, 2005). The OBL method has been applied to SKF and is named as Opposition-based SKF (OBSKF) (Mohd Azmi, 2017). The OBSKF algorithm has not applied to adaptive beamforming also. Other than the OBL technique, the SKF algorithm itself can be improved for better performance when applied to adaptive beamforming.

1.3 Research Motivation

The motivation of this research is to introduce a new metaheuristic algorithm named Simulated Kalman Filter (SKF) (Ibrahim et al., 2015) to adaptive beamforming application. SKF algorithms has not been applied to adaptive beamforming. SKF algorithm is inspired by the estimation capabilities of Kalman Filter (Kalman, 1960), where the Kalman Filter can find the optimal solution regardless of the precision of the measurement. The SKF algorithm has high convergence rate (Ibrahim et al., 2015), makes it suitable optimization algorithm for adaptive beamforming, which requires high precision and consistency.

1.4 Research Objective

1. To implement Simulated Kalman Filter (SKF) and Opposition-based SKF (OBSKF) in adaptive beamforming application.
2. To improve measurement-update of the SKF algorithm for adaptive beamforming application.
3. To compare the SKF algorithms and previously published work, Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO) for adaptive beamforming applications.

1.5 Research Scope

There are many types of array geometries such as rectangular array, circular array and more. This research focuses on the use of linear array geometry.

The number of elements used in an array can vary but for this research, the number of elements is set to 10.

The spacing between elements and the operating frequency can vary but this research uses distance between elements is set 0.5λ with operating frequency at 2.4 GHz.

There are many types of elements that can be used such as dipole, patch, reflector and more. This research focuses on the isotropic element.

The simulation assumes that the direction is known, therefore, direction of arrival (DOA) algorithm will not be used. Moreover, it is also assumed that the arriving angle of both desired and interference to be static and not dynamic.

The number of desired and interference signal and the angle of arrival for each signal can vary. This research uses one desired signal fixed to 30° and 8 interference signals fixed to $-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ$.

There are many optimization algorithms such as Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Firefly Algorithm (FA) and more, but, in this research, Simulated Kalman Filter (SKF) is used for application in adaptive beamforming.

In this research, various platforms such as Computer Simulation Technology (Roslee, Subari, & Shahdan, 2011) and Antenna Magus (K.V. Rop, 2012) can be used to perform simulations. However, MATLAB is chosen as the platform for performing the simulation and testing.

1.6 Research Contributions

This research presents the first application of SKF in adaptive beamforming on antenna arrays. The SKF algorithm is proven to have better consistency (Ibrahim et al., 2015) and is suitable for application to adaptive beamforming.

The SKF algorithm, however, lacks the exploration capabilities and sometimes converging prematurely. The Opposition-based Learning (OBL) technique improves the exploration capabilities of metaheuristic algorithm, to get better solutions. The Opposition-based SKF (OBSKF), introduced by K.Z.M. Azmi, is applied to increase the

exploration SKF (Mohd Azmi, 2017). The OBSKF algorithm is suitable for application to adaptive beamforming. With increased exploration, better weights that gives maximum SINR can be determined.

In addition, SKF algorithm with modified measurement (SKFMM), is also introduced. The compromise between the exploration and exploitation of the SKF algorithm is dependent on the measurement-update. By modifying the measurement-update, the exploration of SKF algorithm can be increased, thus, improving the chance to find better weights that gives maximum SINR.

1.7 Thesis Organization

This thesis is organized into five chapters.

Chapter 2 presents the literature review on existing beamforming techniques for adaptive array antenna.

Chapter 3 discusses on the array system model and its formulation of fitness function. In this chapter, SKF, OBSKF and SKFMM application in adaptive beamforming is also discussed.

Chapter 4 shows the simulation results of SKF, OBSKF and SKFMM for adaptive beamforming application and comparison with algorithm from previously published work is presented.

Chapter 5 concludes the thesis and the future implementations for SKF, OBSKF and SKFMM for adaptive beamforming application.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter briefly summarizes the classification of beamforming techniques that are currently available.

2.2 Beamforming Techniques

Beamforming in array antenna is a process where the main beam of the radiation pattern is directed towards the desired direction and the nulls are steered towards the interference signal. This is achieved by controlling the amplitude and phase of individual elements in an array. Beamforming technique is classified into three categories; Statistically Optimum Beamforming, Adaptive Algorithms for Beamforming and Beamforming using Metaheuristic Algorithm. Figure 2.1 shows the classification of beamforming technique.

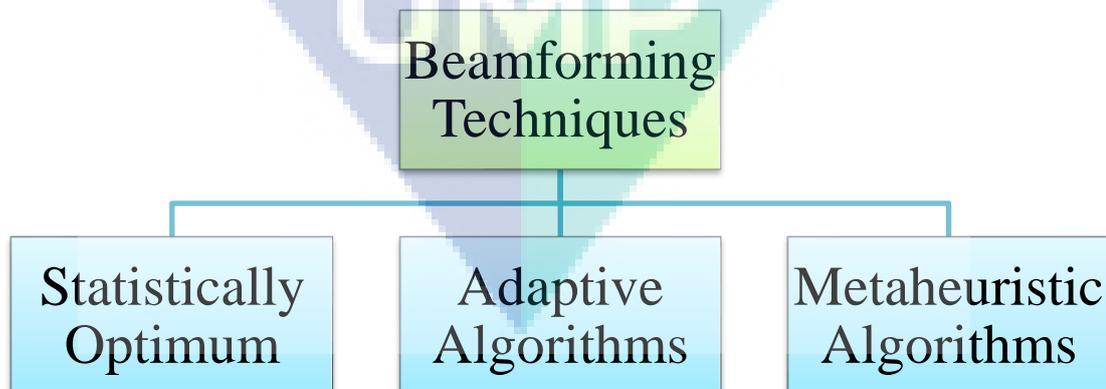


Figure 2.1 Classification of Beamforming Technique

2.3 Statistically Optimum Beamforming

Statistically optimum is where the weights of individual elements do not depend on the received array data and are chosen to present a specific response for all signal and interference scenarios. Matching randomly perturbed signal with arbitrary characteristics can be realized only in a statistical sense by using a matrix weights of input data that adapts to characteristics of received signal. Statistically optimum beamforming finds the weight vectors based on statistics of received data (Constantine A. Balanis, 2007). Some of the well-known Statistically optimum beamforming techniques are Maximize Signal to Interference Plus Noise Ratio (Maximum SINR) (Constantine A. Balanis, 2007; Frank B. Gross, 2015), Minimum Mean Square Error (MMSE) (Constantine A. Balanis, 2007; Frank B. Gross, 2015; Noordin et al., 2011), Linearly Constrained Minimum Variance (LCMV) (Constantine A. Balanis, 2007) and Minimum Variance Distortionless Response (MVDR) (Balasem, Tiong, & Koh, 2011; Frank B. Gross, 2015; Souden, Benesty, & Affes, 2010). Figure 2.2 shows the classification of statistically optimum beamforming algorithm.

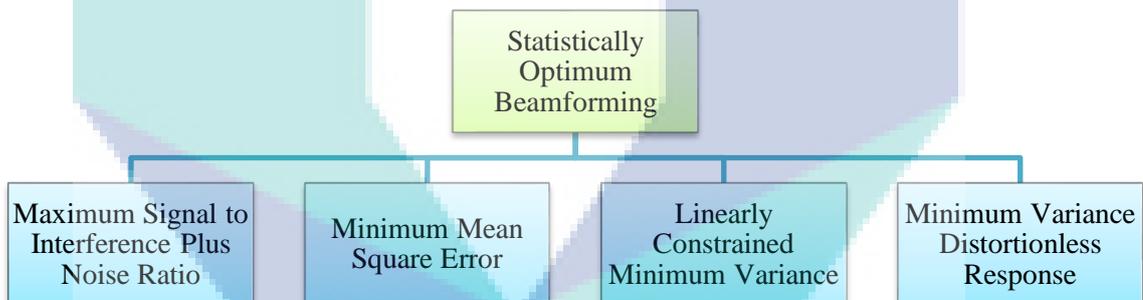


Figure 2.2 Classification of Statistically Optimum Beamforming

2.3.1 Maximum Signal to Interference Plus Noise Ratio

Maximum signal to interference plus noise ratio (maximum SINR) techniques is used to maximize the signal to interference plus noise power ratio (Constantine A. Balanis, 2007). This method has been applied various times in adaptive beamforming application (Darzi et al., 2014; Darzi, Islam, Tiong, Kibria, & Singh, 2015; Darzi, Sieh Kiong, Tariqul Islam, Rezai Soleymanpour, & Kibria, 2016; Darzi, Tiong, Islam, Ismail, & Kibria, 2015; Darzi, Tiong, Tariqul Islam, Rezai Soleymanpour, & Kibria, 2016; Doroody, Tiong, & Darzi, 2015; Zaharis & Yioultsis, 2011). However, to determine the

optimal weight vectors that gives maximum SINR requires a good estimate of the second-order statistics such as the desired signal correlation matrix and the undesired signal correlation matrix (Shiu, 1998).

2.3.2 Minimum Mean Square Error

Minimum mean square error (MMSE) is a method where the best weights of the array are determined by minimizing the mean square error between the reference signal and the output signal (Constantine A. Balanis, 2007). It is preferable that the reference signal be highly correlated with the desired signal and uncorrelated with the interference signal (Frank B. Gross, 2015). However, MMSE has a serious drawback because it relies on the reference signal. It is very difficult to produce an accurate reference signal with limited or no knowledge of the received signal (Constantine A. Balanis, 2007). Therefore, poor reference signal can degrade the performance of the array antenna.

2.3.3 Linear Constrained Minimum Variance

Linear constrained minimum variance (LCMV) constraints the beamformer response so that the desired signal passes with specified gain and phase (Constantine A. Balanis, 2007). The advantage of LCMV is that it does not require the knowledge of the desired signal correlation matrix, undesired signal correlation matrix and reference signal (Shiu, 1998). The disadvantages of this method is the computational complexity (Shiu, 1998).

2.3.4 Minimum Variance Distortionless Response

Another well-known beamforming technique is the minimum variance distortionless response (MVDR). The term distortionless undistorted received signal. The main goal of MVDR is to minimize the array output noise variance (Frank B. Gross, 2015). MVDR gives the optimum weights with minimized power of the undesired output signal while the desired output signal is maintained (Zaharis & Yioultis, 2011). However, weights obtained using MVDR is not able to produce deep nulls at the direction interference signal (Darzi, Islam, et al., 2015; Darzi, Sieh Kiong, et al., 2016; Darzi, Tiong, et al., 2015, 2016).

2.4 Adaptive Algorithms for Beamforming

Statistically optimum weight vectors can be made adaptive using the Wiener solution with the knowledge of asymptotic second-order statistics of the signal (Constantine A. Balanis, 2007). However, the knowledge of asymptotic second-order statistics of the signal and the interference plus noise was only assumed to be available. These statistics are usually not known except when assuming that the time average equals the ensemble average, the second-order statistics can be estimated from available data (Shiu, 1998). If the arriving signal changes with time, the statistics also changes with time. Therefore, it is important for the weights of individual elements in an array to be recalculated every time to find the optimum array weights. Therefore, an optimization scheme is used to continuously adapt to the changes of the arriving signal. Some of the well-known adaptive beamforming algorithms are Least Mean-Square (LMS), Recursive Least Squares (RLS) and Constant Modulus (CM) (Frank B. Gross, 2015).

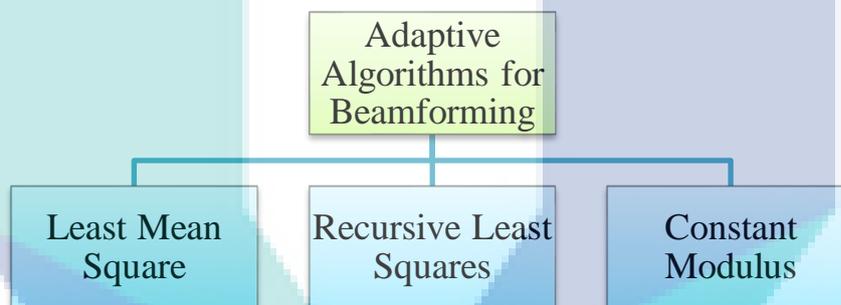


Figure 2.3 Classification of Adaptive Algorithms for Beamforming

2.4.1 Least Mean-Square

Least Mean-Square (LMS) is widely used in communication systems due to its low computational complexity and robustness (Frank B. Gross, 2015). LMS is a member of the stochastic gradient algorithms (Allen, 2005) where it iteratively minimizes the Mean Square Error (MSE). Since the error in MSE is squared, it gives a quadratic characteristic with one minimum. By using the negative steepest descent method, it updates the weight vectors with the direction of the estimated gradient which later converge to a one minimum. However, the convergence of LMS depends fully on the accuracy of the reference signal. Another drawback with LMS is the convergence characteristics where it depends on the eigen structure. Widely spread eigenvalues will

increase the convergence time (Allen, 2005; Constantine A. Balanis, 2007; Frank B. Gross, 2015; Saxena & Kothari, 2014).

2.4.2 Recursive Least Squares

Recursive Least Squares (RLS) approximates the solution directly using method of least squares to adjust the weight, without imposing additional burden of approximating an optimization procedure. In RLS, least-square method chooses the weights that minimize the fitness function which is the sum of error squares over a time window. RLS has a faster convergence speed compared to LMS. This is due to the ability of RLS algorithm to utilize information from input data and extend back to the point the algorithm begins to run. The improved performance of RLS does have its drawbacks because of increase of computational complexity (Allen, 2005; Constantine A. Balanis, 2007; Frank B. Gross, 2015; Godara, 2004; Gross, 2005).

2.4.3 Constant Modulus

Both LMS and RLS adaptive beamforming algorithm are based on the minimizing the error between the reference signal and the array output. However, reference signal needs prior knowledge of the arriving signal. When the direction of arriving signal is not known, an optimization technique needs to blindly estimate the incoming signals. Many wireless signals are phase and frequency modulated signals such as frequency modulation (FM), frequency shift keying (FSK) and phase shift keying (PSK). These signals have constant complex envelope (Constantine A. Balanis, 2007) or constant amplitude (Frank B. Gross, 2015). This constant amplitude is referred to as constant modulus (Constantine A. Balanis, 2007; Frank B. Gross, 2015). Constant Modulus (CM) works similarly to LMS algorithm using gradient based approach. However, CM algorithm does not need reference signal to work unlike LMS and RLS. This makes CM algorithm a blind adaptive beamforming algorithm (Constantine A. Balanis, 2007; Frank B. Gross, 2015; Godara, 2004). The only disadvantage of CM algorithm is that it has a slow convergence time (Frank B. Gross, 2015).

2.5 Beamforming Using Metaheuristic Algorithm

Previous sub-chapter explains the use of adaptive algorithms for beamforming. These adaptive beamforming algorithms are also known as optimization scheme used for

adaptive beamforming application (Frank B. Gross, 2015). Over the years, many metaheuristic algorithms have been applied for adaptive beamforming application. These metaheuristic algorithms stochastically estimate the weights of the elements in an array antenna. Figure 2.4 shows the summary of metaheuristic algorithms applied to adaptive beamforming application.

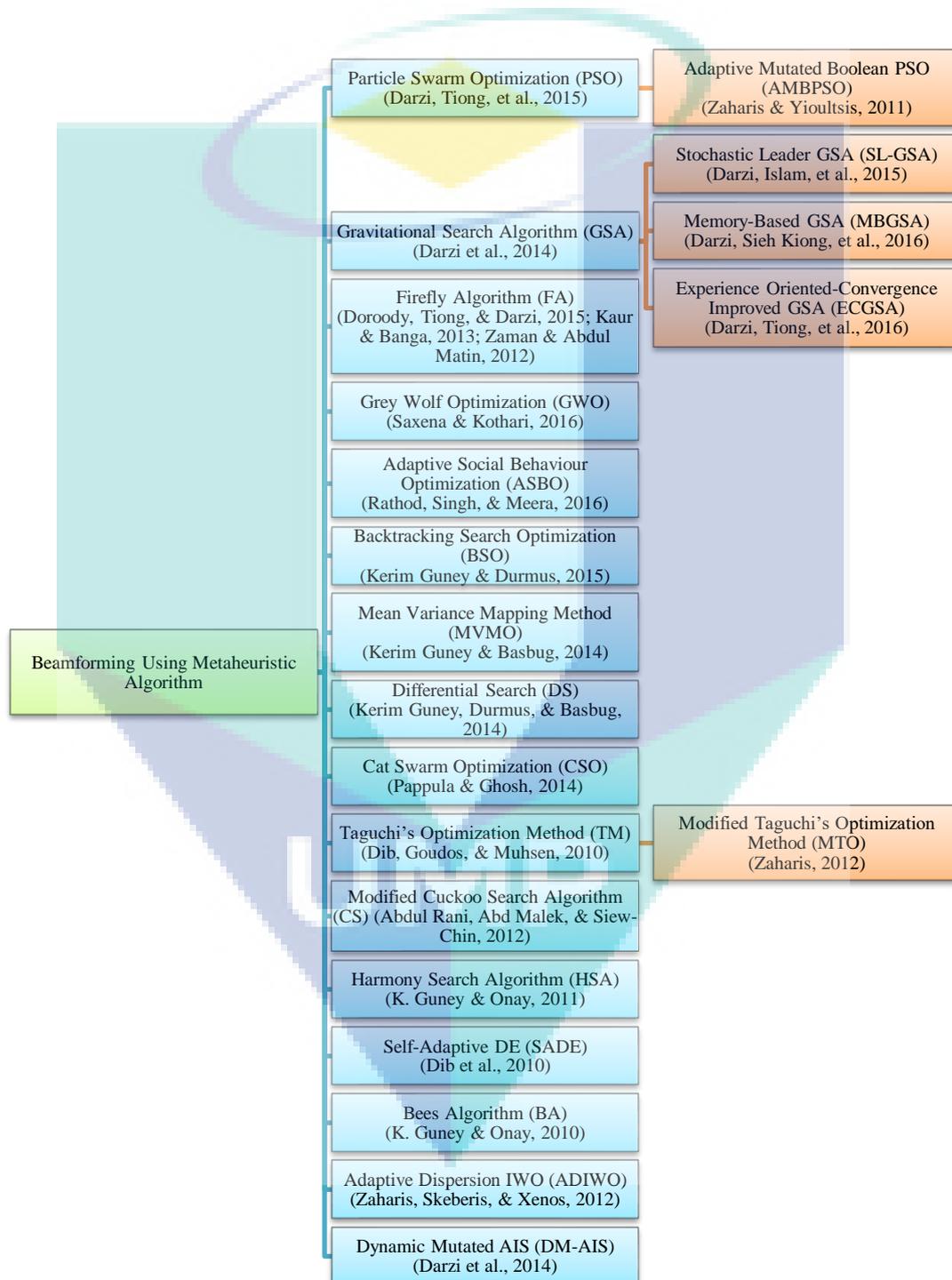


Figure 2.4 Summary of Optimization Algorithm Applied to Beamforming

2.5.1 Particle Swarm Optimization

Darzi et al. (2014) proposed the use of Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) algorithm to incorporate with Linearly Constrained Minimum Variance (LCMV) (Darzi et al., 2014). The main disadvantage of LCMV beamforming technique is low convergence rate with weak radiation pattern and low signal to interference plus noise ratio (SINR) (Darzi et al., 2014). Therefore, the main goal is to introduce PSO to optimize the weights produced by LCMV technique to get a higher SINR value. PSO is also used to optimize the weights obtained using Minimum Variance Distortionless Response (MVDR) (Darzi, Tiong, et al., 2015). Weights obtained using MVDR beamforming technique is not able to produce deep nulls for multiple interference signal scenarios. Therefore, this problem is solved as an optimization problem using PSO algorithm to maximize the SINR (Darzi, Tiong, et al., 2015). PSO is inspired by the behaviours of a flock of birds. The particles in PSO are employed as objects that evaluate through their fitness function (Darzi, Tiong, et al., 2015).

2.5.2 Gravitational Search Algorithm

One of the well-known optimization algorithm is the Gravitational Search Algorithm (GSA) which is based the Newton's law of universal gravitation (Rashedi, Nezamabadi-pour, & Saryazdi, 2009). Darzi et al. (2014) proposed the use of Gravitational Search Algorithm (GSA) to incorporate with Linearly Constrained Minimum Variance (LCMV) (Darzi et al., 2014). Due to LCMV's low convergence rate with weak radiation pattern and low signal to interference plus noise ratio (SINR), GSA is used to optimize the LCMV weights in order to get satisfactory radiation pattern and higher SINR value (Darzi et al., 2014). GSA is also used to incorporate with Minimum Variance Distortionless Response (MVDR) to improve the null depth as conventional MVDR is not able to produce satisfactorily deep nulls at the interference signal.

In GSA, the agents move towards the agent with higher mass due to its gravitational force. This makes agents in GSA highly dependent on the leading agent with maximum value. When the rest of the agents is dependent on the best agent, this cause GSA to have poor exploration of the search space might lead to stagnate at local optima. Therefore, Stochastic Leader GSA (SLGSA) is proposed as adaptive beamforming algorithm (Darzi, Islam, et al., 2015). SL-GSA is used to prevent the domination leading

agent with heavier mass at the beginning of the iteration. This is done by randomly ignoring the best agent and choosing different agent in search space as the lead agent. This method helps in increasing the exploration of search space at the beginning of the iteration. In later part of the iteration, the selection of agents other than the best agents will gradually reduce and this will lead to enhanced exploitation. The results show that SL-GSA can produce much deeper nulls and much higher SINR values compared to normal GSA.

Another variant of GSA named Experienced Oriented-Convergence Improved GSA (ECGSA) is proposed as adaptive beamforming algorithm (Darzi, Tiong, et al., 2016). ECGSA introduced two modifications to standard GSA. The first modification is that it saves the best fitness value of the agents during search process and treat these agents as best agents to apply force to other agents. This prevents the loss of discovered optimal trajectory during search process, unlike GSA, which have an unstable search trajectory. Furthermore, this helps the other agents to search in these optimal trajectories and avoid premature convergence. The second modification is it uses a special parameter known as dynamic gravitational damping coefficient, α , which controls the balance of exploration and exploitation. The α is set relatively low during the early stage of the search process which will give agents larger velocity for exploration and α is rapidly increased at the final stage for agent to converge to optimal solution. The difference between SLGSA with ECGSA is that ECGSA retains memory of results from previous iterations and SLGSA does not. This means SLGSA, like GSA, have no memory of optimal trajectory. ECGSA is less stochastic than SLGSA due to its more conventional approach by having high value of gravitational coefficient function, $G(t)$ and this makes ECGSA better than SLGSA in terms of convergence.

The next variant of GSA which is Memory Based GSA (MBGSA) (Darzi, Sieh Kiong, et al., 2016). MBGSA introduces overall best solution of population in calculation of agent positions unlike GSA, which use the best solution from previous iteration. When agents cluster at local optima, the personal best of agent values recorded from earlier explorations determines the acceleration parameter. This prevents the agents to stagnate at local optima and prevents the loss of optimal search trajectory and thus, improving the convergence. MBGSA is tested with 3 different interference signal scenarios and is compared with PSO and GSA. For each signal scenarios, the simulation is repeated 30

times. After that, Wilcoxon Rank-Sum statistical analysis is performed. The MGSA algorithm is shown to have high performance of convergence compared with GSA and PSO for adaptive beamforming application

2.5.3 Firefly Algorithm

The weights obtained using LCMV does not produce favourable radiation pattern. Therefore, Firefly Algorithm (FA) (Yang, 2010) is introduced to optimize the weights obtained using LCMV beamforming technique (Doroody et al., 2015). FA optimization method is simple and requires few parameters and can efficiently detect the global optima (Doroody et al., 2015). FA optimization method can improve the SINR value at the desired direction and is also effective in nulling multiple interference signals.

Firefly algorithm is also used to minimize the maximum side lobe level (SLL) and perform null steering by controlling phase of individual elements in an linear array antenna (Kaur & Banga, 2013). FA algorithm is inspired by the flashing behaviour of fireflies. The results show that FA outperforms Self-Adaptive Differential Evolution (SADE) and Taguchi's Optimization Method (TM) for linear array antenna in terms of efficiency and success rate. FA is found to superior in finding optimum solution for desired radiation pattern and deeper nulls.

FA is also used to determine the suitable distance between elements for a nonuniformly spaced linear array antenna with a predefined side lobe level (SLL) (Zaman & Abdul Matin, 2012). The simulation testing assumes that the amplitude excitation is constant. The performance of FA algorithm is compared to PSO in terms of best solution and the convergence rate. FA can satisfy the predefined SLL, beamwidth and null level. FA also outperforms PSO in convergence rate and the global best results.

2.5.4 Grey Wolf Optimization

The Grey Wolf Optimization (GWO) (Yang, 2010) algorithm which is inspired by the social hierarchy and hunting behaviour of grey wolves has been introduced to the electromagnetics and antenna community (Saxena & Kothari, 2016). GWO has only few parameters to tune which helps it to have faster convergence rate. The parameters in GWO can be controlled to prevent stagnation at local optimum. GWO is applied to linear array antenna in two ways which is by optimizing the position by assuming the weights

are uniform and by optimizing the amplitude of individual elements while assuming the position and phase of individual elements is constant. This paper presents the first application of GWO in pattern synthesis of linear array antenna to obtain optimal amplitude and position for individual elements. The result obtained using GWO is compared to other optimization algorithms such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Cat Swarm Optimization (CSO) and Biogeography Based Optimization (BBO). GWO outperforms the optimization techniques in pattern synthesis of linear array antenna.

2.5.5 Adaptive Social Behaviour Optimization

Adaptive Social Behaviour Optimization (ASBO) (Singh, 2012) is applied to linear array antenna to synthesize the radiation pattern by controlling the phase parameter (Rathod, Singh, & Meera, 2016). ASBO, inspired by the human social structure, is a heuristic search method that is based on the direct or indirect influences in social life of human. ASBO is tested with three set of problem characteristics with varied desired signal direction, varied number of unsymmetrical interference signal and varied number of elements. The performance of ASBO is compared with PSO. The results show that ASBO outperforms PSO, producing satisfactory radiation pattern.

2.5.6 Backtracking Search Algorithm

Backtracking Search Algorithm (BSA) is classified under evolutionary algorithms. BSA has five evolutionary steps that is initialization, selection-I, mutation, crossover and selection-II. The mutation and crossover operators in BSA produces very efficient trial populations in each generation. The generation strategy for BSA can control the amplitude of the search direction. The crossover strategy in BSA has non-linear and complex structure which ensures the creation of new trial individuals in each generation. The boundary control mechanism in BSA can achieve population diversity and ensures efficient searches, even in advanced generations (Civicioglu, 2013). BSA is proposed for linear array antenna pattern synthesis with null at the interference signal. BSA is used to control the amplitude, phase and position of individual elements in an array. BSA is compared to results obtained using seventeen different algorithms. The simulation results show that BSA can produce much lower side lobe level and much deeper nulls. (Kerim Guney & Durmus, 2015).

2.5.7 Mean Variance Mapping Optimization

Mean Variance Mapping Optimization (MVMO) (Erlich, Venayagamoorthy, & Worawat, 2010) is stochastic algorithm which is based on the strategic transformation used for mutating the offspring built on mean variance of certain dynamic population. MVMO is applied to three sets of antenna array synthesis examples (Kerim Guney & Basbug, 2014). The first example is the sidelobe level (SLL) control by using MVMO to determine the suitable distance between elements. The second example is estimate the amplitude and phase using MVMO to get a desirable radiation pattern. The third is to use MVMO to recalculate the amplitude in an event of failure of elements. MVMO is also modified to allow a good balance between exploitation and exploration, to improve the accuracy and stability of MVMO algorithm and reduce premature convergence risk. The modification done to MVMO is to increase the dimensions at the beginning of the iteration to promote good exploration and reduce the number of dimensions with the increase of iteration numbers to promote exploitation. Simulation results showed that the proposed MVMO algorithm is effective on array antenna problems and can produce a good balance between exploration and exploitation.

2.5.8 Differential Search Algorithm

Differential Search Algorithm (DS) (Civicioglu, 2012) is a stochastic search algorithm that follows the migration behaviour of organisms which use the Brownian-like random-walk movement. Population in DS is represented by artificial organisms. The artificial organisms change its position in solution space by migration movement. The population will decide whether to stay or to migrate to a more better position in the search space. The migration movement will go on iteratively until the stopping criteria is satisfied. DS is applied to three group of examples: sidelobe level (SLL) and wide nulls using amplitude control only, produce individual nulls by controlling amplitude-only, phase-only and position-only and array antenna failure correction by recalculating the amplitude. DS is very capable in solving different types of array antenna synthesis problems (Kerim Guney, Durmus, & Basbug, 2014).

2.5.9 Cat Swarm Optimization

Cat Swarm Optimization (CSO) (Chu, Tsai, & Pan, 2006) is modelled according to the features of cat's behaviour. The features are known as seeking mode and tracing

mode. Cats rest most of the time and are always alert by observing the environment. This behaviour is known as seeking mode. Tracing mode is inspired by the cat's behaviour while tracing targets and cats spend a lot of energy and move quickly while chasing a target. CSO is not difficult to implement and can easily be applied to synthesize linear array antenna by controlling the element position to suppress the sidelobe level (SLL) and produce nulls at the interference direction. The results show that CSO can find suitable position for individual element that gives low SLL and placing strong nulls at the interference directions. The CSO algorithm is compared to PSO algorithm and is found out that CSO is more computational than PSO. However, CSO can produce same computational run time as to PSO algorithm. The results also show that CSO also produce better accuracy and high convergence speed compared to PSO algorithm (Pappula & Ghosh, 2014).

2.5.10 Taguchi's Optimization Method

Taguchi's Optimization Method (TM) (Genichi Taguchi Yuin Wu, 2005) was applied to linear array antenna to minimize the sidelobe level (SLL) and place nulls at the interference signal (Dib, Goudos, & Muhsen, 2010). TM is used to control the amplitude, phase and position of individual elements in an array. TM can solve complex problems and is also easy to be implemented. The simulation results show that TM can find optimum values within 100 iterations.

Taguchi's Optimization (TO) method is an attractive choice for adaptive beamforming application due to its high convergence speed. The convergence speed of TO is controlled by a reduced rate parameter. The reduced rate parameter is the only parameter to adjust in the algorithm, thus, makes it simple and attractive for many applications. Increasing the reduced rate value will give better solution but it will also increase the convergence time. Modified TO is proposed to further decrease the computational time for adaptive beamforming application (Zaharis, 2012). The problem with TO is that it only allows positive fitness value for later logarithmic conversion to negative value in dB. Modified TO will not convert the values to dB, to allow positive and negative fitness values. Since there is no conversion to dB, there will be no average fitness value in dB and is replaced by sum of fitness. Since the conversion to dB and average fitness calculation is removed, the computational time is further reduced which makes it good for adaptive beamforming application.

2.5.11 Modified Cuckoo Search Algorithm

Modified Cuckoo Search Algorithm (MCS) is modified from the original Cuckoo Search (CS) (Genichi Taguchi Yuin Wu, 2005) algorithm. The CS is based on the brood parasitic behaviour of some cuckoo species with the Levy flight behaviour of some birds and fruit flies. The standard CS algorithm is modified to improve its performance. MCS is used to determine a suitable amplitude and phase excitation which suppress the SLL and place nulls at the intended interference direction. The results show MCS can outperform CS due to the Roulette wheel selection operator to obtain the best host nests and the dynamic inertia weight coefficient to control the global search. MCS is also compared with PSO and GA is found to be slightly better (Abdul Rani, Abd Malek, & Siew-Chin, 2012).

2.5.12 Harmony Search Algorithm

Harmony Search Algorithm (HSA) (Genichi Taguchi Yuin Wu, 2005) is stochastic algorithm that is based on the musician behaviour in the improvisation process whereby each musician tries to get the best tune in order to produce a better state of harmony. HSA is applied to linear array antenna to steer the nulls towards the interference signal direction by controlling amplitude-only, phase-only and position-only. The results show that HSA can synthesize array patterns with single, multiple and broad nulls at the direction of interference angle. HSA is also compared with 13 different optimization algorithms in literature and HSA performs better (K. Guney & Onay, 2011).

2.5.13 Self-Adaptive Differential Evolution

Self-Adaptive Differential Evolution (SADE) is modified version of the original Differential Evolution (DE) (Genichi Taguchi Yuin Wu, 2005). DE is a stochastic algorithm that consists of three operators: mutation, crossover and selection in which the population evolves in each generation. SADE add extra control parameters that self-adjust in every generation for each particle. SADE is applied linear array antenna to minimize the sidelobe level (SLL) and place nulls at the interference signal by controlling the amplitude-only, phase-only and position-only of each individual element (Dib et al., 2010).

2.5.14 Bees Algorithm

Bees Algorithm (BA) (Pham et al., 2006) is a parameter optimization algorithm inspired by the foraging behaviour of honey bees. The bee colony attempts to use the colony members in an optimal manner by recruiting more bees for visiting flowers. Food search process starts by scout bees from the colony go to look for and evaluate potential flower patches near the hive. If the scout bees more food, it will communicate back to other bees in the hive. Waggle dance is form of essential bee communication which gives information on the direction of the flower patch, distance from hive and quality of the food. This information allows the bees in the colony to accurately find the patch of flowers. After the waggle dance, the scout bee along with some other bees will fly back to the patch of flowers. The bees accompanying the scout bee determines quality of the flower patch. This process allows the bees to food fast and efficiently. BA is applied to array antenna to steer the nulls in the direction of the interference signal by controlling phase-only and controlling both amplitude and phase. The simulation results show that BA can accurately determine the element excitation to place nulls at the interference angle (K. Guney & Onay, 2010).

2.5.15 Adaptive Invasive Weed Optimization

Adaptive Dispersion Invasive Weed Optimization (ADIWO) is an improved version of Invasive Weed Optimization (IWO) (Pham et al., 2006), used for adaptive beamforming application (Zaharis, Skeberis, & Xenos, 2012). ADIWO the improved version of Invasive Weed Optimization (IWO). The exploration of IWO algorithm is based on the dispersion of seed by a weed in the search space. In IWO, the standard deviation of the seed dispersion decreases as a function of number of iteration. When iteration increases, the exploration reduces. At the end of the iteration, the exploration ends and the weed can only fine tune its position. Therefore, if optimal solution is not found, it will never be found. ADIWO prevents this by maintaining the seed dispersion. The best weed will disperse seed with lower standard deviation and worst weed will disperse seed with higher standard deviation. This adaptive seed dispersion helps in exploring the search space further and is maintained until the end of the iteration. Moreover, ADIWO converges faster than IWO because of the adaptive seed dispersion.

2.5.16 Dynamic Mutated Artificial Immune System

Dynamic Mutated Artificial Immune System (DM-AIS) is an improved variation of Artificial Immune System (AIS) (Farmer, Packard, & Perelson, 1986). AIS is based on the behaviour of human immune system reacting to foreign elements in host body. When antigens attack the body, human immune system make active antibodies. The antibodies are produced in great amounts through cloning process to fix powerfully to specific antigen. The rate of mutation of cloned antibodies is inversely proportional to the affinity of antigens. Lowest affinity antibodies have highest mutation rate. DM-AIS has a new dynamic mutation function where the population of antibodies is derived from the fitness value based on dynamic mutation rate. DM-AIS improves the convergence rate of the antibody solution. DM-AIS is used to optimize the weights obtained using LCMV beamformer to increase the SINR. The results show that DM-AIS can improve the SINR compared to the use of conventional LCMV method (Darzi et al., 2014).

2.6 Simulated Kalman Filter

Simulated Kalman Filter (SKF) (Ibrahim et al., 2015) is a new metaheuristics optimization algorithm proposed to solve unimodal optimization problems. The SKF algorithm is inspired by the estimation capabilities of Kalman Filter (Kalman, 1960), which is a well-known state estimation method. The SKF algorithm simulates the measurement process as an individual agent's update mechanism, acting as a feedback in estimating the optimum. Figure 2.5 shows the principle of the SKF algorithm. SKF solves optimization problem by finding the estimate of the optimum. The method and concept of the SKF algorithm will be further explained in sub-chapter 3.3.

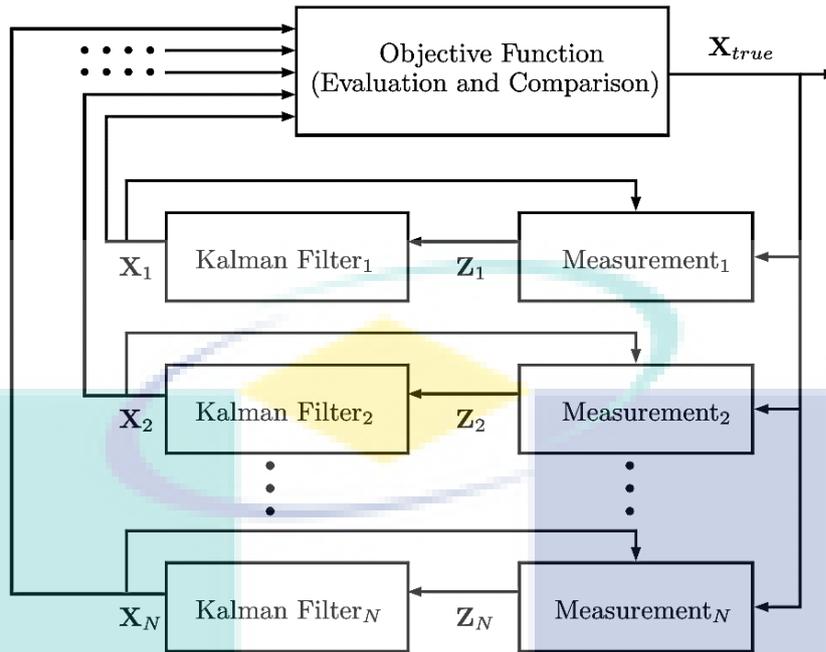


Figure 2.5 The Principle of SKF algorithm
 Source: Ibrahim et al. (2015)

In modelling the optimization problem as an estimation process of the optimum, the static model of the Discrete Kalman Filter was employed because the estimated optimum solution is not time dependent. Since the optimum solution to be estimated is not time dependent, the state vector, which holds an agent's estimated position in the search space, is reduced to scalar form. These estimated state is used in the calculation of the fitness based on the fitness function. The SKF algorithm begins with initialization of the population and the solution of the initial population are evaluated and a true value is updated. The SKF algorithm iteratively improves the estimation by using standard Kalman Filter framework which comprises of predict, measure and estimate.

The SKF algorithm was tested using CEC2014's benchmark functions (Suganthan et al., 2013). All the benchmark functions are minimization problem and rotated to the global minimum. The search space for all the benchmark functions is the same which is $[-100,100]$. The experiment was conducted using 100 agents, randomly distributed over the search space of 50 dimensions. The maximum number of iterations used for the benchmark testing was 2000 iterations. The SKF algorithm was compared with other metaheuristic algorithms such as Heuristic Kalman Filter (HKA), Gravitational Search Algorithm (GSA) and Black Hole algorithm (BH) with the same parameter setting. The experiment was run 50 times to get average performance.

The results show SKF algorithm produces the most consistent performance compared to HKA, GSA and BH algorithms for unimodal functions. The results also show that both SKF and HKA, which uses Kalman Filter approach, have high convergence rate than GSA and BH. After 50 runs, statistical analysis is performed using Friedman Rank Test. The performance of algorithm is ranked based on the mean value over the total number of runs for all 3 unimodal benchmark functions. Among the four algorithms, SKF is ranked the best for average Friedman rank. From the average Friedman rank, the Friedman statistical value was calculated and compared to the critical value according to the chi-square distribution with 3 degrees of freedom. The Friedman test shows there is no significant difference between the algorithms.

2.7 Opposition-Based Simulated Kalman Filter

The SKF algorithm has shown to produce good results in solving unimodal benchmark functions. The goal of a metaheuristic algorithm to find more accurate solution for optimization problems. Therefore, the Opposition-Based Simulated Kalman Filter (OBSKF) (Mohd Azmi, 2017) was introduced to improve the SKF algorithm using Opposition-Based Learning (OBL) technique. The OBL (Tizhoosh, 2005) technique has shown to improve many metaheuristic algorithms in literature (Xu, Wang, Wang, Hei, & Zhao, 2014). The OBL technique will be further explained in sub-chapter 3.4 and OBSKF algorithm will be further explained in sub-chapter 3.5.

2.8 Adaptive Mutated Boolean Particle Swarm Optimization

Adaptive Mutated Boolean PSO (AMBPSO) is one of the variant of PSO that is used for adaptive beamforming application (Zaharis & Yioultis, 2011). Unlike conventional PSO, AMBPSO uses Boolean form and adaptive mutated as update mechanism. This makes AMBPSO a robust optimization algorithm for adaptive beamforming. Boolean PSO is a binary version of PSO. What make AMBPSO different than Binary PSO is that the update mechanism uses Boolean update, whereas, Binary PSO which uses only real number expression. AMBPSO can control the convergence speed by adjusting the maximum allowed velocity using negative selection (NS) which is a basic mechanism of Artificial Immune System (AIS). After completion of NS, exploitation begins through adaptive mutation process with mutation probability.

Mutation probability will decrease with the number of iteration until end of optimization process.

AMBPSO is tested using 10 elements, arranged in linear geometry, with one desired signal and eight different interference signals. AMBPSO is compared to MVDR in terms of radiation pattern and statistical result. For radiation pattern comparison, four cases were studied with different distance between elements and with different Signal to Noise (SNR). For statistical result, the experiment is repeated for 100 times for various SNR inputs and the best, worst, mean and standard deviation for Signal to Interference plus Noise Ratio (SINR) is determined. The results show that AMBPSO can produce much deeper nulls and lower sidelobe level compared to MVDR for all the four cases. For the statistical results, AMBPSO produce much better mean SINR values compared to the SINR values obtained using MVDR beamforming technique regardless of the SNR input values. AMBPSO is also shown to have low standard deviation values which proves that AMBPSO is much efficient and stable algorithm for adaptive beamforming.

Since AMBPSO is a stable and efficient algorithm for adaptive beamforming, it is a suitable algorithm to be compared with the SKF algorithm for adaptive beamforming. One of the most important variable for calculating the SINR is the noise power. Some publications (Darzi et al., 2014; Darzi, Islam, et al., 2015; Darzi, Sieh Kiong, et al., 2016; Darzi, Tiong, et al., 2015, 2016; Doroody et al., 2015) never uncover or record the noise power used during experimentation which makes it difficult to implement the same methodology to other algorithms for comparison. However, the methodology introduced by Zaharis and Yioultsis for testing AMBPSO does not have any hidden variables such as the noise power, therefore, makes the methodology suitable to implement other algorithms for comparison.

2.9 Research Gap Analysis

Previous sub-chapter explains some of the beamforming techniques available and one of the technique is to use metaheuristic algorithm. Over the years, many metaheuristic algorithms have been applied to adaptive beamforming. Some of the metaheuristic algorithms have been modified from the original algorithm to improve the performance for adaptive beamforming application (Darzi, Islam, et al., 2015; Darzi, Sieh Kiong, et al., 2016; Darzi, Tiong, et al., 2016; Zaharis, 2012; Zaharis & Yioultsis, 2011). However,

a new metaheuristic algorithm named Simulated Kalman Filter (SKF) (Ibrahim et al., 2015) has not been applied to adaptive beamforming. Therefore, the purpose of this research is to introduce SKF into adaptive beamforming application and to apply improved versions of SKF algorithm for adaptive beamforming.

Out of all the algorithms reviewed, none of the algorithms showed any statistical analysis (except for MBGSA (Darzi, Sieh Kiong, et al., 2016) and AMBPSO (Zaharis & Yioultsis, 2011)), to prove the consistency of the optimization algorithm is solving adaptive array antenna problems. Comparison is made only with AMBPSO because the method introduced for testing with MBGSA has hidden variable such the noise power, which makes it impossible for the experimental condition to be replicated with the SKF algorithms.

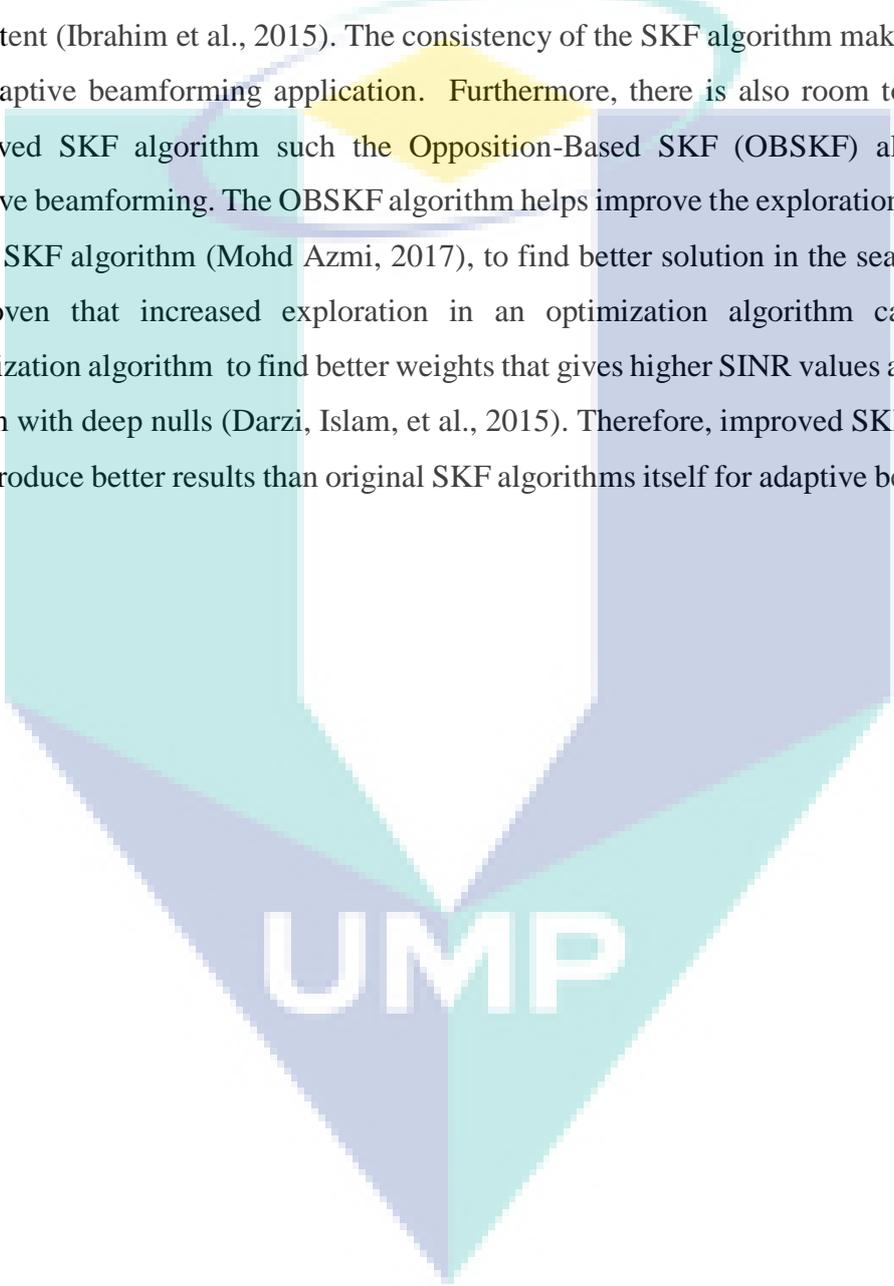
2.10 Summary

Several metaheuristic algorithms applied to adaptive beamforming has been reviewed. Each metaheuristic algorithm has different ways of solving adaptive beamforming problems. For most of the publication reviewed, the deeper nulls and lower sidelobe level were some of the problems focused where the metaheuristic algorithms are used to get deeper nulls and much lower sidelobe level (Abdul Rani et al., 2012; Dib et al., 2010; K. Guney & Onay, 2010, 2011; Kerim Guney & Basbug, 2014; Kerim Guney & Durmus, 2015; Kerim Guney et al., 2014; Kaur & Banga, 2013; Pappula & Ghosh, 2014; Rathod et al., 2016; Saxena & Kothari, 2016; Zaman & Abdul Matin, 2012). Some of the metaheuristic algorithms have been modified from the original algorithm to improve the performance for adaptive beamforming application (Darzi, Islam, et al., 2015; Darzi, Sieh Kiong, et al., 2016; Darzi, Tiong, et al., 2016; Zaharis, 2012; Zaharis & Yioultsis, 2011).

The Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO) algorithm, introduced by Z.D Zaharis and T.V. Yioultsis, is proven to be consistent for adaptive beamforming application (Zaharis & Yioultsis, 2011). AMBPSO has better consistency due to its low standard deviation and its mean value close to the best SINR values regardless of the Signal to Noise (SNR) values. Furthermore, the methodology used to test AMBPSO for adaptive beamforming is easy to implement and does not have any hidden variables such as the noise power. Comparison is made only with AMBPSO

because the method introduced for testing with MBGSA has hidden variable such the noise power, which makes it impossible for the experimental condition to be replicated with the SKF algorithm.

Simulated Kalman Filter (SKF) is a new metaheuristic algorithm and has not been applied to adaptive beamforming. The performance SKF algorithm is proven to be consistent (Ibrahim et al., 2015). The consistency of the SKF algorithm makes it suitable for adaptive beamforming application. Furthermore, there is also room to implement improved SKF algorithm such the Opposition-Based SKF (OBSKF) algorithm for adaptive beamforming. The OBSKF algorithm helps improve the exploration capabilities of the SKF algorithm (Mohd Azmi, 2017), to find better solution in the search space. It is proven that increased exploration in an optimization algorithm can help the optimization algorithm to find better weights that gives higher SINR values and radiation pattern with deep nulls (Darzi, Islam, et al., 2015). Therefore, improved SKF algorithms may produce better results than original SKF algorithms itself for adaptive beamforming.

The logo of UWP (Universiti Wawasan Putrajaya) is a large, stylized letter 'V' shape. The left side of the 'V' is light blue, the right side is light purple, and the bottom point is a darker blue. The letters 'UWP' are written in white, bold, sans-serif font across the bottom of the 'V'.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter explains the Kalman Filter, Simulated Kalman Filter (SKF) algorithm, Opposition-based SKF (OBSKF) and the SKF with Modified Measurement (SKFMM). This chapter also presents the formulation of the fitness function from the array system model for adaptive beamforming.

3.2 Kalman Filter

Kalman Filter (Kalman, 1960) is an optimal recursive data processing algorithm that processes all measurements, regardless of the precision, to estimate the current value. Kalman Filter estimates with the help of (1) measurement device dynamics and with the knowledge of system, (2) statistical description of the system noises, measurement errors and uncertainty in the dynamics models and (3) initial condition of the variable of interest if available (Maybeck, 1979).

The estimation process of Kalman Filter uses a form of feedback control where the filter estimates the process state and obtain feedback in the form of measurements. The equation in Kalman Filter is divided into two groups: the time-update equations and the measurement-update equations. Figure 3.1 illustrates the time-update and the measurement-update cycle Kalman Filter (Welch & Bishop, 2006).

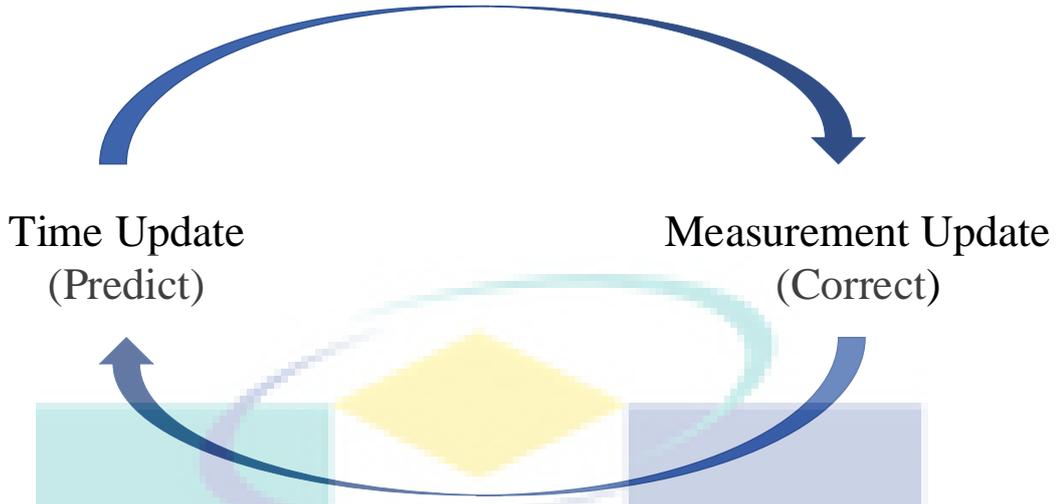


Figure 3.1 Ongoing Cycle of Time Update and Measurement Update in Kalman Filter

Source: Welch & Bishop (2006)

The time-update equations or also known as *predictor* equations use the current state and error covariance estimates to obtain *a priori* estimate for the next step. The specific equations for time-update is presented in equation 3.1 and 3.2

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad 3.1$$

$$P_k^- = AP_{k-1}A^T + Q \quad 3.2$$

where \hat{x}_k^- is the state covariance estimates and P_k^- represents the error covariance estimates forward from time step $k - 1$ to k . A is the state transition matrix which applies the effect of each system state parameter at time $k - 1$ on the system state at time t and B is the control input matrix which applies the effect of each control input parameter in the vector u_k on the state vector. Q represents the process noise covariance matrix (Faragher, 2012; Welch & Bishop, 2006).

The measurement-update equations or also known as *corrector* equations are used to incorporate new measurement into the *a priori* estimate to obtain the improved *a posteriori* estimate. In measurement-update, equations 3.3, 3.4 and 3.5 are used

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \quad 3.3$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad 3.4$$

$$P_k = (1 - K_k H)P_k^- \quad 3.5$$

where K_k is the Kalman gain, \hat{x}_k is the *a posteriori* state estimate which is computed from the sum of the *a priori* state estimate, \hat{x}_k^- , and the difference between the measurement, z_k , and measurement prediction, $H\hat{x}_k^-$. The H is the transformation matrix that maps the state vector parameters into the measurement domain (Faragher, 2012; Welch & Bishop, 2006).

3.3 Simulated Kalman Filter

Simulated Kalman Filter (SKF) is a new metaheuristic optimization technique inspired by the estimation capabilities of Kalman Filter (Ibrahim et al., 2015). Since then, SKF has been applied to solve various optimization problems (Md Yusof et al., 2016a, 2016b; Md Yusof, Ibrahim, et al., 2015; Md Yusof, Satiman, et al., 2015; Muhammad et al., 2015). SKF finds optimum solution by prediction, measurement and estimation.

Considering t as the number of iteration and N as the number of agents, the estimated solution of an optimization problem of the i^{th} agent at a time t , $X_i(t)$, is defines as:

$$X_i(t) = \{x_i^1(t), x_i^2(t), \dots, x_i^d(t), \dots, x_i^D(t)\} \quad 3.6$$

where $x_i^d(t)$ is the estimated state of the i^{th} agent in the d^{th} dimension with D representing the maximum number of dimensions. In an iteration t , several agents are involved in the calculation of fitness and an agent with the best fitness, $X_{best}(t)$, is identified. The SKF algorithm performs a simulated measurement process, which leads to a true value, X_{true} . The X_{true} is the best-so-far solution and is updated when a better solution of X_{true} is found.

The illustration SKF algorithm is shown Figure 3.2. Similar to other optimization algorithms, SKF begins with random initialization of its agents' estimated state, $X(0)$, within the search space. The initial value of error covariance estimate, $P(0)$, the process noise, Q , and the measurement noise, R , are also defined during initialization. The maximum number iteration, $tMax$, is also initialized.

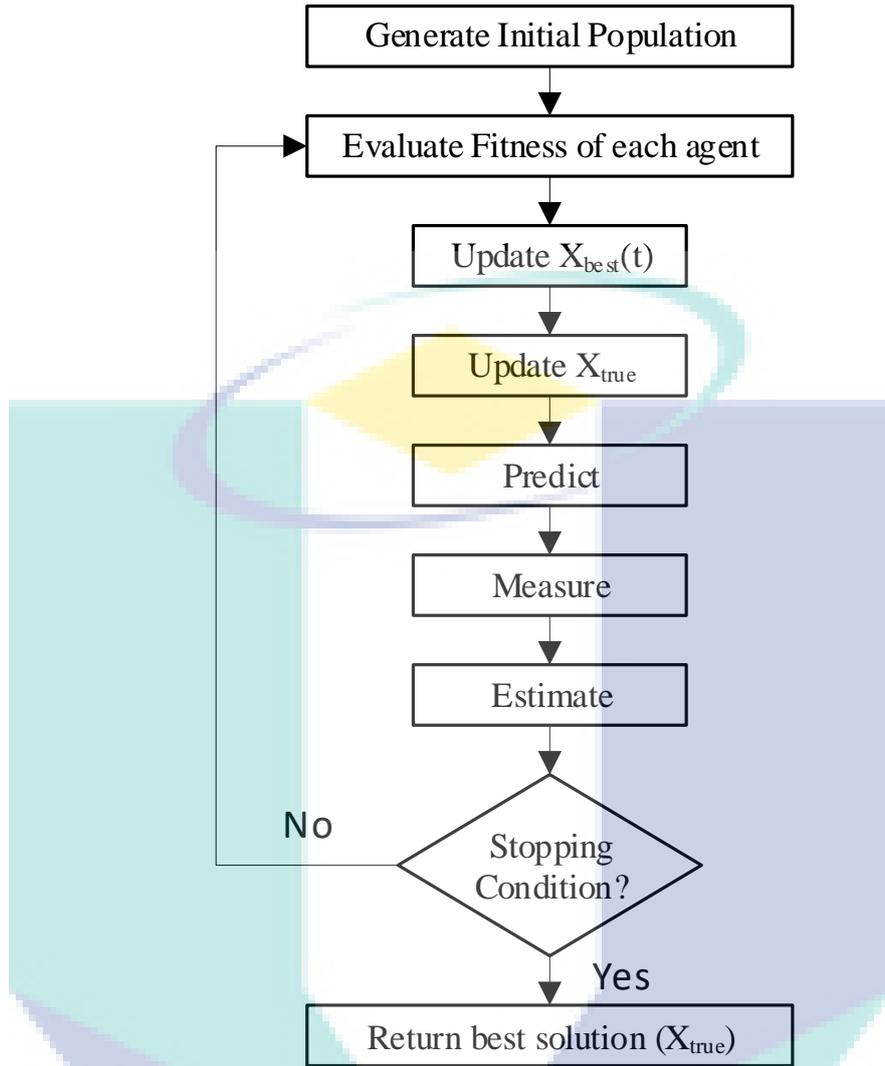


Figure 3.2 Flowchart of SKF algorithm

Source: Ibrahim et al. (2015)

The iteration starts with the fitness calculation of the i^{th} agent, $fit(X(t))$. The agent with best fitness, $X_{best}(t)$, is identified. For minimization problem,

$$X_{best}(t) = \min_{i \in \{1, 2, \dots, N\}} fit_i(X(t)) \quad 3.7$$

whereas, for maximization problem,

$$X_{best}(t) = \max_{i \in \{1, 2, \dots, N\}} fit_i(X(t)) \quad 3.8$$

SKF performs a simulated measurement process to get the true value, X_{true} . The X_{true} is the best-so-far solution that is updated iteratively when better solution than X_{true}

is found. For X_{true} to be updated, $X_{best}(t) < X_{true}$ for minimization problem, or $X_{best}(t) > X_{true}$ for maximization problem.

SKF search strategy follows three simple steps that are predict, measure, and estimate. Therefore, two sets of Kalman equation are used in SKF; the time-update equation and the measurement-update equation. In prediction, the time-update equations are used to obtain the *a priori* estimates for the next time step. After the measurement process, estimation equations are used to obtain the improved *a posteriori* estimates.

In prediction step, the time-update equations:

$$x_i(t|t+1) = x_i(t) \quad 3.9$$

$$P(t|t+1) = P(t) + Q \quad 3.10$$

are employed to make a prediction of the state and error covariance estimates given the prior estimates. These estimates are called the *a priori* estimates. $x_i(t|t+1)$ represents the predicted state, $x_i(t)$ is the previous estimated state error, $P(t|t+1)$ is the predicted error covariance due to estimation, $P(t)$ is the previous estimated error covariance, and Q is the process noise.

The next step is measurement. Measurements act as feedback to estimation process. Measurement of each individual agent is simulated based on the following equation:

$$z_i(t) = x_i(t|t+1) + \sin(rand \times 2\pi) \times |x_i(t|t+1) - x_{true}| \quad 3.11$$

where $z_i(t)$ is the value of measured position for agents, and $x_i(t|t+1)$ is the predicted state estimate. Measurement may take any random value from the predicted state, $x_i(t|t+1)$, to the true value, X_{true} . The random element, $rand$, in $\sin(rand \times 2\pi)$, is a uniformly distributed random number in the range of $[0,1]$, is important to induce stochastic behaviour in SKF.

The final step is the estimation. This step updates the error covariance estimate with the Kalman gain value. The Kalman gain, $K(t)$, is computed as follows:

$$K(t) = \frac{P(t|t+1)}{P(t|t+1) + R} \quad 3.12$$

where R represents the measurement noise. Then, the measured position with the influence of Kalman gain updates the position of each agents using equation 3.13. Equation 3.14 updates the error covariance estimate with the influence of Kalman gain.

$$x_i(t+1) = x_i(t|t+1) + K(t) \times (z_i(t) - x_i(t|t+1)) \quad 3.13$$

$$P(t+1) = (1 - K(t)) \times P(t|t+1) \quad 3.14$$

After this, the next iteration will be executed until the maximum number of iterations, $tMax$, is reached.

3.4 Opposition-Based Learning

Opposition-Based Learning (OBL) is one of the techniques used improve the optimization algorithms. The OBL technique is introduced by Tizhoosh (Tizhoosh, 2005). Many of the optimization algorithms begins with the randomization of initial starting points. If the starting point is closer to the optimal solution, the speed of convergence is faster. However, if the starting point is far away from the optimal solution, the speed of convergence will be slower and sometimes the solution might not converge to the optimal solution. The main concept of OBL technique is to further explore the search space by continuously checking the current solution with the opposite solution within the search space (Xu et al., 2014). According to a survey done by Xu *et al.*, OBL technique has been applied to improve various optimization algorithm such as Differential Evolution (DE), Particle Swarm Optimization (PSO), Reinforcement Learning (RL), Biogeography-Based Optimization (BBO), Artificial Neural Network (ANN), Harmony Search (HS), Ant Colony System (ACS) and Artificial Bee Colony (ABC) (Xu et al., 2014). Figure 3.3 shows the illustration of the opposition point, ox , with current point, x in domain $[a, b]$.

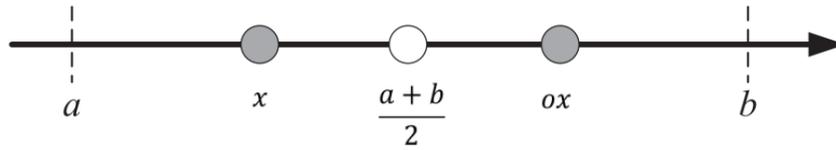


Figure 3.3 Current Point, x and Opposite Point, ox in Domain $[a, b]$

Source: Xu et al. (2014)

where equation 3.15 is used to determine the opposite position, ox .

$$ox = a + b - x \quad 3.15$$

The opposite point in D -dimensional space can be defined by assuming $P = (x_1, x_2, \dots, x_D)$ is point in D -dimensional space where $x_i \in [a_i, b_i]$, the opposite point, $OP = (ox_1, ox_2, \dots, ox_D)$ is completely determined by its coordinates:

$$ox_i = a_i + b_i - x_i \quad 3.16$$

By using the definition of the opposite point, the opposition-based optimization can be defined by assuming $P = (x_1, x_2, \dots, x_D)$ is point in D -dimensional space (i.e., candidate solution), $f(\cdot)$ is a fitness function to measure the candidate's fitness. Based on the opposite point definition earlier, $OP = (ox_1, ox_2, \dots, ox_D)$ is the opposite for $P = (x_1, x_2, \dots, x_D)$. If the OP has better fitness than P , $f(OP) \geq f(P)$, then point P can be replaced with OP ; otherwise the point P remains. The current point, P , and the opposite point, OP , are evaluated simultaneously to find the best point that gives the best fitness value.

3.5 Opposition Based Simulated Kalman Filter

Like any other optimization algorithm, SKF also begins by randomly initializing the agents in the search space. To increase the exploration of SKF, Opposition-Based Simulated Kalman Filter (OBSKF) (Mohd Azmi, 2017) is introduced. Figure 3.4 shows the flowchart of OBSKF.

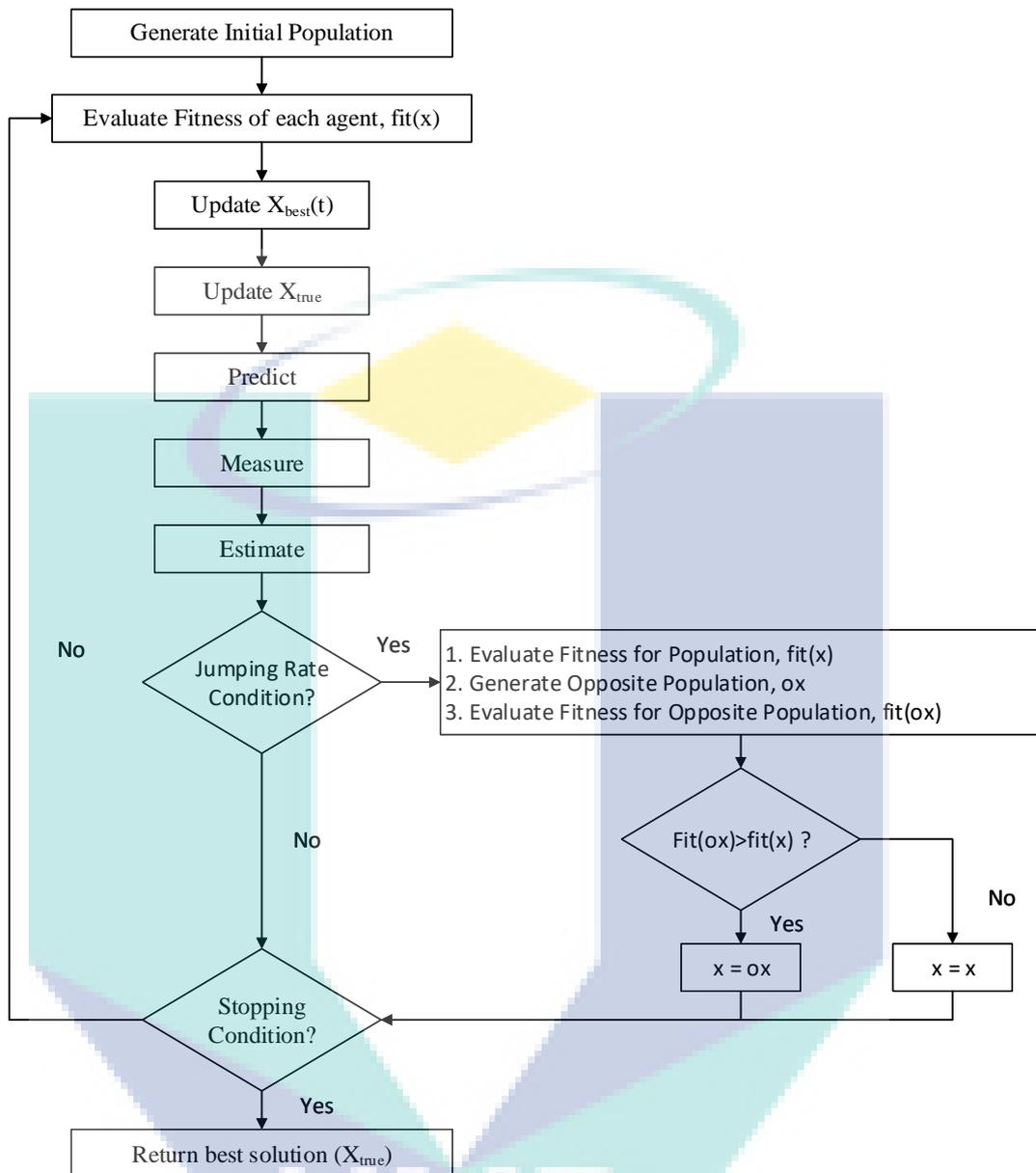


Figure 3.4 Flowchart of Opposition-Based SKF Algorithm

Source: Mohd Azmi (2017)

Similar to SKF, OBSKF begins by randomly initializing the population in the search space. Then, the calculation of fitness is performed and the best agent, $X_{best}(t)$, is identified. After that, the best-so-far solution, X_{true} , is determined and the three-simple process of SKF, predict-measure-estimate is performed. Opposition-based learning is performed after these steps.

OBL is applied to generate the opposite solution, ox , and perform fitness evaluation on opposite solution, $fit(ox)$. The fitness obtained from opposite solution, $fit(ox)$, will be compared with fitness of current solution, $fit(x)$. If the opposite fitness,

$fit(ox)$, is found to be better than the current fitness, $fit(x)$, then the current solution, x , will be replaced with the opposite solution, ox . If current fitness, $fit(x)$, is better than the opposite fitness, $fit(ox)$, the current solution remains unchanged.

The execution of OBL is controlled by a parameter known as jumping rate, $Jr \in [0,1]$. The jumping rate, Jr , is a control parameter assigned by the user. The OBL process will only begin when the random values in the range of $[0,1]$ are less than the value set for the jumping rate, Jr . The random values are obtained from the MATLAB's *rand* function.

The opposite population is calculated using dynamically updated interval boundaries $[a_j(t), b_j(t)]$ as follows (Wang, Wu, Rahnamayan, Liu, & Ventresca, 2011).

$$OX_{i,j} = [a_j(t) + b_j(t)] - X_{i,j} , \quad 3.17$$

$$a_j(t) = \min(X_{i,j}(t)), \quad b_j = \max(X_{i,j}(t))$$

$$i = 1, 2, \dots, N; \quad j = 1, 2, \dots, D$$

where $X_{i,j}$ is the current population, and $OX_{i,j}$ is the opposite population. The $a_j(t)$ and $b_j(t)$ are the lowest and the highest values of the j th dimension in current population, respectively. The N represents the maximum number of agents and D represents the maximum number of dimensions. The t represents the iteration number.

Finally, by simultaneously checking the fitness of current population and the opposite population, OBSKF can explore the search space thoroughly. With better exploration of search space, stagnation at local optimum can be avoided.

3.6 Simulated Kalman Filter with Modified Measurement

The SKF algorithm has a limitation where it is converging prematurely at local optimum. The convergence of SKF algorithm is dependent on the measurement update of each agent, $Z_i(t)$. At the beginning of the iterative process in SKF algorithm, during exploration, the difference between the best-so-far solution, X_{true} , and the predicted state estimate, $x_i(t|t + 1)$ is larger, thus, increasing the difference between the measured value

for each agent, $Z_i(t)$, and the predicted state estimate, $x_i(t|t+1)$. As the iteration progresses, the difference between the best-so-far value, X_{true} , and the predicted state estimate, $x_i(t|t+1)$ becomes smaller, thus, decreasing the difference between measured value for each agent, $Z_i(t)$, and the predicted state estimate, $x_i(t|t+1)$, promoting exploitation. However, the reducing difference between measured value for each agent, $Z_i(t)$, and the predicted state estimate, $x_i(t|t+1)$, causing SKF to converge prematurely, very early during the iterative process.

To prevent SKF from converging prematurely, a modification to the measurement update equation, $Z_i(t)$, is proposed. The modification in the measurement update is prevent the SKF algorithm to converge at local optimum. The modified measurement update, $Z_{mm_i}(t)$, is as shown below

$$Z_{mm_i}(t) = x_i(t|t+1) + \sin(rand \times 2\pi) \times |x_i(t|t+1) - X_{true}| + \sin(rand \times 2\pi) \times |X_{pbest_i} - x_i(t|t+1)| \quad 3.18$$

where X_{pbest_i} represents the best-so-far value for every agent, unlike $X_{best}(t)$, which is a single best value for every agent.

In SKF algorithm, the $X_{best}(t)$ retains only a single best value of an agent the gives the best fitness and the $X_{best}(t)$ changes with every iteration when agent with better fitness is determined. The X_{true} , on the other hand, is used to retain the value of agents with the best fitness for every iteration where X_{true} is only updated when better $X_{best}(t)$ is found. Unlike the $X_{best}(t)$ and X_{true} in the SKF algorithm, the X_{pbest_i} in SKFMM retains the best value for every agent, similar to the best position, P_{best} in Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995). The X_{pbest_i} in SKFMM helps provides better diversity because of its best-so-far values for every agent compared to single best agent for every iteration, $X_{best}(t)$ and the single best agent overall during the iterative process, X_{true} . As iteration progresses, the values of X_{pbest_i} will converge towards the best-so-far value, X_{true} .

The new estimation equation or also known as the *a posteriori* estimates is represented as

$$x_i(t + 1) = x_i(t|t + 1) + K(t) \times (Z_{mm_i}(t) - x_i(t|t + 1)) \quad 3.19$$

where the new element, $\sin(rand \times 2\pi) \times |X_{pbest_i} - x_i(t|t + 1)|$, helps to influence the differences between the modified measurement, $Z_{mm_i}(t)$, and the predicted state estimates or the *a priori* estimates, $x_i(t|t + 1)$ in the estimation process. In an iteration, the difference of values between the modified measurement and the predicted state estimates for each agent can either be positively large or positively small, or negatively large or negatively small, stochastically dependent on both the $\sin(rand \times 2\pi)$ in equation 3.18. The flowchart of SKFMM is shown in Figure 3.5.

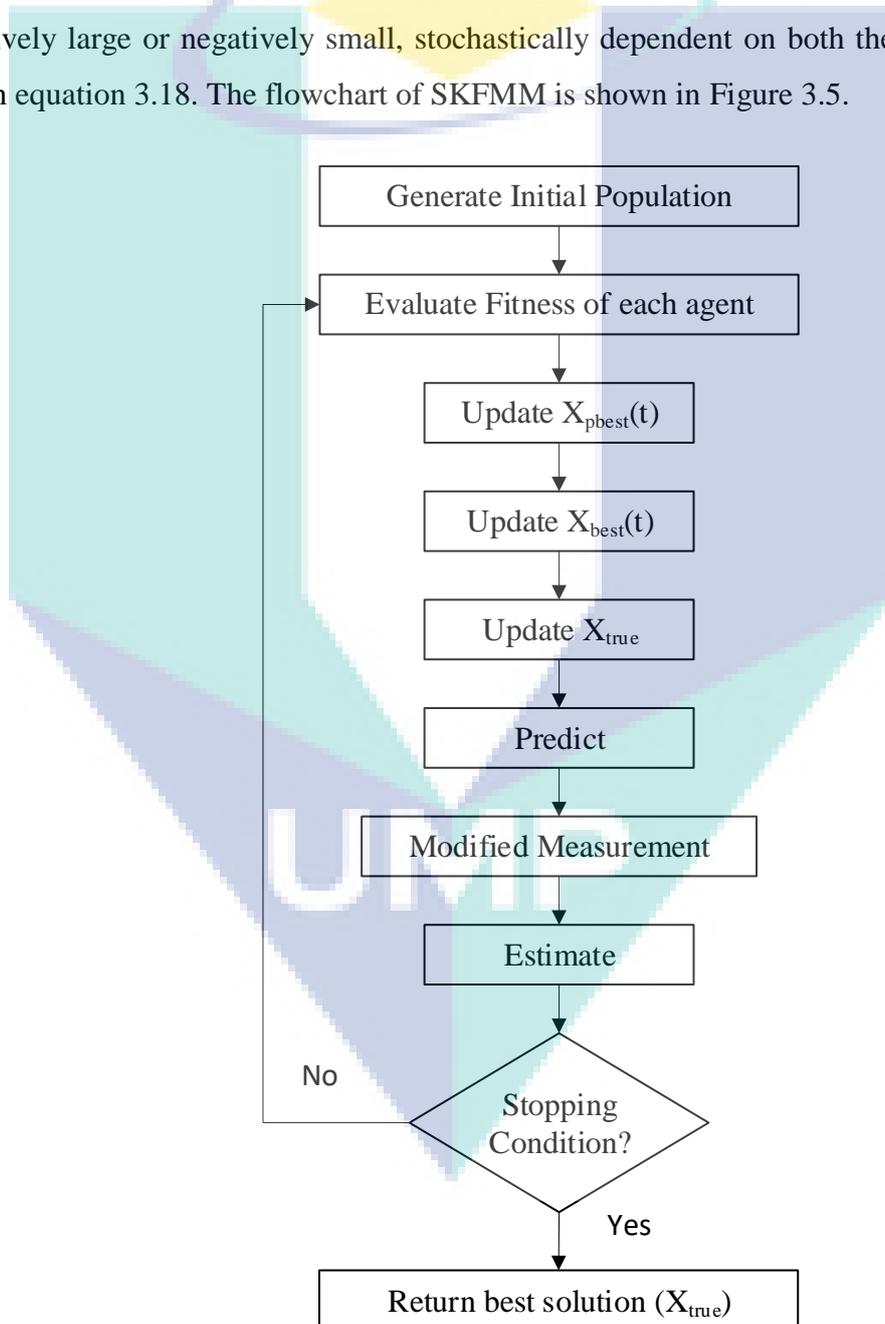


Figure 3.5 Flowchart of SKF with Modified Measurement (SKFMM)

3.7 Array System Model and Fitness Function Formulation

For an antenna system to be adaptive, it must have more than one elements. A system with more than one elements are known as array antenna. Assuming an isotropic array antenna with N number of elements with d distance between each element as shown in Figure 3.6.

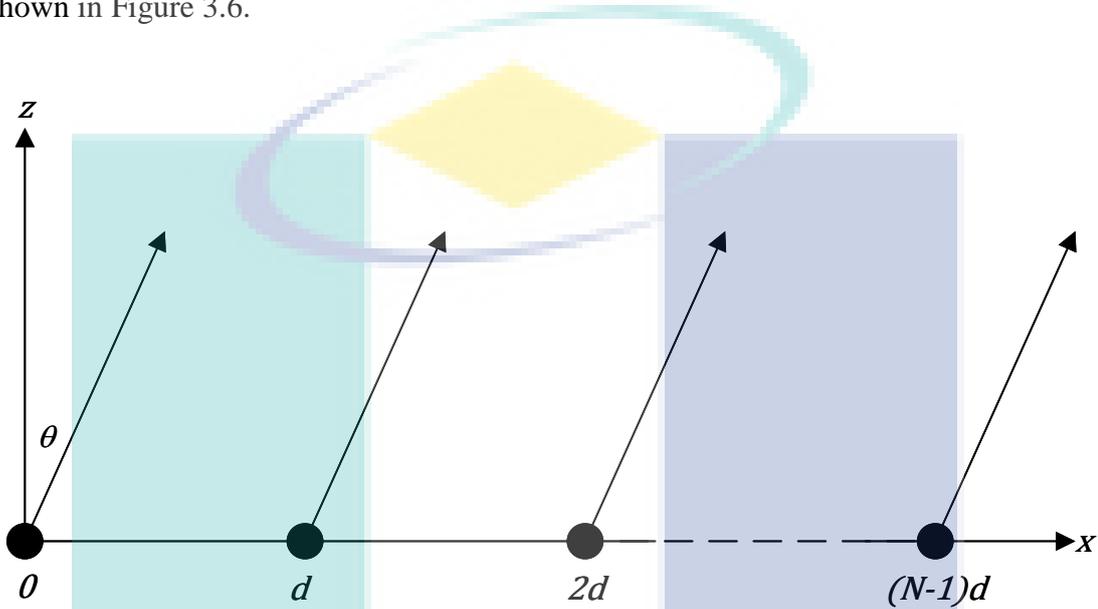


Figure 3.6 N -Element Linear Array

The following array factor, AF is derived

$$AF = A_n(1 + e^{j(kd \sin\theta + \delta)} + e^{j2(kd \sin\theta + \delta)} + \dots + e^{j(N-1)(kd \sin\theta + \delta)}) \quad 3.20$$

where $k = \frac{2\pi}{\lambda}$, denoted the wavenumber, A_n is the signal amplitudes, δ denotes the phase delay and θ denoted the theta angle measured from z axis. By assuming the system is lossless, the amplitude can be assumed as follows:

$$A_n = 1; \quad n = 1, 2, \dots, N \quad 3.21$$

Therefore, the array factor can be represented as

$$AF = \sum_{n=1}^N e^{j(n-1)\psi} \quad 3.22$$

where $\psi = kd \sin\theta + \delta$.

Assuming there are M number of interfering signal with signal of interest (SOI) of k th time sample, $s(k)$, arriving at angle θ_0 and signal not of interest (SNOI), $i_1(k)$, $i_2(k)$, $i_3(k)$, ..., $i_{M-1}(k)$, $i_M(k)$, arriving at angle θ_1 , θ_2 , θ_3 , ..., θ_{M-1} , θ_M , as shown in Figure 3.7.

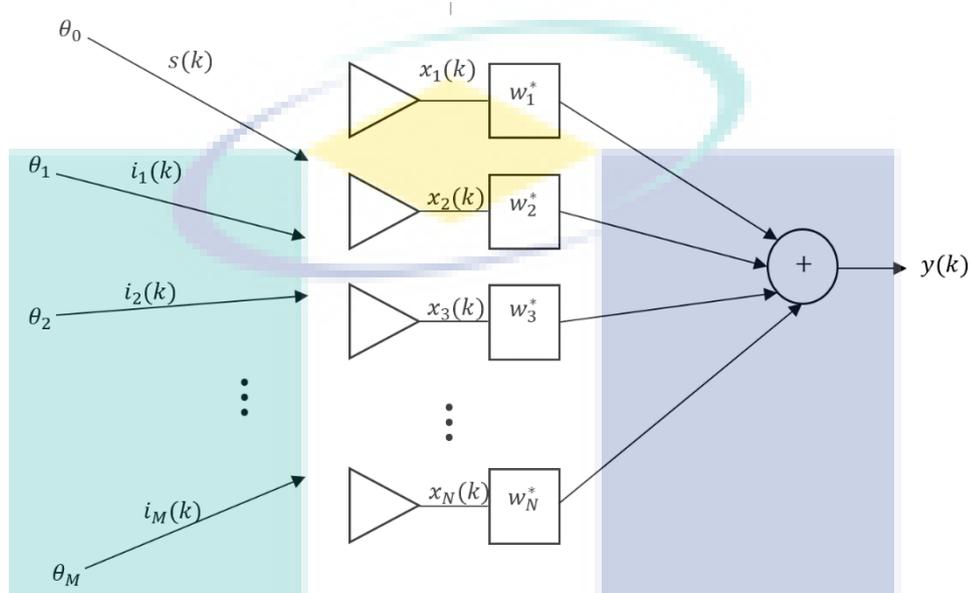


Figure 3.7 Array Model with Arriving Signal

The output of the array model, $y(k)$ can be represented by

$$y(k) = \bar{w}^H \times \bar{x}(k) \quad 3.23$$

where \bar{w} stands for the weights of individual elements in an array, Hermitian transpose is represented by H and the signal vector is represented by $\bar{x}(k)$. The signal vector, $\bar{x}(k)$ is further expanded as shown in equation 3.24

$$\begin{aligned} \bar{x}(k) &= \bar{a}_0 s(k) + [\bar{a}_1 \ \bar{a}_2 \ \dots \ \bar{a}_M] \times \begin{bmatrix} i_1(k) \\ i_2(k) \\ \vdots \\ i_M(k) \end{bmatrix} + n(k) \\ &= \bar{x}_s(k) + \bar{x}_i(k) + \bar{n}(k) \end{aligned} \quad 3.24$$

where $\bar{a}_m = [1 \ e^{j(kd \sin\theta+\delta)} \ e^{j2(kd \sin\theta+\delta)} \ \dots \ e^{j(N-1)(kd \sin\theta+\delta)}]^T$ is the array steering vector for θ_m direction of arrival. $\bar{x}_s(k)$ is the desired signal vector, $\bar{x}_i(k)$ is the interference signal vector and $\bar{n}(k)$ is the noise signal. The array factor, AF in equation 3.20 can be expressed as the sum of the elements of the array vector \bar{a}_m .

$$AF = \text{sum}(\bar{a}_m) \quad 3.25$$

The total array output, $y(k)$ is expanded as

$$y(k) = \bar{w}^H \cdot [\bar{x}_s(k) + \bar{x}_i(k) + \bar{n}(k)] = \bar{w}^H \cdot [\bar{x}_s + \bar{u}(k)] \quad 3.26$$

where the undesired signal, $\bar{u}(k)$ is formulated as

$$\bar{u}(k) = \bar{x}_i(k) + \bar{n}(k) \quad 3.27$$

After that, the array correlation matrices for the desired signal, \bar{R}_{ss} and the array correlation matrices for the undesired signal, \bar{R}_{uu} is calculated. The weighted array output power for desired signal, σ_s^2 is as follows

$$\sigma_s^2 = E[|\bar{w}^H \bar{x}_s|^2] = E[|\bar{w}^H \bar{a}_0 s(k)|^2] = \bar{w}^H \cdot \bar{R}_{ss} \cdot \bar{w} \quad 3.28$$

where \bar{R}_{ss} is the desired signal correlation matrix and E represents the expected value. The \bar{R}_{ss} does not consider the desired signal, $s(k)$, and is formulated as shown in equation 3.29

$$\bar{R}_{ss} = \bar{a}_0 \times \bar{a}_0^H \quad 3.29$$

where \bar{a}_0 is the array vector for desired signal and H represents the Hermitian transpose.

The weighted array output for undesired signal, σ_u^2 , formulated as follows

$$\sigma_u^2 = E[|\bar{w}^H \cdot \bar{u}|^2] = E[|\bar{w}^H [\bar{A} i(k) + \bar{n}(k)]|^2] = \bar{w}^H \cdot \bar{R}_{uu} \cdot \bar{w} \quad 3.30$$

where the undesired correlation matrix, \bar{R}_{uu} is formulated as

$$\bar{R}_{uu} = \bar{R}_{ii} + \bar{R}_{nn} \quad 3.31$$

where \bar{R}_{ii} denotes the interference correlation matrix. The \bar{R}_{ii} does not consider the interference signal, $i(k)$, and is formulated as shown in equation 3.32

$$\bar{R}_{ii} = \bar{A} \times \bar{A}^H \quad 3.32$$

where $\bar{A} = [\bar{a}_m]; m = 1, 2, \dots, M; m \neq 0$, represents the array vector for interference signal and H represents the Hermitian transpose. \bar{R}_{nn} is the noise correlation matrix is formulated as

$$\bar{R}_{nn} = \sigma_n^2 = 10^{-\frac{SNR}{10}} \quad 3.33$$

where SNR represents the signal to noise ratio in dB.

Then, the signal to interference plus noise ratio is formulated as in equation 3.34 (Frank B. Gross, 2015). SINR is the fitness function for the adaptive beamforming algorithm.

$$SINR = \frac{\sigma_s^2}{\sigma_u^2} = \frac{\bar{w}^H \bar{R}_{ss} \bar{w}}{\bar{w}^H \bar{R}_{uu} \bar{w}} \quad 3.34$$

The SINR formulation (Zaharis & Yioultis, 2011) can further be expanded to

$$SINR = \frac{\sigma_s^2}{\sigma_u^2} = \frac{\bar{w}^H \bar{a}_0 \bar{a}_0^H \bar{w}}{\bar{w}^H \bar{A} \bar{A}^H \bar{w} + \bar{w}^H \sigma_n^2 \bar{w}} \quad 3.35$$

3.8 Experimental Setup

This experiment follows the design in previously published work (Zaharis & Yioultis, 2011). Linear array with 10 elements was used in the work. The type of elements used are isotropic with 0.5λ distance between elements. The operating frequency is set to 2.4 GHz. Figure 3.8 shows the linear array design used in the experiment. Since this experiment assumes that the arriving angle is already known and therefore, no direction of arrival (DOA) algorithms will be used. The arriving angle for desired signal, $\theta_0 = 30^\circ$ and the interference signal, $\theta_m \in \{-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ\}$.

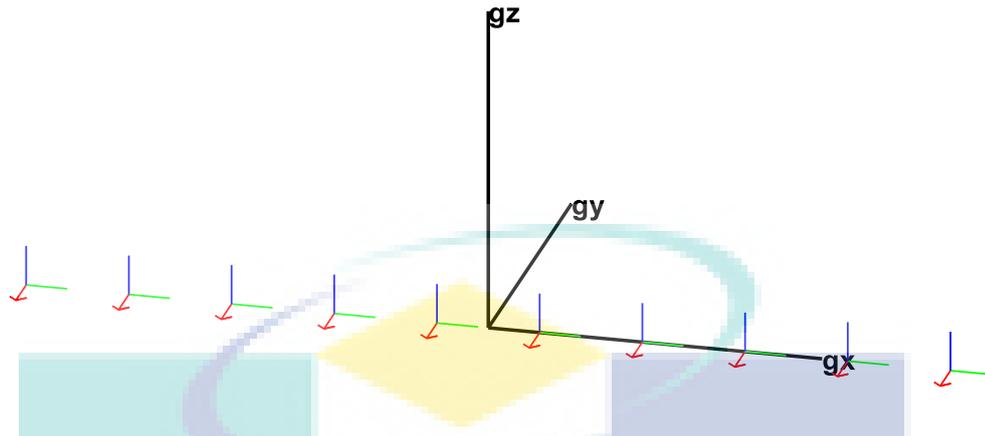


Figure 3.8 Isotropic Linear Array Model in MATLAB

Based on Figure 3.8, the array factor, AF is formulated as

$$AF = 1 + e^{j\psi} + e^{j2\psi} + e^{j3\psi} + e^{j4\psi} + e^{j5\psi} + e^{j6\psi} + e^{j7\psi} + e^{j8\psi} + e^{j9\psi} \quad 3.36$$

where $\psi = kd \sin \theta_m + \delta$, the $k = \frac{2\pi}{\lambda}$ is the wavenumber, $d = 0.5\lambda$, is the distance between elements, δ is the phase delay and θ_m is the theta angle between the z-axis and the interference angle where $m = 1, 2, \dots, M$ and M is the maximum number of interference signal. The wavelength, $\lambda = \frac{c}{f}$, where the speed of light, $c = 3 \times 10^8 \text{ m/s}$ and the antenna operating frequency, $f = 2.4\text{GHz}$. If there is no phase difference, δ between elements, the array vector, \bar{a}_m is represented by

$$\bar{a}_m = \begin{bmatrix} 1 \\ e^{jkd \sin \theta_m} \\ e^{j2kd \sin \theta_m} \\ e^{j3kd \sin \theta_m} \\ e^{j4kd \sin \theta_m} \\ e^{j5kd \sin \theta_m} \\ e^{j6kd \sin \theta_m} \\ e^{j7kd \sin \theta_m} \\ e^{j8kd \sin \theta_m} \\ e^{j9kd \sin \theta_m} \end{bmatrix} \quad 3.37$$

and the interference angle, $\theta_m \in \{-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ\}$.

The array vector, \bar{a}_m can also be represented as

$$\bar{a}_m = \begin{bmatrix} \cos 0 + j \sin 0 \\ \cos jkd \sin \theta_m + j \sin jkd \sin \theta_m \\ \cos j2kd \sin \theta_m + j \sin j2kd \sin \theta_m \\ \cos j3kd \sin \theta_m + j \sin j3kd \sin \theta_m \\ \cos j4kd \sin \theta_m + j \sin j4kd \sin \theta_m \\ \cos j5kd \sin \theta_m + j \sin j5kd \sin \theta_m \\ \cos j6kd \sin \theta_m + j \sin j6kd \sin \theta_m \\ \cos j7kd \sin \theta_m + j \sin j7kd \sin \theta_m \\ \cos j8kd \sin \theta_m + j \sin j8kd \sin \theta_m \\ \cos j9kd \sin \theta_m + j \sin j9kd \sin \theta_m \end{bmatrix} \quad 3.38$$

The array vector for the desired signal, a_0 for desired angle, θ_0 is represented as

$$\bar{a}_0 = \begin{bmatrix} \cos 0 + j \sin 0 \\ \cos jkd \sin \theta_0 + j \sin jkd \sin \theta_0 \\ \cos j2kd \sin \theta_0 + j \sin j2kd \sin \theta_0 \\ \cos j3kd \sin \theta_0 + j \sin j3kd \sin \theta_0 \\ \cos j4kd \sin \theta_0 + j \sin j4kd \sin \theta_0 \\ \cos j5kd \sin \theta_0 + j \sin j5kd \sin \theta_0 \\ \cos j6kd \sin \theta_0 + j \sin j6kd \sin \theta_0 \\ \cos j7kd \sin \theta_0 + j \sin j7kd \sin \theta_0 \\ \cos j8kd \sin \theta_0 + j \sin j8kd \sin \theta_0 \\ \cos j9kd \sin \theta_0 + j \sin j9kd \sin \theta_0 \end{bmatrix} \quad 3.39$$

After that the desired signal correlation matrix, R_{ss} can be formulated based on array vector for desired signal, a_0

$$\bar{R}_{ss} = \bar{a}_0 \times \bar{a}_0^H \quad 3.40$$

and the interference signal correlation matrix, R_{ii} is formulated from array vector for undesired signal, \bar{A}

$$\bar{R}_{ii} = \bar{A} \times \bar{A}^H \quad 3.41$$

where $\bar{A} = [\bar{a}_m] = [a_1 \ a_2 \ \dots \ a_8]$ and m represents the number of interference signal. The noise correlation matrix, \bar{R}_{nn} , is formulated as below

$$\bar{R}_{nn} = 10^{-\frac{SNR}{10}} \quad 3.42$$

where the SNR is the signal to noise ratio. The noise signal, σ_n^2 is represented in the form of identity matrix and the size of the identity matrix, $N \times N$ is determined by number of element, $N = 10$.

$$\sigma_n^2 = \bar{R}_{nn} \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad 3.43$$

Each element in an array has an amplitude control and the phase control. The amplitude and phase control are known as weights, \bar{w} . The weights, \bar{w} is formulated as

$$\bar{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{10} \end{bmatrix} \quad 3.44$$

Then the fitness function signal to interference noise ratio (SINR) can be formulated as

$$SINR = \frac{\bar{w}^H \bar{a}_0 \bar{a}_0^H \bar{w}}{\bar{w}^H \bar{A} \bar{A}^H \bar{w} + \bar{w}^H \sigma_n^2 \bar{w}} \quad 3.45$$

The SKF, OBSKF and SKFMM algorithms are used to determine the best combination of array antenna weights that gives the maximum signal to interference plus noise ratio (SINR). Table 3.1 shows the parameters used for SKF, OBSKF and SKFMM. The dimensions used is dependent on the number elements used in an array. Since the number of elements used is 10, therefore, the number of dimensions is twice the number of elements in an array, which is 20, due to each weight of an element consists of an amplitude and a phase.

The simulation is executed 100 times in order to do comparison with previously publish work, Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO) (Zaharis & Yioultis, 2011). The comparison performed is the best, worst, mean and standard deviation values of SINR after 100 runs. Statistical analysis is also performed

using Wilcoxon Signed Ranked Test to compare between AMBPSO, SKF, OBSKF and SKFMM.

Table 3.1 Parameters of SKF, OBSKF and SKFMM

Parameters	SKF	OBSKF	SKFMM
Iteration	10000	10000	10000
Agents	100	100	20
Dimensions	20	20	20
$P(0)$	1000	1000	1000
Q	0.5	0.5	0.5
R	0.5	0.5	0.5
Jumping Rate, Jr	-	0.1	-

3.8.1 Application of SKF to Adaptive Beamforming

Figure 3.9 shows the flowchart of application of SKF in adaptive beamforming. For adaptive beamforming using SKF method, the array weights, \bar{w} , are randomly initialized in the range of $[-1,1]$ at the initial stage. Since the direction of arrival of the desired signal, θ_0 , and the interference signal, θ_m , is known, the desired signal correlation matrix, R_{SS} , and the undesired signal correlation matrix, R_{uu} , can be determined. After that, the fitness function, $SINR$, in equation 3.45 can be evaluated. The array weights which provide the maximum SINR value is identified as $X_{best}(t)$. The X_{true} represents the best overall array weights during the iterative process. In prediction, the following time-update equations

$$w_i(t|t+1) = w_i(t) \quad 3.46$$

$$P(t|t+1) = P(t) + Q \quad 3.47$$

are employed to make prediction of the array weights estimates, and the error covariance estimates. These estimates are known as *a priori* estimates. The $w_i(t|t+1)$ represents the array weights estimate and $w_i(t)$ represents the previous array weights estimate. The $P(t|t+1)$ represents the predicted error covariance due to estimate, $P(t)$ represents the previous error covariance estimate, and Q represents the process noise.

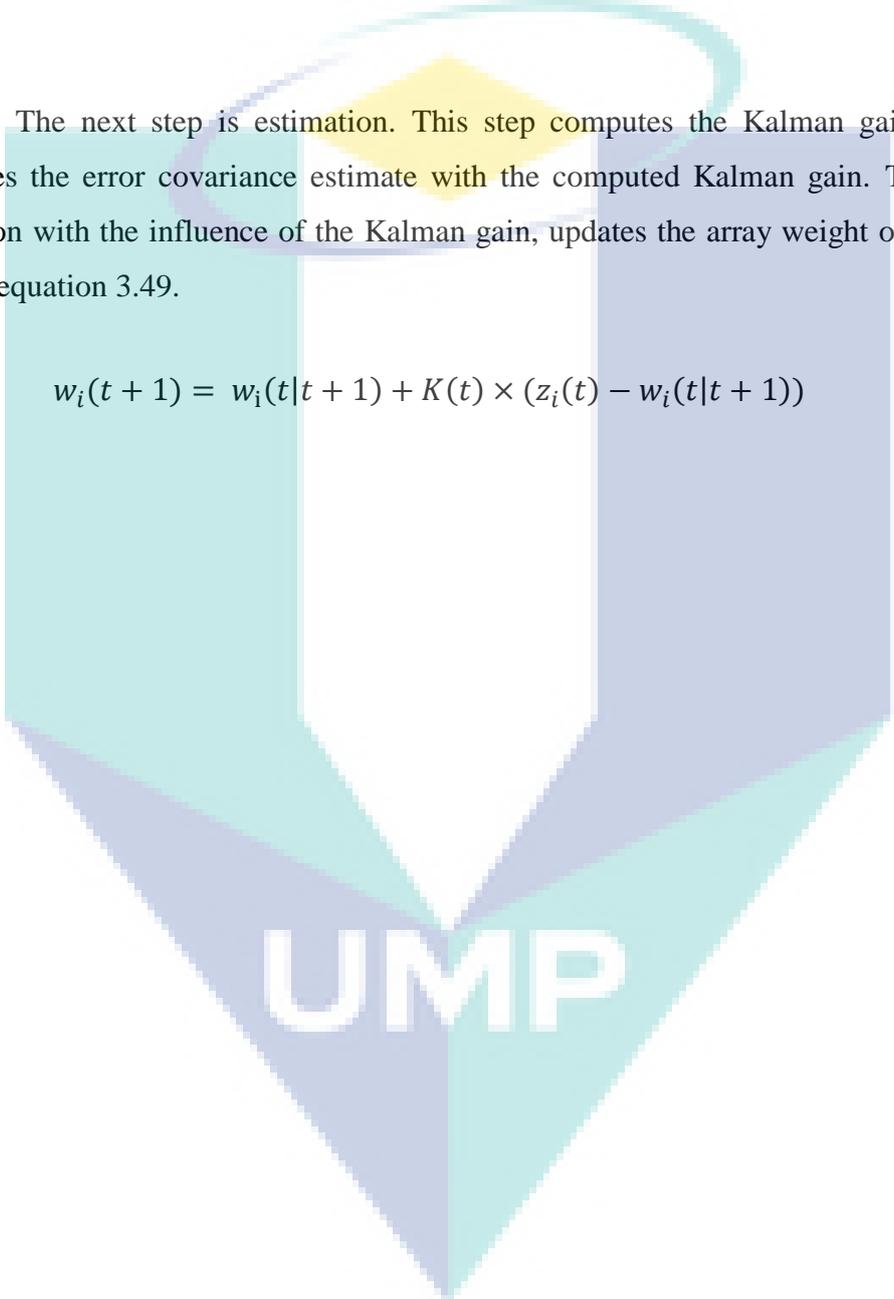
For measurement, each individual agents of array weights are simulated based on equation 3.48

$$z_i(t) = w_i(t|t + 1) + \sin(\text{rand} \times 2\pi) \times |w_i(t|t + 1) - X_{true}| \quad 3.48$$

where $z_i(t)$ represents the value of measure position for agents of array weights, and $w_i(t|t + 1)$ represents the array weights estimates. The measurement may take any random value from the predicted array weights, $w_i(t|t + 1)$, to the true array weights, X_{true} .

The next step is estimation. This step computes the Kalman gain, $K(t)$ and updates the error covariance estimate with the computed Kalman gain. The measure position with the influence of the Kalman gain, updates the array weight of each agent using equation 3.49.

$$w_i(t + 1) = w_i(t|t + 1) + K(t) \times (z_i(t) - w_i(t|t + 1)) \quad 3.49$$

The logo for UMP (Université de Moncton) is a large, stylized letter 'U' composed of several overlapping, semi-transparent shapes in shades of teal, light blue, and yellow. The letters 'UMIP' are printed in a bold, white, sans-serif font across the bottom of the 'U' shape.

UMIP

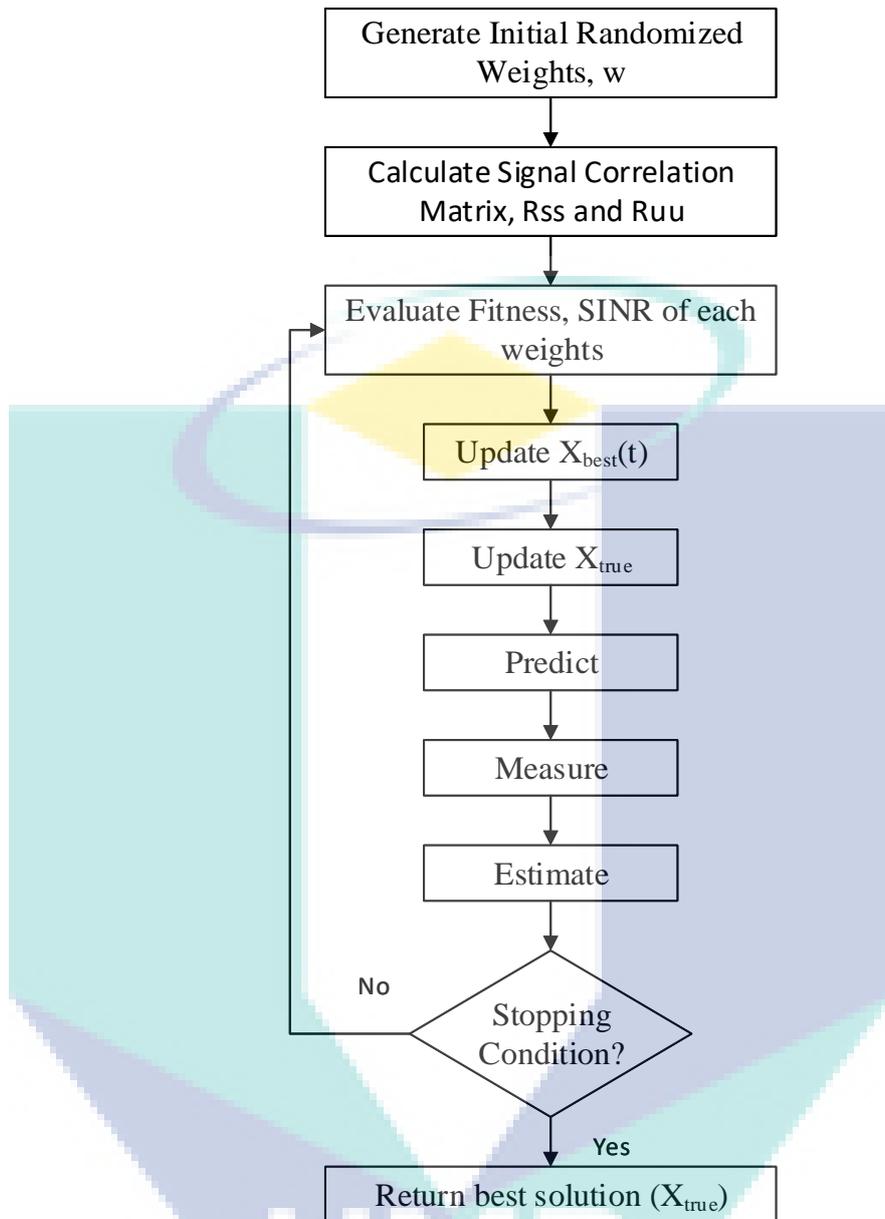


Figure 3.9 Flowchart of SKF for Adaptive Beamforming

3.8.2 Application of OBSKF to Adaptive Beamforming

Figure 3.10 shows the flowchart of application of SKF in adaptive beamforming. First, the array weights, \bar{w} , are randomly initialized in the range of $[-1,1]$ at the initial stage. Since the direction of arrival of the desired signal, θ_0 , and the interference signal, θ_m , is known, the desired signal correlation matrix, R_{ss} , and the undesired signal correlation matrix, R_{uu} , can be determined. After that, the fitness function, $SINR$, in equation 3.45 can be evaluated. Then, the best weight identified as $X_{best}(t)$ and X_{true} is identified as best overall weight. Then, the predict-measure-estimate steps is performed.

The Opposition-Based Learning technique is then applied to generate opposite weight, w_{ob} , from current weight, w . The SINR fitness obtained from the opposite weight, $SINR(w_{ob})$, is compared with the SINR fitness of the current weight, $SINR(w)$. If the SINR obtained using the opposite weight, $SINR(w_{ob})$, is better than the SINR obtained using the current weight, $SINR(w)$, then the current weight, w , will be replaced by the opposite weight, w_{ob} . If the opposite weight, $SINR(w_{ob})$, is worse than the SINR obtained using the current weight, $SINR(w)$, then the current weight, w , remains unchanged.

The execution of the OBL technique is controlled by the jumping rate, $Jr \in [0,1]$. The value of the jumping rate, Jr , is set to 0.1 as shown in Table 3.1. The opposite weight, w_{ob} , is calculated using dynamically updated interval boundaries $[a_j(t), b_j(t)]$ as follows

$$w_{ob_{i,j}} = [a_j(t) + b_j(t)] - w_{i,j} , \quad 3.50$$

$$a_j(t) = \min(w_{i,j}(t)), \quad b_j = \max(w_{i,j}(t))$$

$$i = 1,2, \dots N; \quad j = 1,2, \dots D$$

where $w_{i,j}$ represents the current weight, and $w_{ob_{i,j}}$ represents the opposite weight. The $a_j(t)$ and $b_j(t)$ are the lowest and the highest values of the j th dimension in the current population, respectively. The N represents the maximum number of agents and the D represents the maximum number of dimensions. The t represents the iteration number.

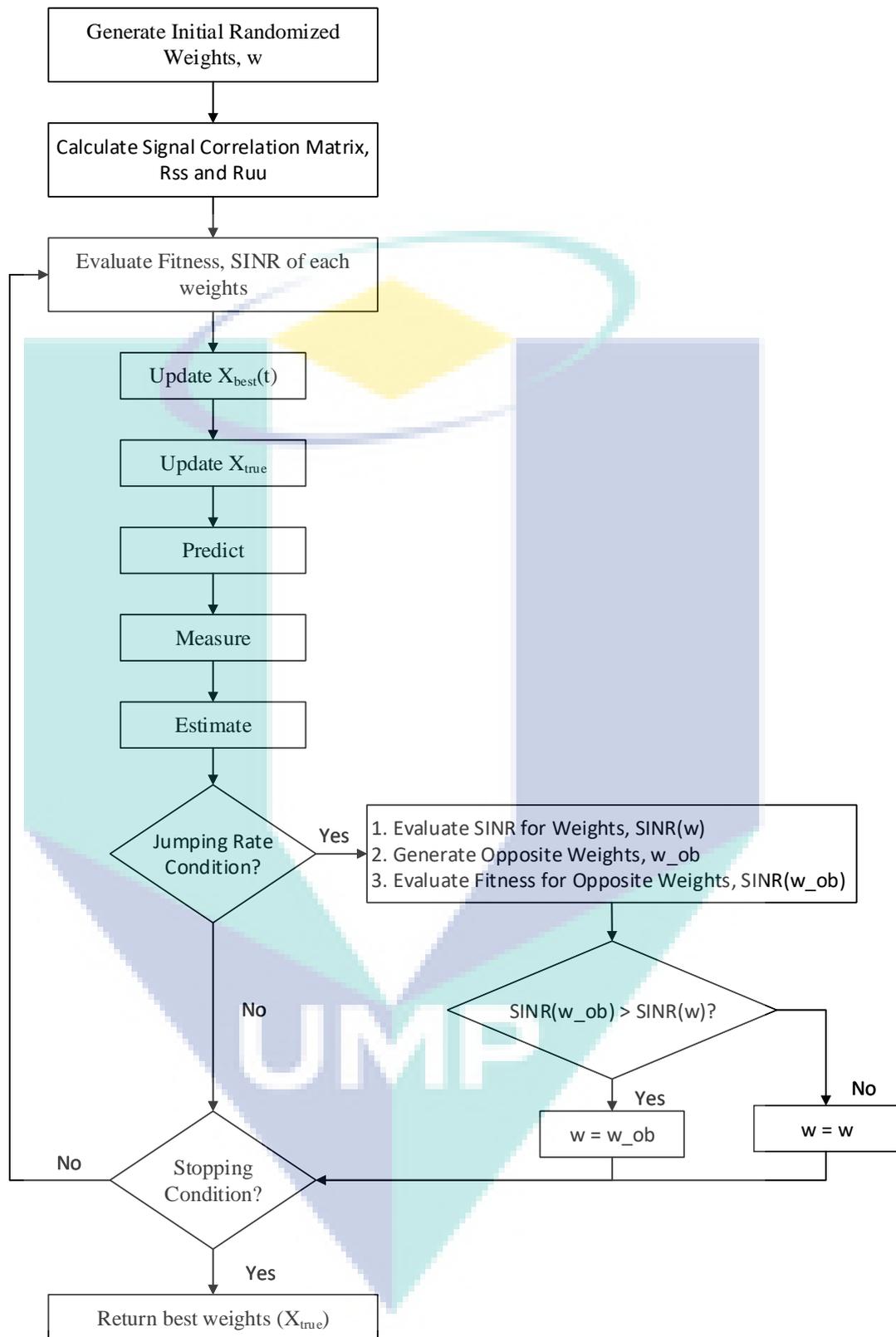


Figure 3.10 Flowchart of OBSKF for Adaptive Beamforming

3.8.3 Application of SKFMM to Adaptive Beamforming

Figure 3.11 shows the flowchart of application of Simulated Kalman Filter with Modified Measurement (SKFMM) in adaptive beamforming. First, the array weights, \bar{w} , are randomly initialized in the range of $[-1,1]$ at the initial stage. Since the direction of arrival of the desired signal, θ_0 , and the interference signal, θ_m , is known, the desired signal correlation matrix, R_{ss} , and the undesired signal correlation matrix, R_{uu} , can be determined. Then, the fitness function, $SINR$, in equation 3.45 can be evaluated. After that, the $X_{pbest}(t)$ is determined. The $X_{pbest}(t)$ represents the best-so-far weights for every agent. The best weight is identified as $X_{best}(t)$ and X_{true} is identified as best overall weight.

For SKFMM, the prediction step remains unchanged and follows exactly as in SKF and OBSKF. In SKFMM, only the measurement step is modified, to increase the exploration capabilities of Simulated Kalman Filter (SKF). The modified measurement update, $Z_{mm_i}(t)$, is as shown below

$$Z_{mm_i}(t) = w_i(t|t + 1) + \sin(rand \times 2\pi) \times |w_i(t|t + 1) - X_{true}| + \sin(rand \times 2\pi) \times |X_{pbest_i} - w_i(t|t + 1)| \quad 3.51$$

where $w_i(t|t + 1)$ represents the array weights estimate. After measurement update, the weights of the array will be update in the estimation process.

UMP

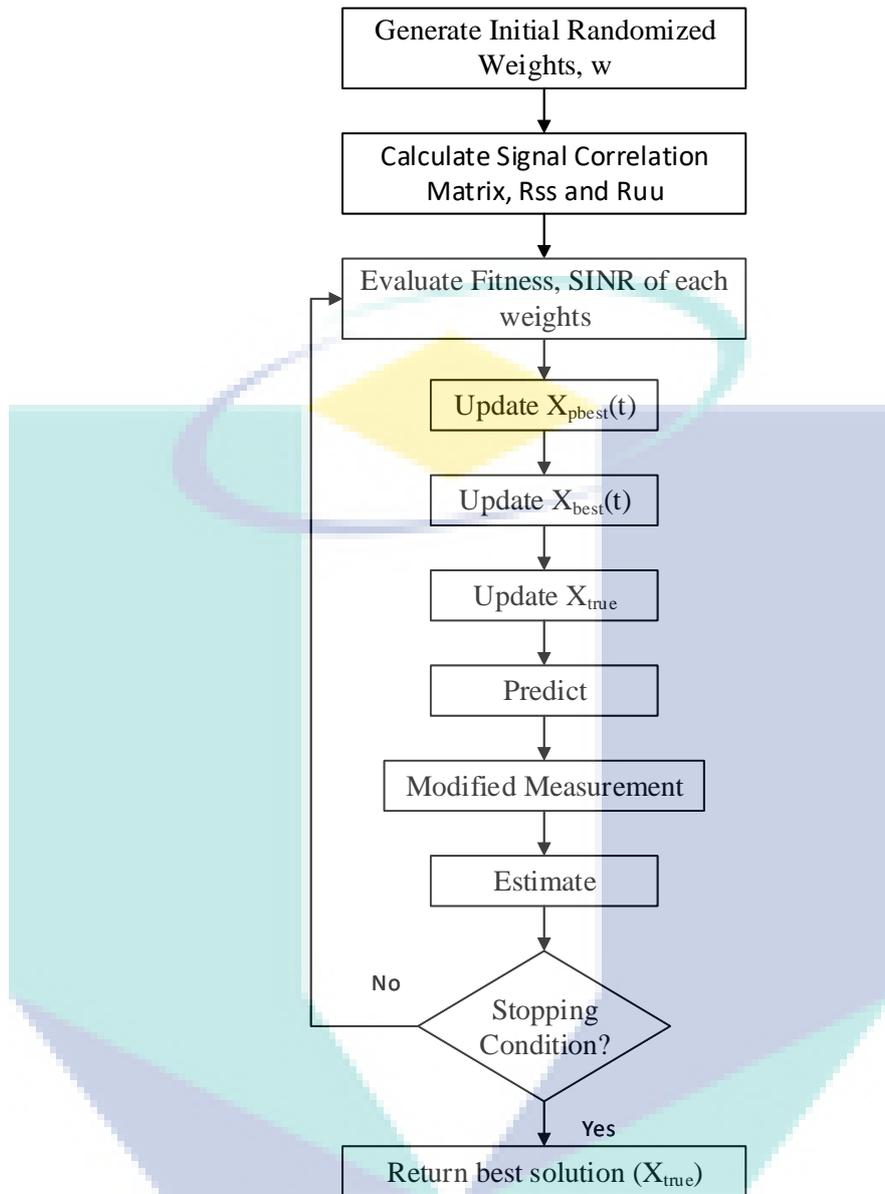


Figure 3.11 Flowchart of SKFMM for Adaptive Beamforming

3.9 Summary

This chapter introduced the basic concept of the Kalman Filter. Three optimization algorithms named Simulated Kalman Filter (SKF), Opposition-Based Simulated Kalman Filter (OBSKF), and the Simulated Kalman Filter with Modified Measurement (SKFMM) are introduced. The array system model and the formulation of fitness function is also explained.

CHAPTER 4

RESULT & DISCUSSION

4.1 Introduction

This chapter presents the results produced by Simulated Kalman Filter (SKF), Opposition-Based SKF (OBSKF) and SKF with Modified Measurement (SKFMM). The results are compared with previously published work, Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO) (Zaharis & Yioultsis, 2011). The work published by Zaharis and Yioultsis provides extensive analysis of statistical results and the experimental results show that AMBPSO have stable and good performance regardless of the signal to noise ratio (SNR) inputs, therefore, AMBPSO is a good algorithm for comparison with SKF algorithms for adaptive beamforming application. Furthermore, the experimental setup used by Zaharis and Yioultsis is easy to replicate and is useful for comparison.

4.2 Radiation Pattern

This subsection will introduce to the comparison of radiation pattern for AMBPSO (Zaharis & Yioultsis, 2011), SKF, OBSKF and SKFMM. Three criteria of the radiation pattern are compared; the null depth, the main beam accuracy and the maximum sidelobe level. The comparison is made for three different values of signal to noise, SNR.

For best performance, it is desirable have a much deeper null depth, minimal error between the main beam angle and the actual desired signal angle, and have a much lower maximum side lobe level. The results are chosen based on the best SINR value after several runs.

4.2.1 Signal to Noise Ratio, $SNR = 15dB$

Figure 4.1 shows the radiation pattern for AMBPSO (red), SKF (pink), OBSKF (blue) and SKFMM (green) for signal to noise, SNR, input of 15 dB.

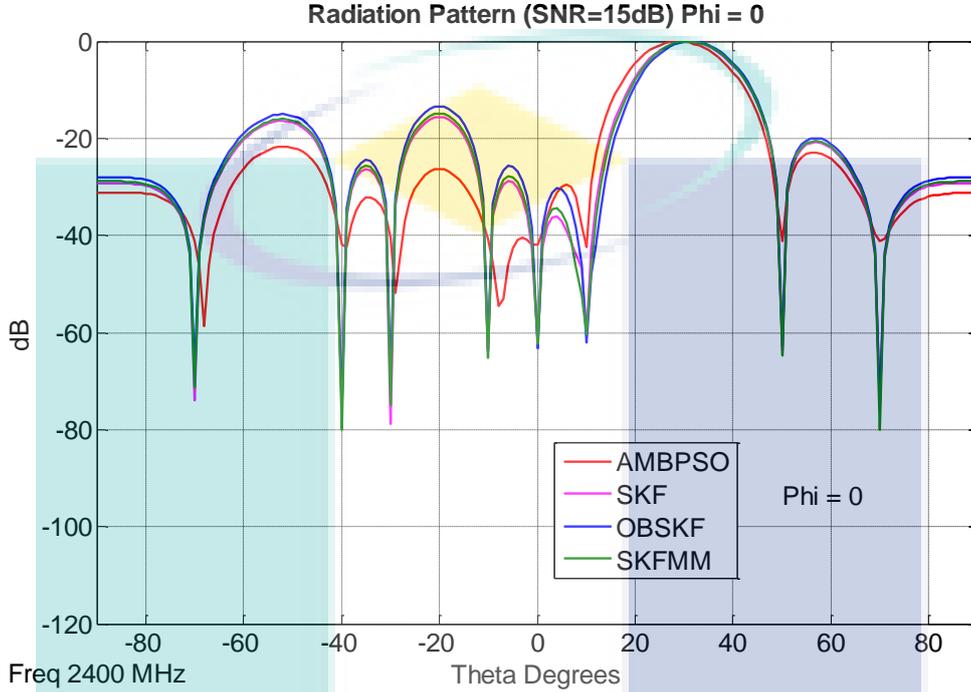


Figure 4.1 Radiation Pattern for $SNR = 15dB$

From Figure 4.1, the null depth, main beam accuracy and the maximum side lobe level for AMBPSO, SKF, OBSKF and SKFMM for interference signal, $\theta_m \in \{-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ\}$ at $SNR = 15dB$, are tabulated in Table 4.1, Table 4.2 and Table 4.3 respectively.

Table 4.1 Null Depth (dB) for $SNR = 15dB$

Null Depth (dB)	Interference Signal Angle ($^\circ$)							
	-70°	-40°	-30°	-10°	0°	10°	50°	70°
AMBPSO	-58.67	-41.82	-51.88	-54.56	-41.79	-42.41	-41.28	-41.12
SKF	-74.03	-74.76	-78.84	-63.9	-62.19	-60.39	-63.87	-74.77
OBSKF	-67.91	-75.95	-69.88	-63.58	-63.21	-62.04	-62.85	-75.61
SKFMM	-71.39	-80.1	-74.96	-65.14	-62.4	-60.38	-64.66	-80.06

Table 4.1 shows comparison of null depth at the interference angle. The highlighted box represents the best null depth produced by the algorithms. All three SKF algorithms, SKF, OBSKF and SKFMM, can produce much deeper nulls compare to AMBPSO for adaptive beamforming application. SKFMM produce deeper nulls at

interference angle, $\theta_m = [-40^\circ, -10^\circ, 0^\circ, 50^\circ, 70^\circ]$ compared to SKF which produce deeper nulls at interference angle, $\theta_m = [-70^\circ, -30^\circ, 10^\circ]$. SKFMM produce deeper nulls at interference angle, $\theta_m = [-70^\circ, -40^\circ, -30^\circ, -10^\circ, 50^\circ, 70^\circ]$ compared to OBSKF which produce deeper nulls at interference angle, $\theta_m = [0^\circ, 10^\circ]$. Overall, SKFMM mostly able to produce much deeper nulls compared to SKF and OBSKF.

Table 4.2 Accuracy of the Main Beam for $SNR = 15dB$

Algorithm	Main Beam Angle (°)
AMBPSO	29°
SKF	31°
OBSKF	31°
SKFMM	31°

Table 4.2 shows the accuracy of the main beam radiating towards the desired signal angle, $\theta_0 = 30^\circ$, for $SNR = 15dB$. AMBPSO, SKF, OBSKF, and SKFMM able to produce a radiation pattern with the main lobe very close to the desired signal angle. The main beam is only off by 1° .

Table 4.3 Maximum Sidelobe Level for $SNR = 15dB$

Algorithm	Maximum Sidelobe Level (dB)
AMBPSO	-21.66
SKF	-15.58
OBSKF	-13.42
SKFMM	-14.89

Table 4.3 shows the maximum sidelobe level for AMBPSO, SKF, OBSKF, and SKFMM for $SNR = 15dB$. From Table 4.3, the previously published algorithm, AMBPSO, can produce much lower maximum side lobe level compared to SKF, OBSKF and SKFMM. SKF comes in second best, SKFMM comes in third and OBSKF is the worst among the 4 algorithms.

4.2.2 Signal to Noise Ratio, $SNR = 30dB$

Figure 4.2 shows the radiation pattern for AMBPSO (red), SKF (pink), OBSKF (blue) and SKFMM (green) for signal to noise, SNR, input of $30 dB$.

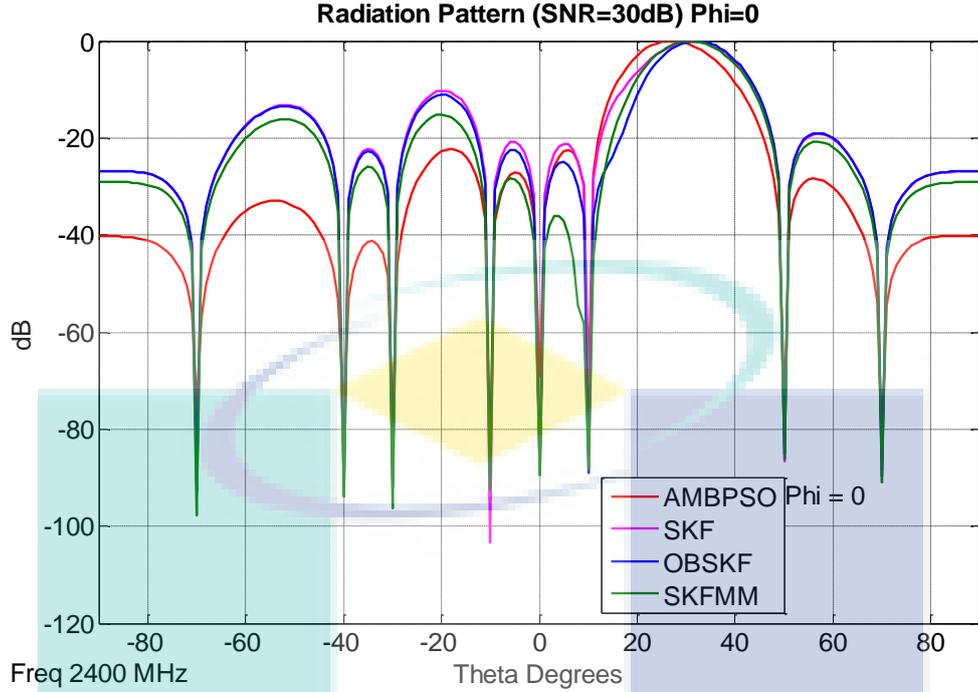


Figure 4.2 Radiation Pattern for $SNR = 30dB$

From Figure 4.2, the null depth, main beam accuracy and the maximum side lobe level for AMBPSO, SKF, OBSKF and SKFMM for interference signal, $\theta_m \in \{-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ\}$ at $SNR = 30dB$, are tabulated in Table 4.4, Table 4.5 and Table 4.6 respectively.

Table 4.4 Null Depth (dB) for $SNR = 30dB$

Null Depth (dB)	Interference Angle ($^\circ$)							
	-70°	-40°	-30°	-10°	0°	10°	50°	70°
AMBPSO	-75.22	-76.25	-79.14	-94.66	-69.09	-69.84	-75.13	-83.36
SKF	-87.57	-86.4	-90.57	-103.3	-89.27	-85.72	-86.67	-88.43
OBSKF	-92.78	-93.16	-92.03	-89.38	-88.67	-88.97	-85.24	-89.75
SKFMM	-97.67	-93.89	-96.33	-92.35	-89.57	-87.99	-86.2	-90.94

Table 4.4 shows the comparison of null depth at the interference angle. The highlighted box represents the best null depth obtained by the algorithms. Again, all three SKF algorithms, SKF, OBSKF and SKFMM, can produce much deeper nulls compared to the previously published algorithm, AMBPSO for adaptive beamforming application. SKFMM produce deeper nulls at interference angle, $\theta_m = [-70^\circ, -40^\circ, -30^\circ, 0^\circ, 10^\circ, 70^\circ]$ compared to SKF which produce deeper nulls at interference angle, $\theta_m = [-10^\circ, 50^\circ]$. SKFMM produce deeper nulls at interference angle, $\theta_m = [-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 50^\circ, 70^\circ]$ compared to OBSKF which produce deeper

nulls at interference angle, $\theta_m = 10^\circ$. SKFMM is shown to produce the most deep nulls compared to SKF and OBSKF.

Table 4.5 Accuracy of Main Beam for $SNR = 30dB$

Algorithm	Main Beam Angle ($^\circ$)
AMBPSO	27 $^\circ$
SKF	31 $^\circ$
OBSKF	32 $^\circ$
SKFMM	31 $^\circ$

Table 4.5 shows the accuracy of the main beam radiating towards the desired signal angle, $\theta_0 = 30^\circ$, for $SNR = 30dB$. AMBPSO performs the worse compared to SKF, OBSKF and SKFMM with the main beam angle off by 3 $^\circ$. OBSKF main beam angle is off by 2 $^\circ$. The main beam angle for both SKF and OBSKF is off by 1 $^\circ$.

Table 4.6 Maximum Sidelobe Level for $SNR = 30dB$

Algorithm	Maximum Sidelobe Level (dB)
AMBPSO	-22.25
SKF	-10.21
OBSKF	-11.05
SKFMM	-15.18

Table 4.6 shows the maximum sidelobe level for AMBPSO, SKF, OBSKF, and SKFMM for $SNR = 30dB$. From Table 4.6, the previously published algorithm, AMBPSO, can produce much lower maximum side lobe level compared to SKF, OBSKF and SKFMM. SKFMM comes in second, followed by OBSKF and then SKF.

4.2.3 Signal to Noise Ratio, $SNR = 50dB$

Figure 4.3 shows the radiation pattern for AMBPSO (red), SKF (pink), OBSKF (blue) and SKFMM (green) for signal to noise, SNR, input of 50 dB.

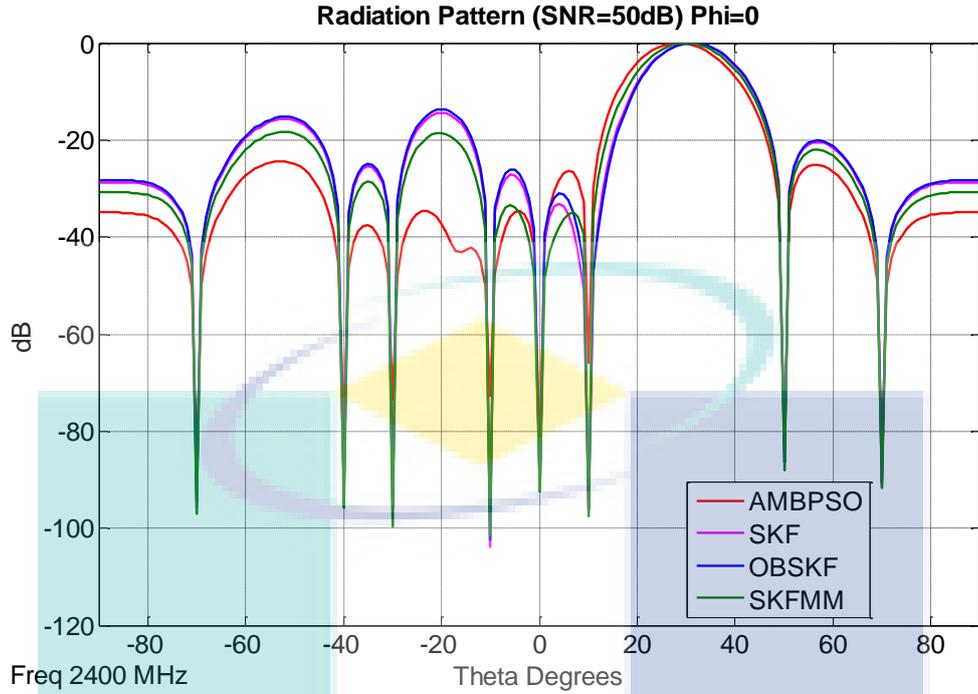


Figure 4.3 Radiation Pattern for $SNR = 50dB$

From Figure 4.3, the null depth, main beam accuracy and the maximum side lobe level for AMBPSO, SKF, OBSKF and SKFMM for interference signal, $\theta_m \in \{-70^\circ, -40^\circ, -30^\circ, -10^\circ, 0^\circ, 10^\circ, 50^\circ, 70^\circ\}$ at $SNR = 50dB$, are tabulated in Table 4.7, Table 4.8 and Table 4.9 respectively.

Table 4.7 Null Depth (dB) for $SNR = 50dB$

Null Depth (dB)	Interference Angle ($^\circ$)							
	-70°	-40°	-30°	-10°	0°	10°	50°	70°
AMBPSO	-86.55	-77.81	-73.37	-72.66	-85.97	-65.94	-84.05	-84.53
SKF	-95.75	-92.66	-96.09	-103.9	-92.52	-95.82	-86.6	-90.93
OBSKF	-94.72	-93.16	-95.45	-102.5	-92.54	-96.41	-86.37	-90.53
SKFMM	-97.09	-95.94	-99.68	-101.3	-92.49	-97.49	-88.05	-91.74

Table 4.7 shows the comparison of null depth at the interference angle. The highlighted box represents the best null depth obtained by the algorithms. Again, all three SKF algorithms, SKF, OBSKF and SKFMM, can produce deeper nulls compared to AMBPSO for adaptive beamforming application. SKFMM produce deeper nulls at interference angle, $\theta_m = [-70^\circ, -40^\circ, -30^\circ, 10^\circ, 50^\circ, 70^\circ]$ compared to SKF and OBSKF which produce deeper nulls at interference angle, $\theta_m = [-10^\circ, 0^\circ]$. Once again, SKFMM is shown to be superior in producing deeper nulls compared to SKF and OBSKF.

Table 4.8 Accuracy of Main Beam for $SNR = 50dB$

Algorithm	Main Beam Angle (°)
AMBPSO	28°
SKF	31°
OBSKF	31°
SKFMM	30°

Table 4.8 shows the accuracy of the main beam radiating towards the desired signal angle, $\theta_0 = 30^\circ$, for $SNR = 50dB$. AMBPSO performs worse compared SKF, OBSKF and SKFMM with the main beam angle off by 2° . The main beam angle for both SKF and OBSKF is off by 1° . SKFMM, however, produces a main beam accurately towards the desired angle at 30° .

Table 4.9 Maximum Sidelobe Level for $SNR = 50dB$

Algorithm	Maximum Sidelobe Level (dB)
AMBPSO	-24.29
SKF	-14.40
OBSKF	-13.68
SKFMM	-18.27

Table 4.9 shows the maximum sidelobe level for AMBPSO, SKF, OBSKF, and SKFMM for $SNR = 50dB$. Yet again, AMBPSO, produce much lower maximum side lobe level SKF, OBSKF and SKFMM. SKFMM comes in second and followed by SKF and then OBSKF.

4.3 Convergence Curve

This subsection presents the convergence curve comparison between SKF, OBSKF and SKFMM for $SNR(dB) = \{15, 30, 50\}$. The convergence curve obtained is based on the radiation pattern shown in sub-chapter 4.2.

4.3.1 Signal to Noise Ratio, $SNR = 15dB$

Figure 4.4 shows the convergence curve for SKF, OBSKF and SKFMM for $SNR = 15dB$.

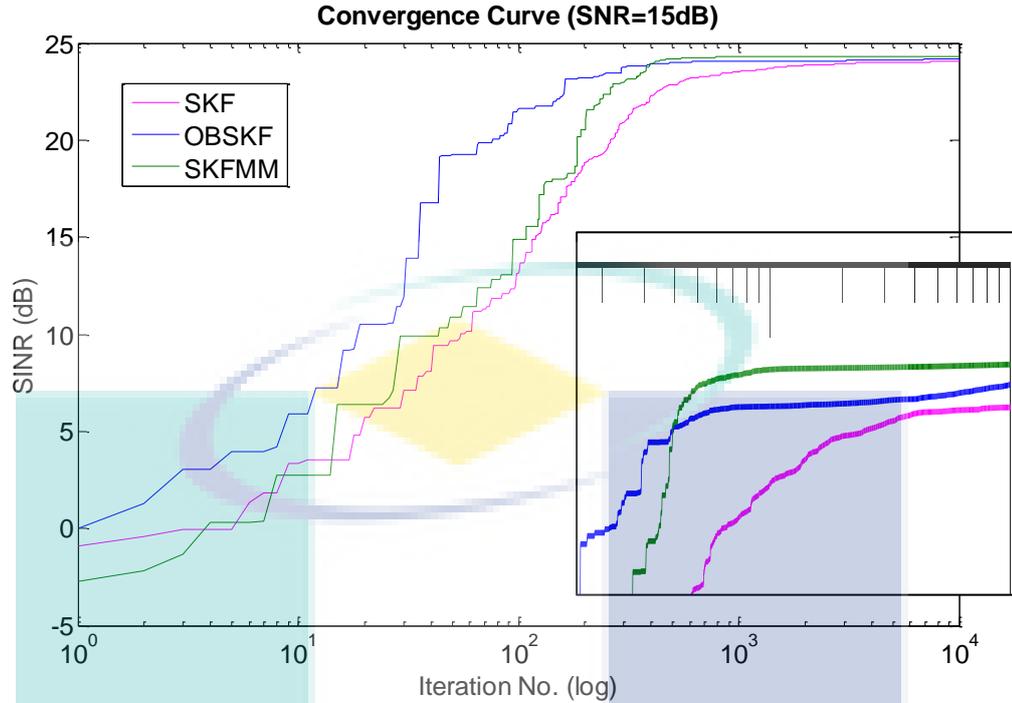


Figure 4.4 Convergence Curve for $SNR = 15dB$

From Figure 4.4, SKFMM is shown to produce much higher SINR value, followed by OBSKF and then SKF. OBSKF shows faster convergence for the first 800 iterations compared to SKFMM and SKF. After 800 iterations, OBSKF begins to converge slowly to the maximum value but not enough to get a higher SINR value than SKFMM after 10000 iterations. SKFMM, however, can converge to the maximum value within 1000 iterations. SKF, on the other hand, showed slow convergence and can't to reach its maximum value. The maximum SINR value after 10000 iterations for SKF, OBSKF and SKFMM at $SNR = 15dB$ are recorded in Table 4.10.

Table 4.10 Maximum SINR Value After 10000 Iteration for $SNR = 15dB$

Algorithm	Maximum SINR Value (dB)
SKF	24.04
OBSKF	24.19
SKFMM	24.33

From Table 4.10, SKFMM produce much higher SINR value after 10000 iterations compared to OBSKF and SKF. OBSKF produced the second highest SINR value and then followed by SKF.

4.3.2 Signal to Noise Ratio, $SNR = 30dB$

Figure 4.5 shows the convergence curve for SKF, OBSKF and SKFMM for $SNR = 30dB$.

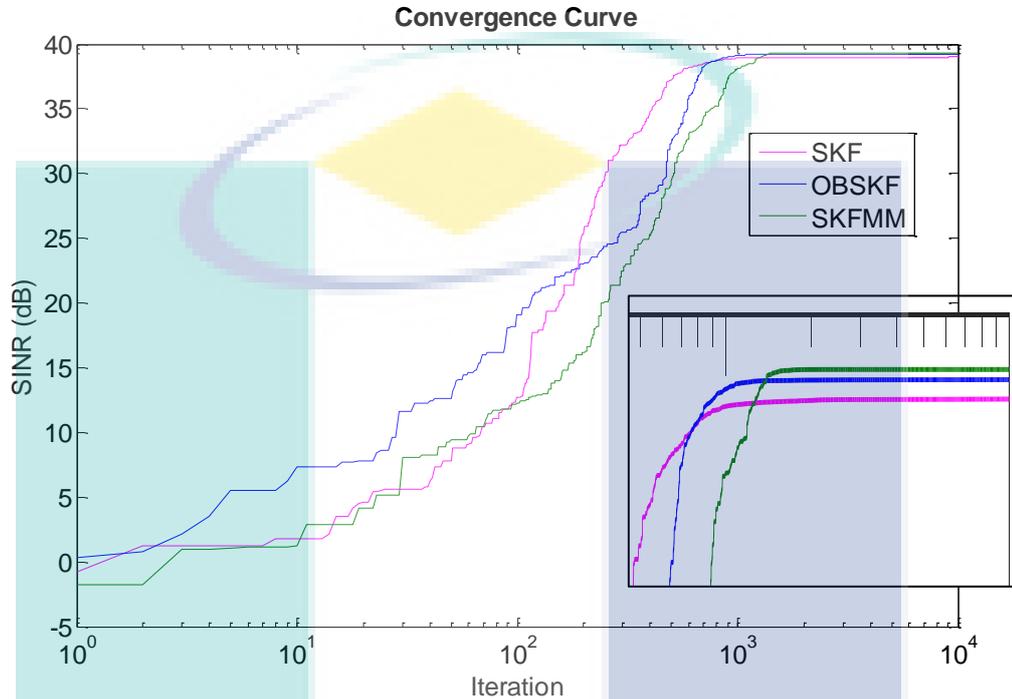


Figure 4.5 Convergence Curve for $SNR = 30dB$

From Figure 4.5, SKFMM, again, is shown to produce much higher SINR value, followed by OBSKF and then, SKF. SKF and OBSKF is shown to converge faster and then, stagnate at local optimum. SKFMM takes more iteration before reaching its maximum SINR value. All these algorithms converge within 2000 iterations. Table 4.11 shows the maximum SINR value reached after 10000 iterations for $SNR = 30dB$.

Table 4.11 Maximum SINR Value After 10000 Iteration for $SNR = 30dB$

Algorithm	Maximum SINR Value (dB)
SKF	39.00
OBSKF	39.23
SKFMM	39.35

From Table 4.11, the SKFMM algorithm can produce much higher SINR values compared to OBSKF and SKF algorithms at $SNR = 30dB$. OBSKF algorithm produces second highest SINR values and followed by SKF algorithm.

4.3.3 Signal to Noise Ratio, $SNR = 50dB$

Figure 4.6 shows the convergence curve for SKF, OBSKF and SKFMM for $SNR = 50dB$.

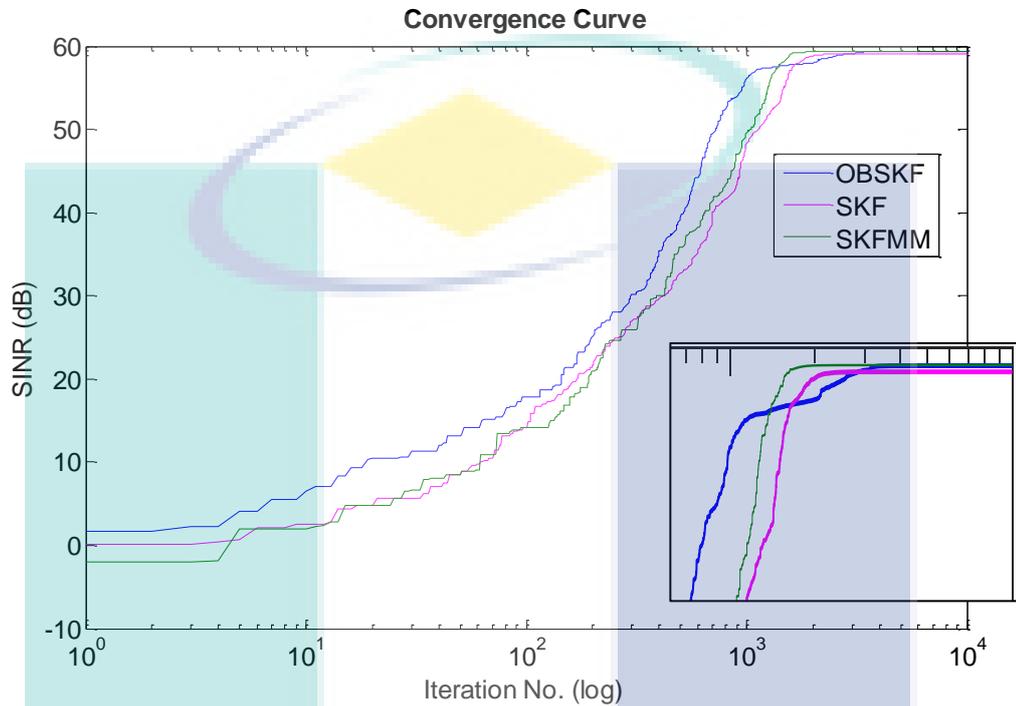


Figure 4.6 Convergence Curve for $SNR = 50dB$

From Figure 4.6, SKFMM algorithm is shown to converge faster than SKF and OBSKF. SKFMM converges within 2000 iterations, SKF converges after 2000 iterations and OBSKF converge to maximum after 3000 iterations. Both SKFMM and OBSKF produces a maximum value very close to each after 10000 iterations. Table 4.12 shows the maximum SINR value reached after 10000 iterations for $SNR = 50dB$.

Table 4.12 Maximum SINR Value After 10000 Iteration for $SNR = 50dB$

Algorithm	Maximum SINR Value (dB)
SKF	59.07
OBSKF	59.30
SKFMM	59.33

From Table 4.12, SKFMM once again produce higher SINR value compared to SKF and OBSKF. However, the difference between the maximum SINR value for OBSKF and SKFMM is very small.

4.4 Statistical Results

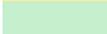
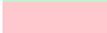
The experiment is repeated 100 times for $SNR(dB) = \{-20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60\}$. After 100 runs, the best SINR values, worst SINR values, mean SINR values and the standard deviation values are recorded. Then statistical analysis is performed using Wilcoxon Signed Ranked Test.

4.4.1 AMBPSO vs SKF

Table 4.13 shows the best, worst, mean and standard deviation (STD) of SINR(dB) values after the experiment is repeated 100 times. Comparison is made between AMBPSO and SKF for various SNR(dB) input values.

Table 4.13 Best, Worst, Mean and Standard Deviation(STD) of SINR(dB) for AMBPSO vs SKF

SNR (dB)	AMBPSO				SKF			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
-20	-10.0522	-10.0548	-10.0523	0.0004	-10.0522	-10.0549	-10.0523	0.0004
-15	-5.1395	-5.1512	-5.1399	0.0020	-5.1395	-5.1464	-5.1398	0.0011
-10	-0.2975	-0.3692	-0.2998	0.0098	-0.2975	-0.3028	-0.2977	0.0007
-5	4.5321	4.3422	4.5269	0.0218	4.5321	4.3828	4.5295	0.0175
0	9.4241	8.5481	9.3749	0.1643	9.4241	9.3237	9.4208	0.0142
5	14.3768	12.0647	14.2676	0.3628	14.3768	14.1437	14.3631	0.0306
10	19.3598	15.1371	19.2810	0.4463	19.3590	18.8208	19.2736	0.0954
15	24.3542	16.5370	24.1008	1.0643	24.3535	23.0034	24.1049	0.2935
20	29.3509	17.2416	29.0332	1.2290	29.3469	28.0453	29.0012	0.3236
25	34.3515	22.4314	33.6680	1.4163	34.3504	32.7718	33.9332	0.3790
30	39.3341	30.2715	38.7648	0.8722	39.3477	37.5436	38.9249	0.3946
35	44.3440	32.6393	43.1564	1.5287	44.3510	41.8109	43.9153	0.4384
40	49.3461	36.6898	48.1781	1.2409	49.3456	46.3051	48.9181	0.4529
45	54.3358	43.2386	52.5134	1.5788	54.3504	51.9201	53.8808	0.4746
50	59.3317	47.6499	58.3221	1.7439	59.3355	55.8211	58.8351	0.6189
55	64.3458	52.7630	63.0660	1.6119	64.3468	62.0234	63.8755	0.5364
60	69.3468	56.3379	67.5858	1.8369	69.3396	67.7525	68.9202	0.4239

Neutral 
 Good 
 Bad 

From Table 4.13, the best SINR values for both AMBPSO and SKF is the same for $SNR(dB) = [-20, -15, -10, -5, 0, 5]$. AMBPSO can get better maximum SINR values for input $SNR(dB) = [10, 15, 20, 25, 40, 60]$ and SKF produce best SINR values at $SNR(dB) = [30, 35, 45, 50, 55]$. On the other hand, AMBPSO produces the lowest

worst SINR values compared to SKF's worst SINR for almost all the SNR values except for $SNR = -20dB$. Since the worst SINR values for SKF is higher than the worst SINR values of AMBPSO, SKF can produce much higher SINR mean values for almost all the SINR input except for $SNR(dB) = [10, 20]$ and for $SNR(dB) = -20$, the $SINR(dB)$ value is the same for both AMBPSO and SKF. Based on the standard deviation (STD) values for both AMBPSO and SKF, SKF produces much lower STD values compared to AMBPSO. This proves that the SINR values produced by SKF is much more consistent and does not fluctuate a lot compared to AMBPSO. The consistency of SKF in producing high mean SINR values and high values for worst SINR can lead to higher mean SINR values.

Based on the mean results for both AMBPSO and SKF from Table 4.13, Wilcoxon Signed Ranked Test statistical analysis is performed. Wilcoxon Signed Rank Test calculates the sum of ranks where the first algorithm outperforms the second, R^+ and the sum of ranks where the second algorithm outperforms the second, R^- . Table 4.14 shows the sum of ranks where AMBPSO outperforms SKF, R^+ , and sum of ranks where SKF outperforms AMBPSO, R^- .

Table 4.14 Sum of Ranks for AMBPSO vs SKF

AMBPSO vs SKF	Sum of Ranks
AMBPSO Outperforms SKF, R^+	14
SKF Outperforms AMBPSO, R^-	139

After the sum of ranks are obtained, the test statistic, T is chosen. The test statistic, T value is determined from the smallest value between R^+ and R^- as shown in equation 4.1.

$$T = \min(R^+, R^-) \quad 4.1$$

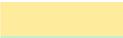
The null hypothesis, H_0 states that the algorithms compared are equals with no significant difference and H_1 states that the algorithms compared are not equals and have significant difference. In Wilcoxon signed ranked test, the null hypothesis, H_0 , is rejected when the test statistic, T , is less than the critical value, T_0 . By referring Appendix A, at $\alpha = 0.05$ and number of test inputs, $n = 17$, the critical value, $T_0 = 35$. Since, the test statistic, $T = 14$ and is less than the critical value, T_0 , therefore, the null hypothesis can be rejected and there is significant difference between SKF and AMBPSO.

4.4.2 AMBPSO vs OBSKF

Table 4.15 shows the best, worst, mean and standard deviation (STD) of SINR(dB) values after the experiment is repeated 100 times. Comparison is made between AMBPSO and OBSKF for various SNR(dB) input values.

Table 4.15 Best, Worst, Mean and Standard Deviation(STD) of SINR(dB) for AMBPSO vs OBSKF

SNR (dB)	AMBPSO				OBSKF			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
-20	-10.0522	-10.0548	-10.0523	0.0004	-10.0522	-10.0522	-10.0522	1.39E-07
-15	-5.1395	-5.1512	-5.1399	0.0020	-5.1395	-5.1397	-5.1395	2.29E-05
-10	-0.2975	-0.3692	-0.2998	0.0098	-0.2975	-0.2975	-0.2975	2.20E-07
-5	4.5321	4.3422	4.5269	0.0218	4.5321	4.5316	4.5321	4.33E-05
0	9.4241	8.5481	9.3749	0.1643	9.4241	9.4177	9.4240	0.0007
5	14.3768	12.0647	14.2676	0.3628	14.3768	14.2131	14.3677	0.0245
10	19.3598	15.1371	19.2810	0.4463	19.3564	18.8957	19.2854	0.0823
15	24.3542	16.5370	24.1008	1.0643	24.3538	23.5108	24.1784	0.1529
20	29.3509	17.2416	29.0332	1.2290	29.3438	27.1338	28.9768	0.4337
25	34.3515	22.4314	33.6680	1.4163	34.3491	31.5473	33.9442	0.3964
30	39.3341	30.2715	38.7648	0.8722	39.3467	37.3418	38.9474	0.3955
35	44.3440	32.6393	43.1564	1.5287	44.3499	41.7999	43.8989	0.4392
40	49.3461	36.6898	48.1781	1.2409	49.3341	47.0214	48.9543	0.3544
45	54.3358	43.2386	52.5134	1.5788	54.3425	52.6826	53.9350	0.3521
50	59.3317	47.6499	58.3221	1.7439	59.3478	56.6860	58.8876	0.4814
55	64.3458	52.7630	63.0660	1.6119	64.3424	61.9313	63.9258	0.4389
60	69.3468	56.3379	67.5858	1.8369	69.3464	66.8448	68.8720	0.5243

Neutral	
Good	
Bad	

Based on Table 4.15, the best SINR values for both AMBPSO and OBSKF is the same for $SNR(dB) = [-20, -15, -10, -5, 0, 5]$. AMBPSO has better maximum SINR values for input $SNR(dB) = [10, 15, 20, 25, 40, 55, 60]$ and OBSKF produce maximum best SINR values at $SNR(dB) = [30, 35, 45, 50]$. This shows that AMBPSO can mostly get better maximum SINR values compared to OBSKF. On the other hand, AMBPSO produces the lowest worst SINR values compared to OBSKF's worst SINR values for all the SNR input values. OBSKF produces higher mean SINR values for all the SNR input compared to AMBPSO. Based on the standard deviation (STD) values for both AMBPSO and OBSKF, OBSKF produces much lower STD values compared to AMBPSO. This proves that the SINR values produced by OBSKF is much more stable and does not fluctuate a lot compared to AMBPSO.

Based on the mean results for both AMBPSO and OBSKF from Table 4.15, Wilcoxon Signed Ranked test statistical analysis is performed. Wilcoxon Signed Rank test calculates the sum of ranks where the first algorithm outperforms the second, R^+ and the sum of ranks where the second algorithm outperforms the first, R^- . Table 4.16 shows that the sum of ranks where AMBPSO outperforms OBSKF, R^+ and sum of ranks where OBSKF outperforms AMBPSO, R^- .

Table 4.16 Sum of Ranks for AMBPSO vs OBSKF

AMBPSO vs OBSKF	Sum of Ranks
AMBPSO Outperform OBSKF, R^+	7
OBSKF Outperform AMBPSO, R^-	146

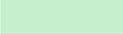
After the sum of ranks are obtained, the test statistic, T is chosen. The test statistic, T value is the smallest value between R^+ and R^- as shown in equation 4.1. The null hypothesis states that the algorithms compared are equal with no significant difference. In Wilcoxon signed ranked test, the null hypothesis is rejected when the test statistic, T is less than the critical value, T_0 . By referring Appendix A, at $\alpha = 0.05$ and number of test inputs, $n = 17$, the critical value, $T_0 = 35$. Since, the test statistic, $T = 7$ and is less than the critical value, T_0 , therefore, the null hypothesis can be rejected and there is significant difference between OBSKF and AMBPSO.

4.4.3 AMBPSO vs SKFMM

Table 4.17 shows the best, worst, mean and standard deviation (STD) of SINR(dB) values after the experiment is repeated 100 times. Comparison is made between AMBPSO and SKF for various SNR(dB) input values.

Table 4.17 Best, Worst, Mean and Standard Deviation(STD) of SINR(dB) for AMBPSO vs SKFMM

SNR (dB)	AMBPSO				SKFMM			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
-20	-10.0522	-10.0548	-10.0523	0.0004	-10.0522	-10.0522	-10.0522	1.50E-14
-15	-5.1395	-5.1512	-5.1399	0.0020	-5.1395	-5.1395	-5.1395	6.15E-15
-10	-0.2975	-0.3692	-0.2998	0.0098	-0.2975	-0.2975	-0.2975	5.04E-16
-5	4.5321	4.3422	4.5269	0.0218	4.5321	4.5321	4.5321	7.90E-14
0	9.4241	8.5481	9.3749	0.1643	9.4241	9.4241	9.4241	2.12E-06
5	14.3768	12.0647	14.2676	0.3628	14.3768	14.3601	14.3724	0.0039
10	19.3598	15.1371	19.2810	0.4463	19.3592	19.0344	19.2735	0.0781
15	24.3542	16.5370	24.1008	1.0643	24.3515	23.5126	24.1363	0.1903
20	29.3509	17.2416	29.0332	1.2290	29.3507	27.3467	29.0632	0.3056
25	34.3515	22.4314	33.6680	1.4163	34.3434	32.8106	33.9984	0.3204
30	39.3341	30.2715	38.7648	0.8722	39.3473	37.5921	39.0208	0.3171
35	44.3440	32.6393	43.1564	1.5287	44.3484	42.2910	43.9895	0.4356
40	49.3461	36.6898	48.1781	1.2409	49.3502	48.0500	49.0078	0.3099
45	54.3358	43.2386	52.5134	1.5788	54.3408	52.7866	54.0659	0.3231
50	59.3317	47.6499	58.3221	1.7439	59.3485	56.9759	59.0361	0.3208
55	64.3458	52.7630	63.0660	1.6119	64.3512	62.4793	63.9707	0.3903
60	69.3468	56.3379	67.5858	1.8369	69.3466	66.8526	69.0277	0.3680

Neutral 
 Good 
 Bad 

Based on Table 4.17, the best SINR values for both AMBPSO and SKFMM is the same for $SNR(dB) = [-20, -15, -10, -5, 0, 5]$. AMBPSO has the best SINR values for input $SNR(dB) = [10, 15, 20, 25, 60]$ and SKFMM produce best SINR values at $SNR(dB) = [30, 35, 40, 45, 50, 55,]$. For the worst SINR values, AMBPSO produces much lower worst SINR values compared SKFMM algorithm. The high values for worst SINR leads to much higher values for mean SINR. Moreover, the difference between best SINR and worst SINR for AMBPSO is much larger compared to SKFMM. This proves that SKFMM is more consistent than AMBPSO. The consistency of SKFMM can be further proven with standard deviation (STD) value obtained. The standard deviation of SKFMM is much lower than the standard deviation for AMBPSO, proves that SKFMM can produce much more consistent high SINR values than AMBPSO.

Based on the mean results for both AMBPSO and SKFMM from Table 4.17, Wilcoxon Signed Ranked test statistical analysis is performed. Wilcoxon Signed Rank test calculates the sum of ranks where the first algorithm outperforms the second, R^+ and the sum of ranks where the second algorithm outperforms the second, R^- . Table 4.19

shows that the sum of ranks where AMBPSO outperforms SKFMM, R^+ and sum of ranks where SKFMM outperforms AMBPSO, R^- .

Table 4.18 Sum of Ranks for AMBPSO vs SKFMM

AMBPSO vs SKFMM	Sum of Ranks
AMBPSO Outperform SKFMM, R^+	5
SKFMM Outperform AMBPSO, R^-	148

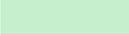
After the sum of ranks are obtained, the test statistic, T is chosen. The test statistic, T value is the smallest value between R^+ and R^- as shown in equation 4.1. The null hypothesis states that the algorithms compared are equals with no significant difference. In Wilcoxon signed ranked test, the null hypothesis is rejected when the test statistic, T is less than the critical value, T_0 . By referring appendix A, at $\alpha = 0.05$ and number of test inputs, $n = 17$, the critical value, $T_0 = 35$. Since, the test statistic, $T = 5$ and is less than the critical value, T_0 , therefore, the null hypothesis can be rejected and there is significant difference between SKFMM and AMBPSO.

4.4.4 SKF vs OBSKF

Table 4.19 shows the best, worst, mean and standard deviation (STD) of SINR(dB) values after the experiment is repeated 100 times. Comparison is made between SKF and OBSKF for various SNR(dB) input values.

Table 4.19 Best, Worst, Mean and Standard Deviation(STD) of SINR for SKF vs OBSKF

SNR (dB)	SKF				OBSKF			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
-20	-10.0522	-10.0549	-10.0523	0.0004	-10.0522	-10.0522	-10.0522	1.39E-07
-15	-5.1395	-5.1464	-5.1398	0.0011	-5.1395	-5.1397	-5.1395	2.29E-05
-10	-0.2975	-0.3028	-0.2977	0.0007	-0.2975	-0.2975	-0.2975	2.20E-07
-5	4.5321	4.3828	4.5295	0.0175	4.5321	4.5316	4.5321	4.33E-05
0	9.4241	9.3237	9.4208	0.0142	9.4241	9.4177	9.4240	0.0007
5	14.3768	14.1437	14.3631	0.0306	14.3768	14.2131	14.3677	0.0245
10	19.3590	18.8208	19.2736	0.0954	19.3564	18.8957	19.2854	0.0823
15	24.3535	23.0034	24.1049	0.2935	24.3538	23.5108	24.1784	0.1529
20	29.3469	28.0453	29.0012	0.3236	29.3438	27.1338	28.9768	0.4337
25	34.3504	32.7718	33.9332	0.3790	34.3491	31.5473	33.9442	0.3964
30	39.3477	37.5436	38.9249	0.3946	39.3467	37.3418	38.9474	0.3955
35	44.3510	41.8109	43.9153	0.4384	44.3499	41.7999	43.8989	0.4392
40	49.3456	46.3051	48.9181	0.4529	49.3341	47.0214	48.9543	0.3544
45	54.3504	51.9201	53.8808	0.4746	54.3425	52.6826	53.9350	0.3521
50	59.3355	55.8211	58.8351	0.6189	59.3478	56.6860	58.8876	0.4814
55	64.3468	62.0234	63.8755	0.5364	64.3424	61.9313	63.9258	0.4389
60	69.3396	67.7525	68.9202	0.4239	69.3464	66.8448	68.8720	0.5243

Neutral 
 Good 
 Bad 

Based on Table 4.19, the best SINR values for both SKF and OBSKF is the same for $SNR(dB) = [-20, -15, -10, -5, 0, 5]$. SKF has the best SINR values for input $SNR(dB) = [10, 20, 25, 30, 35, 40, 45, 55]$ and OBSKF produce best $SINR(dB)$ values at $SNR(dB) = [15, 50, 60]$. On the other hand, OBSKF can produce better worst, mean and standard deviation (STD) of SINR values than SKF for $SNR(dB) = [-20, -15, -10, -5, 0, 5, 10]$. This proves that OBSKF is most effective in getting maximum SINR values at lower SNR inputs. Overall, OBSKF can mostly produce much higher mean SINR value than SKF and is much more consistent than SKF due to low standard deviation values. The exploration of opposite agents provided by the OBL technique helps improve the exploration of OBSKF, leading towards higher mean SINR and higher stability of SINR values.

Based on the mean results for both SKF and OBSKF from Table 4.19, Wilcoxon Signed Ranked test statistical analysis is performed. Wilcoxon Signed Rank test calculates the sum of ranks where the first algorithm outperforms the second, R^+ and the sum of ranks where the second algorithm outperforms the second, R^- . Table 4.20 shows

the sum of ranks where SKF outperforms OBSKF, R^+ and sum of ranks where OBSKF outperforms SKF, R^- .

Table 4.20 Sum of Ranks for SKF vs OBSKF

SKF vs OBSKF	Sum of Ranks
SKF Outperform OBSKF, R^+	33
OBSKF Outperform SKF, R^-	120

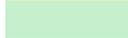
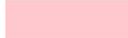
After the sum of ranks are obtained, the test statistic, T is chosen. The test statistic, T value is the smallest value between R^+ and R^- as shown in equation 4.1. The null hypothesis states that the algorithms compared are equals with no significant difference. In Wilcoxon signed ranked test, the null hypothesis is rejected when the test statistic, T is less than the critical value, T_0 . By referring appendix A, at $\alpha = 0.05$ and number of test inputs, $n = 17$, the critical value, $T_0 = 35$. Since, the test statistic, $T = 33$ and is less than the critical value, T_0 , therefore, the null hypothesis can be rejected and there is significant difference between OBSKF and SKF.

4.4.5 SKF vs SKFMM

Table 4.21 shows the best, worst, mean and standard deviation (STD) of SINR(dB) values after the experiment is repeated 100 times. Comparison is made between SKF and SKFMM for various SNR(dB) input values.

Table 4.21 Best, Worst, Mean and Standard Deviation(STD) of SINR for SKF vs SKFMM

SNR (dB)	SKF				SKFMM			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
-20	-10.0522	-10.0549	-10.0523	0.0004	-10.0522	-10.0522	-10.0522	1.50E-14
-15	-5.1395	-5.1464	-5.1398	0.0011	-5.1395	-5.1395	-5.1395	6.15E-15
-10	-0.2975	-0.3028	-0.2977	0.0007	-0.2975	-0.2975	-0.2975	5.04E-16
-5	4.5321	4.3828	4.5295	0.0175	4.5321	4.5321	4.5321	7.90E-14
0	9.4241	9.3237	9.4208	0.0142	9.4241	9.4241	9.4241	2.12E-06
5	14.3768	14.1437	14.3631	0.0306	14.3768	14.3601	14.3724	0.0039
10	19.3590	18.8208	19.2736	0.0954	19.3592	19.0344	19.2735	0.0781
15	24.3535	23.0034	24.1049	0.2935	24.3515	23.5126	24.1363	0.1903
20	29.3469	28.0453	29.0012	0.3236	29.3507	27.3467	29.0632	0.3056
25	34.3504	32.7718	33.9332	0.3790	34.3434	32.8106	33.9984	0.3204
30	39.3477	37.5436	38.9249	0.3946	39.3473	37.5921	39.0208	0.3171
35	44.3510	41.8109	43.9153	0.4384	44.3484	42.2910	43.9895	0.4356
40	49.3456	46.3051	48.9181	0.4529	49.3502	48.0500	49.0078	0.3099
45	54.3504	51.9201	53.8808	0.4746	54.3408	52.7866	54.0659	0.3231
50	59.3355	55.8211	58.8351	0.6189	59.3485	56.9759	59.0361	0.3208
55	64.3468	62.0234	63.8755	0.5364	64.3512	62.4793	63.9707	0.3903
60	69.3396	67.7525	68.9202	0.4239	69.3466	66.8526	69.0277	0.3680

Neutral 
 Good 
 Bad 

Based on Table 4.21, the best SINR values for both SKF and SKFMM is the same for $SNR(dB) = [-20, -15, -10, -5, 0, 5]$. SKF has the best SINR values for input $SNR(dB) = [15, 25, 30, 35, 45]$ and SKFMM produce best $SINR(dB)$ values at $SNR(dB) = [10, 20, 40, 50, 55, 60]$. Overall, the difference between the best SINR and the worst SINR for SKFMM is smaller compared to SKF. The lower difference between the best SINR and worst SINR means SKFMM can maintain a higher SINR values compared to SKF. SKFMM also produced higher mean SINR values and much lower standard deviation values compared to SKF. The high mean SINR and low standard deviation values for SKFMM proves that SKFMM is very consistent in maintaining higher mean SINR values. With the modified measurement in SKFMM algorithm, the SKFMM algorithm can prevent premature convergence and achieve much higher mean SINR values.

Based on the mean results for both SKF and SKFMM from Table 4.21, Wilcoxon Signed Ranked test statistical analysis is performed. Wilcoxon Signed Rank test

calculates the sum of ranks where the first algorithm outperforms the second, R^+ and the sum of ranks where the second algorithm outperforms the first, R^- . Table 4.20 shows the sum of ranks where SKF outperforms SKFMM, R^+ and sum of ranks where SKFMM outperforms SKF, R^- .

Table 4.22 Sum of Ranks for SKF vs SKFMM

SKF vs SKFMM	Sum of Ranks
SKF Outperform SKFMM, R^+	2
SKFMM Outperform SKF, R^-	151

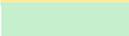
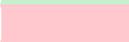
After the sum of ranks are obtained, the test statistic, T is chosen. The test statistic, T value is the smallest value between R^+ and R^- as shown in equation 4.1. The null hypothesis states that the algorithms compared are equals with no significant difference. In Wilcoxon signed ranked test, the null hypothesis is rejected when the test statistic, T is less than the critical value, T_0 . By referring appendix A, at $\alpha = 0.05$ and number of test inputs, $n = 17$, the critical value, $T_0 = 35$. Since, the test statistic, $T = 2$ and is less than the critical value, T_0 , therefore, the null hypothesis can be rejected and there is significant difference between SKF and SKFMM.

4.4.6 OBSKF vs SKFMM

Table 4.23 shows the best, worst, mean and standard deviation (STD) of SINR(dB) values after the experiment is repeated 100 times. Comparison is made between OBSKF and SKFMM for various SNR(dB) input values.

Table 4.23 Best, Worst, Mean and Standard Deviation(STD) of SINR for OBSKF vs SKFMM

SNR (dB)	OBSKF				SKFMM			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
-20	-10.0522	-10.0522	-10.0522	1.39E-07	-10.0522	-10.0522	-10.0522	1.50E-14
-15	-5.1395	-5.1397	-5.1395	2.29E-05	-5.1395	-5.1395	-5.1395	6.15E-15
-10	-0.2975	-0.2975	-0.2975	2.20E-07	-0.2975	-0.2975	-0.2975	5.04E-16
-5	4.5321	4.5316	4.5321	4.33E-05	4.5321	4.5321	4.5321	7.90E-14
0	9.4241	9.4177	9.4240	0.0007	9.4241	9.4241	9.4241	2.12E-06
5	14.3768	14.2131	14.3677	0.0245	14.3768	14.3601	14.3724	0.0039
10	19.3564	18.8957	19.2854	0.0823	19.3592	19.0344	19.2735	0.0781
15	24.3538	23.5108	24.1784	0.1529	24.3515	23.5126	24.1363	0.1903
20	29.3438	27.1338	28.9768	0.4337	29.3507	27.3467	29.0632	0.3056
25	34.3491	31.5473	33.9442	0.3964	34.3434	32.8106	33.9984	0.3204
30	39.3467	37.3418	38.9474	0.3955	39.3473	37.5921	39.0208	0.3171
35	44.3499	41.7999	43.8989	0.4392	44.3484	42.2910	43.9895	0.4356
40	49.3341	47.0214	48.9543	0.3544	49.3502	48.0500	49.0078	0.3099
45	54.3425	52.6826	53.9350	0.3521	54.3408	52.7866	54.0659	0.3231
50	59.3478	56.6860	58.8876	0.4814	59.3485	56.9759	59.0361	0.3208
55	64.3424	61.9313	63.9258	0.4389	64.3512	62.4793	63.9707	0.3903
60	69.3464	66.8448	68.8720	0.5243	69.3466	66.8526	69.0277	0.3680

Neutral 
 Good 
 Bad 

Based on Table 4.23, the best SINR values for both OBSKF and SKFMM is the same for $SNR(dB) = [-20, -15, -10, -5, 0, 5]$. OBSKF has the best SINR values for input $SNR(dB) = [15, 25, 35, 45]$ and SKFMM produce best $SINR(dB)$ values at $SNR(dB) = [10, 20, 30, 40, 50, 55, 60]$. Overall, OBSKF produced much lower values for worst SINR compared to SKFMM. The higher values for worst SINR leads SKFMM to produce a much higher mean SINR values compared to OBSKF. SKFMM is also more consistent than OBSKF due to the lower standard deviation values. Both OBSKF and SKFMM are the modified versions of SKF, designed to prevent premature convergence. Since SKFMM can produce much higher mean SINR values than OBSKF, SKFMM algorithm is proven to be better algorithm than OBSKF in preventing premature convergence.

Based on the mean results for both OBSKF and SKFMM from Table 4.23, Wilcoxon Signed Ranked test statistical analysis is performed. Wilcoxon Signed Rank test calculates the sum of ranks where the first algorithm outperforms the second, R^+ and

the sum of ranks where the second algorithm outperforms the first, R^- . Table 4.24 shows the sum of ranks where OBSKF outperforms SKFMM, R^+ and sum of ranks where SKFMM outperforms OBSKF, R^- .

Table 4.24 Sum of Ranks for OBSKF vs SKFMM

OBSKF vs SKFMM	Sum of Ranks
OBSKF Outperform SKFMM, R^+	15
SKFMM Outperform OBSKF, R^-	138

After the sum of ranks are obtained, the test statistic, T is chosen. The test statistic, T value is the smallest value between R^+ and R^- as shown in equation 4.1. The null hypothesis states that the algorithms compared are equal with no significant difference. In Wilcoxon Signed Ranked Test, the null hypothesis is rejected when the test statistic, T is less than the critical value, T_0 . By referring appendix A, at $\alpha = 0.05$ and number of test inputs, $n = 17$, the critical value, $T_0 = 35$. Since, the test statistic, $T = 15$ and is less than the critical value, T_0 , the null hypothesis can be rejected and there is significant difference between OBSKF and SKFMM.

4.5 Discussion

Figure 4.7 shows the graph of the standard deviation values for AMBPSO, SKF, OBSKF and SKFMM for various SNR after 100 runs. From Figure 4.7, it shows that as the SNR increases, the values of standard deviation also increase, or the algorithms become less consistent in finding the best SINR. As the SNR increases, the noise power decreases. The noise power is relevant in finding the best SINR. When the noise power decreases, the algorithm finds it difficult to find suitable weights to null the interference signal, therefore, decreasing the algorithms consistency as the SNR increases. Figure 4.7 also it is shown that all the SKF algorithms are much more consistent compared to AMBPSO for adaptive beamforming application. The AMBPSO algorithm begins to lose its consistency beginning from $SNR = -5dB$.

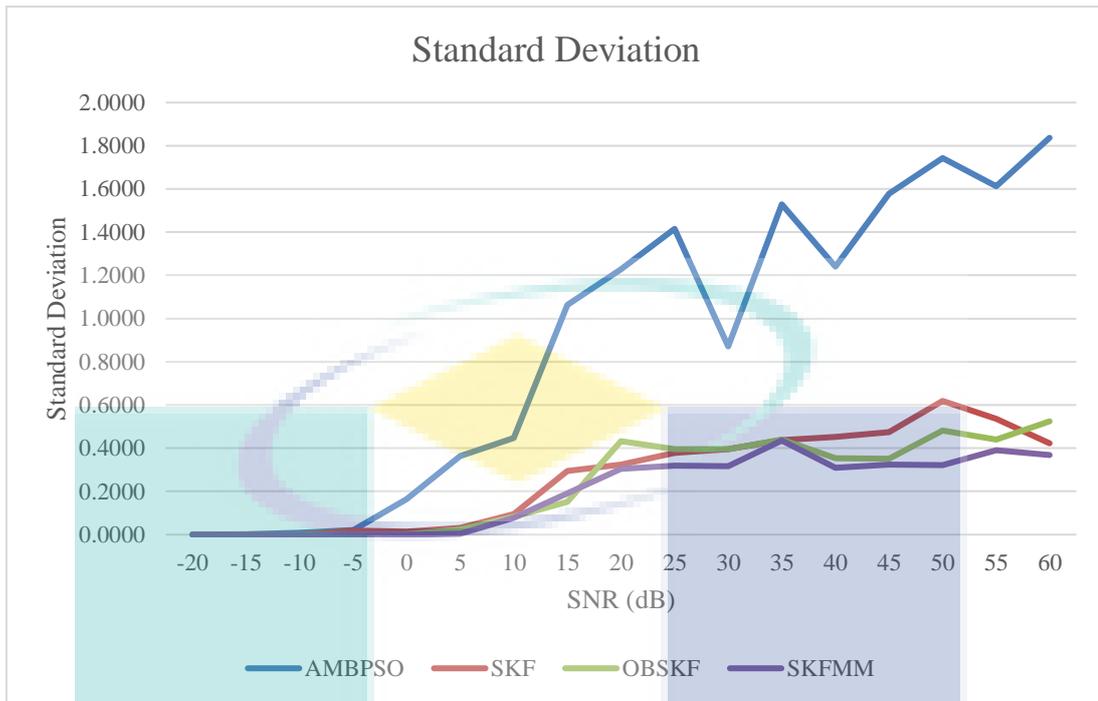


Figure 4.7 Graph of Standard Deviation of AMBPSO, SKF, OBSKF and SKFMM

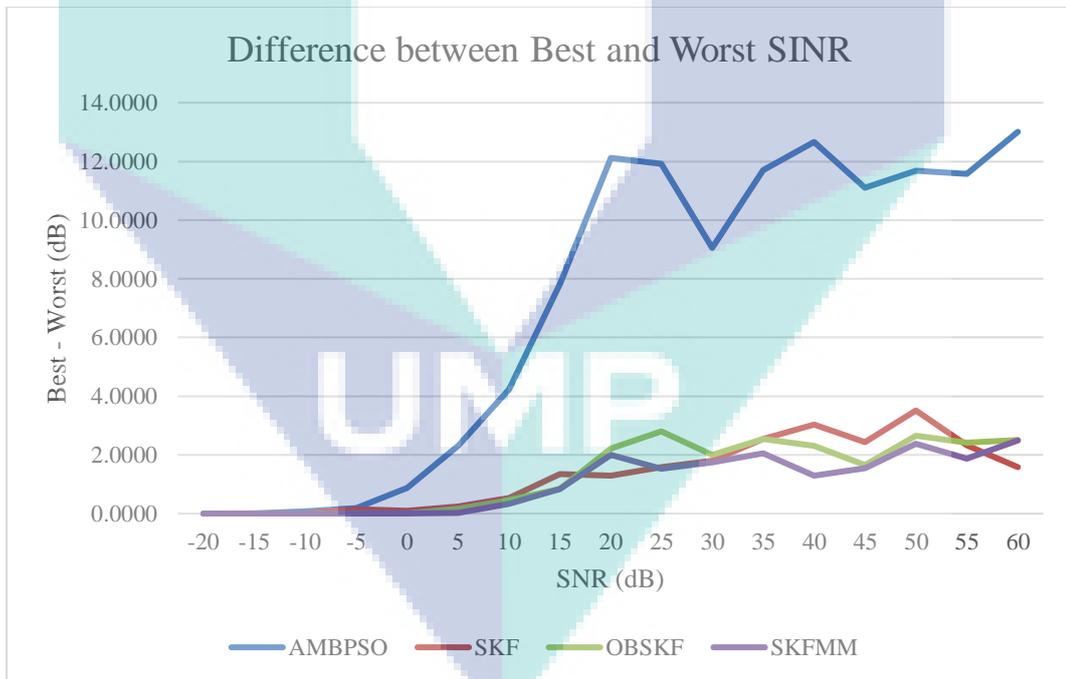


Figure 4.8 Difference Between Best and Worst SINR

Originally, the AMBPSO was designed to increase the exploration ability of the particles using Boolean method for PSO’s velocity update and control the speed of convergence using “negative selection” (NS), which is a basic mechanism of Artificial Immune System (AIS) (Zaharis & Yioultsis, 2011). To increase the exploitation ability

of the particles, after the completion of the NS, an adaptive mutation process is applied with “mutation probability” (m). The AMBPSO avoids pure random search by starting with low values of “mutation probability” (m) (Zaharis & Yioultsis, 2011). In every iteration, the m value goes down linearly until it reaches zero, for exploitation. When the mutation probability, m value decreases, the exploration capabilities of AMBPSO also decreases. With the decrease in exploration in the end of the iterative process, the algorithm may find the optimum weights and may get stuck in local optimum (Zaharis et al., 2012). As a result, the AMBPSO algorithm can't maintain its consistency after the SNR reaches $-5dB$ and more, therefore, AMPSO can't find better array weights that gives deeper nulls.

In the SKF algorithm, the algorithm is shown to have the best and the most consistent performance in CEC2014 unimodal functions (Ibrahim et al., 2015). Each agent in the SKF algorithm acts as individual Kalman Filter. The Kalman Filter is an optimal recursive data processing algorithm that processes all measurement regardless of the precision, to estimate the current value (Kalman, 1960). The Kalman Filter can find the optimal solution regardless of the precision of the measurement. In SKF algorithm, no measurement is taken, therefore, the measured position is stochastically determined using $\sin(rand \times 2\pi)$. The measured position acts as a feedback to the estimation process. After that, with the help of the Kalman gain and the measured position, a new estimate can be determined. At the beginning of the iterative process the Kalman gain is higher, therefore, more of the measured position is used to influence the estimation (exploration). As the iteration progresses, the value of the Kalman gain becomes smaller, therefore, less of the measure position is used to influence the estimated value (exploitation). However, the values of the process noise, Q and the measurement noise, R , remains fixed in SKF, from the beginning of the iterative process, therefore, limiting the Kalman gain's ability to influence the exploration and exploitation. From Figure 4.9 and Figure 4.10, it shows that the error covariance estimate, P , values and the Kalman gain, K , values only influence the SKF algorithm for the first 10 iterations, and does not have any influence during exploitation, at the end of the iterative process.

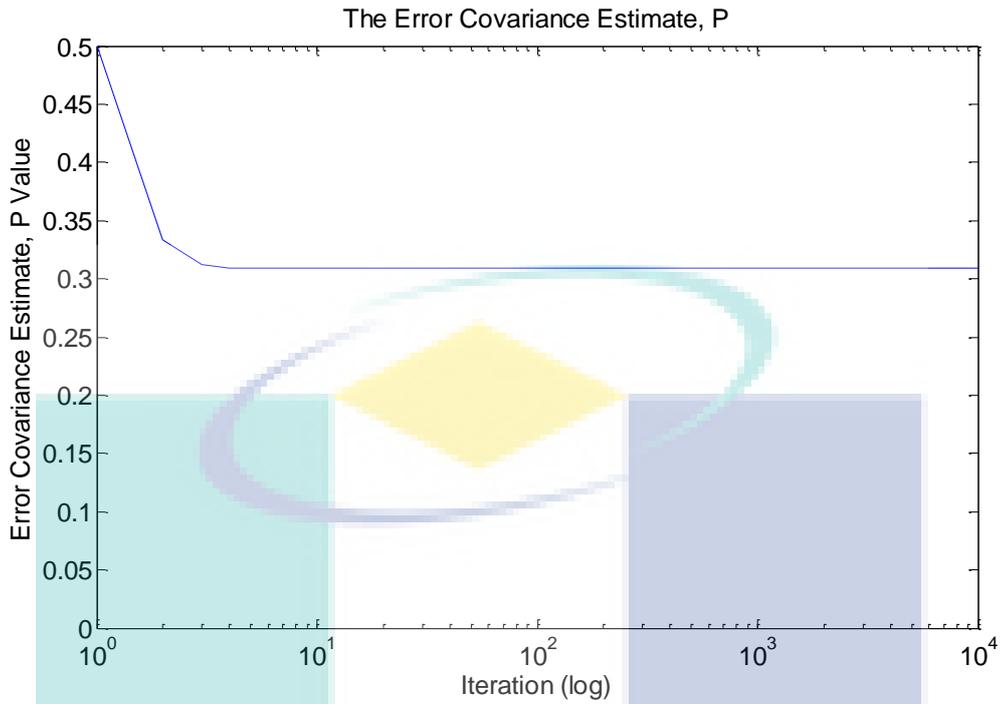


Figure 4.9 Graph of Error Covariance Estimate, P , Values vs. Iteration Number

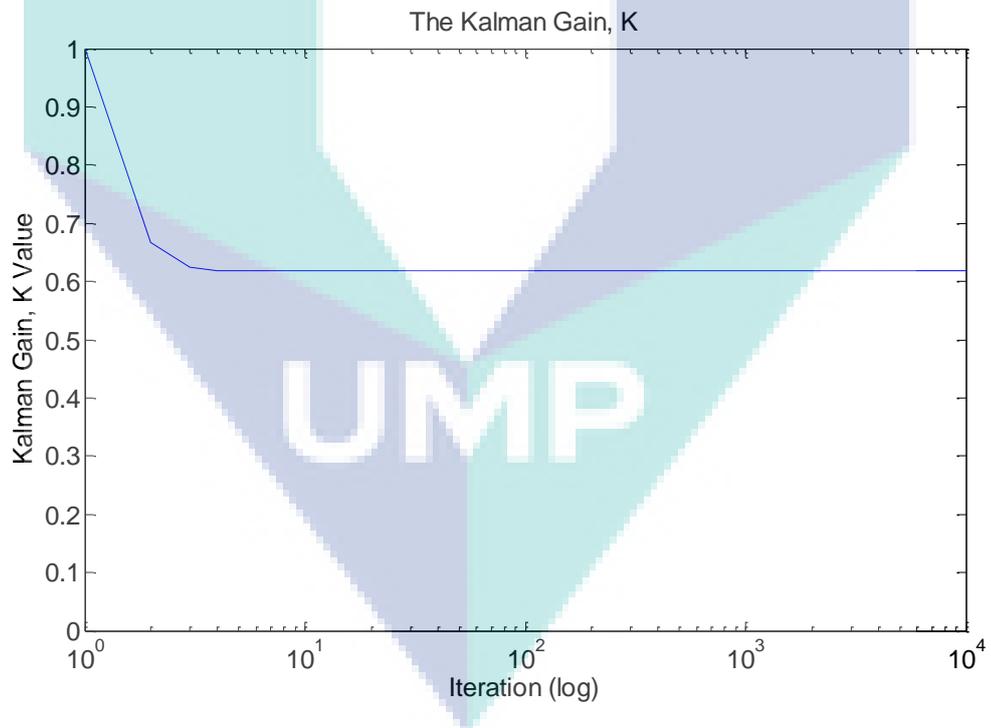


Figure 4.10 Graph of Kalman Gain, K , Values vs. Iteration Number

In SKF algorithm, the measurement-update mostly influence the exploration and exploitation in SKF algorithm. In the beginning of the iterative process, during exploration, the difference between the best-so-far solution and the predicted solution is

larger, which gives a larger measurement value. Towards the end of the iterative process the difference between the best-so-far solution and the predicted solution becomes smaller, promoting exploration. The measurement-update process in the SKF algorithm gives the most influence in finding the best array weights. The measurement-update in SKF algorithm, however, lacks in exploration towards the end of the iterative process, where the reducing difference between the best-so-far solution and the predicted solution reduces the exploration capabilities of the SKF algorithm until to almost no exploration at all. The lack of exploration towards the end of the iterative process can cause the algorithm to get stuck at local optimum. Sometimes, it is important to retain some amount of exploration towards the end of the iterative process, for algorithm to find much better solution (Zaharis et al., 2012).

The OBSKF, on the other hand, is designed to further improve the exploration capabilities of the SKF algorithm, in addition with the exploration capabilities provided by the measurement update. The OBSKF algorithm applies Opposition-Based Learning (OBL) technique to generate opposite solution, perform fitness evaluation on opposite solution and compare with the fitness obtained using current solution. This method increases the chances the algorithm to find better weights that give maximum SINR by exploring the search space more. The increased exploration provided by the OBL technique has increase the OBSKF ability to find better mean array weights than the original SKF algorithm as shown in Table 4.19. Furthermore, OBSKF is also proven to be more consistent, producing lower standard deviation values than the original SKF algorithm. The execution of the OBL technique in OBSKF is controlled by the jumping rate condition. In OBSKF, if the randomly distributed values between $[0,1]$ is below the jumping rate condition, the OBL technique will be executed. The jumping rate condition is set to a low value of 0.1 for best performance. When the jumping rate condition is set to 0.1, only 1000 iterations out of the 10000 iterations, will use the OBL technique depending on the random values and the rest of the iterations uses the original SKF algorithm. Therefore, the measurement-update is the one that contributes more to the exploration and exploitation than the OBL technique in the OBSKF algorithm.

Since the measurement-update contributes mostly to the exploration and exploitation of the OBSKF algorithm, a modification of the measurement-update was proposed. The main aim of the modification to the measurement-update is to increase the

measurement error. Increase in measurement error promotes extra exploration to find the optimum solution and prevents the SKF algorithm from converging prematurely at local optimum. Since the SKFMM has increased exploration, the SKFMM algorithm can find better array weights that gives maximum mean SINR. Furthermore, the SKFMM algorithm is the most consistent algorithm in finding the maximum SINR compared to AMBPSO, SKF and OBSKF. Table 4.25 shows the sum of standard deviation for algorithms AMBPSO, SKF, OBSKF and SKFMM. All the SKF algorithms is shown to have better consistency than the existing AMBPSO algorithm. The SKFMM has the best consistency overall with the lowest standard deviation values. Table 4.26 shows the improvement in consistency of algorithms in adaptive beamforming. From Table 4.26, all the SKF algorithms shows an improvement of more than 70 % in terms of consistency, compared to existing AMBPSO algorithm. Among the SKF algorithms, the SKFMM has the best consistency, with 22.50 % improvement in consistency to the SKF algorithm and with 17.50 % improvement in consistency to the OBSKF algorithm, for adaptive beamforming application.

Table 4.25 Sum of Standard Deviation for AMBPSO, SKF, OBSKF and SKFMM

Algorithms	Sum of Standard Deviation Values
AMBPSO	15.1303
SKF	4.4958
OBSKF	4.0764
SKFMM	3.3631

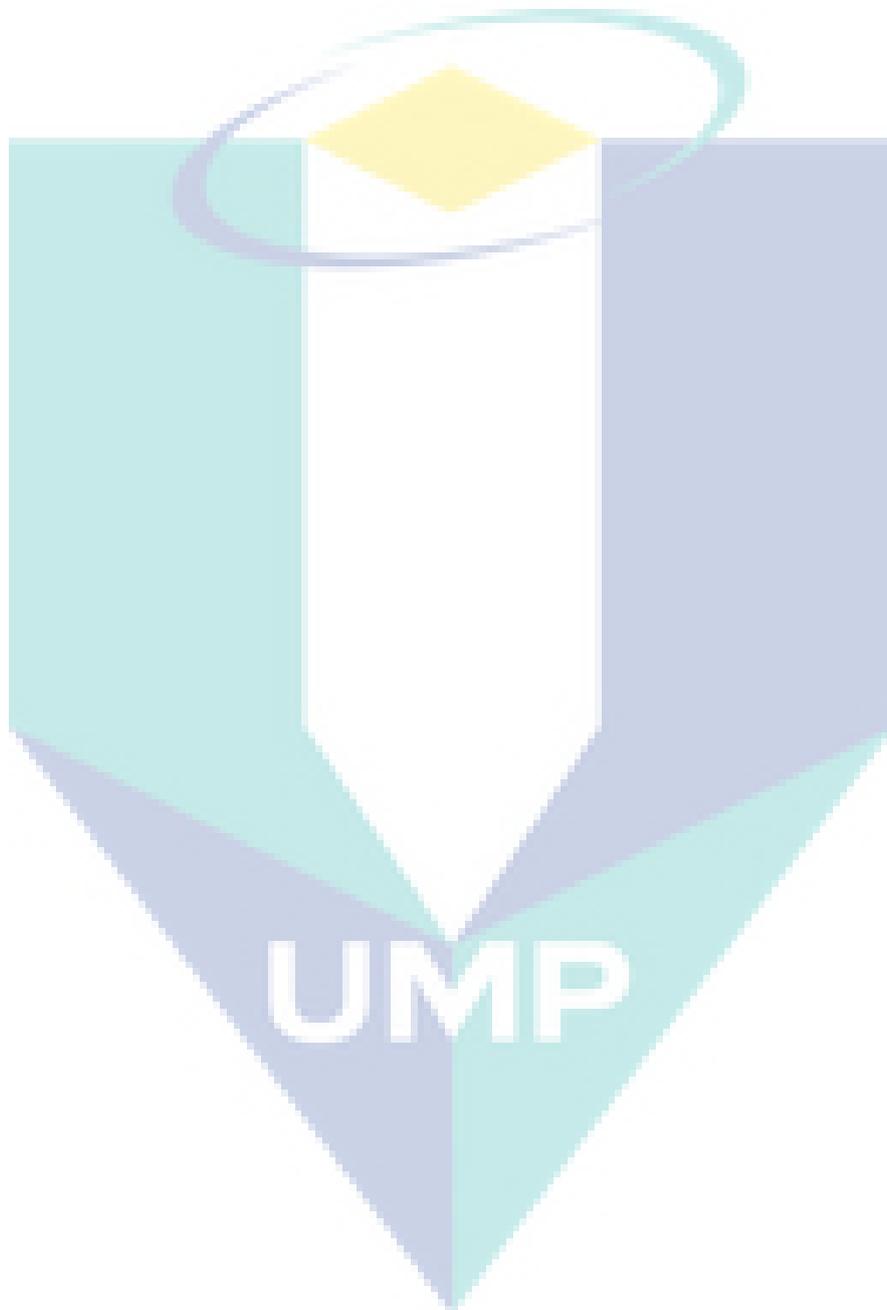
Table 4.26 Percentage of Improvement in Consistency

Algorithms	SKF	OBSKF	SKFMM
vs. AMBPSO	70.29 %	73.06 %	77.77 %
vs. SKF	-	9.33 %	25.20 %
vs. OBSKF	-	-	17.50 %

4.6 Summary

The performance of SKF algorithm and its variations, OBSKF and SKFMM, is evaluated and compared with the previously published work AMBPSO (Zaharis & Yioultsis, 2011). The radiation pattern produced by SKF algorithms produces much deeper nulls compared to AMBPSO. However, the AMBPSO still produces lowest maximum sidelobe level compared to all the SKF algorithms. The results show that the

SKF algorithms can produce high mean SINR values and lower standard deviation values. Of all the SKF algorithms, SKFMM is proven to be much better than OBSKF and SKF. The modified measurement in SKFMM is proven better in minimizing premature convergence.



CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Simulated Kalman Filter (SKF) has been proposed for the first time to improve the performance of the adaptive array antenna by maximizing the signal to interference plus noise ratio (SINR). Other than the original SKF algorithm, the Opposition-Based Simulated Kalman Filter (OBSKF) algorithm has also been proposed for adaptive beamforming. Furthermore, a new SKF algorithm with modified measurement (SKFMM) is introduced and applied for adaptive beamforming. All the SKF algorithms is compared with the existing algorithm, Adaptive Mutated Boolean Particle Swarm Optimization (AMBPSO). The SKF algorithms can produce deeper nulls and higher mean SINR values compared to AMBPSO. All three SKF algorithms are also found to be the most consistent algorithm for adaptive beamforming, producing an improvement of more than 70%, compared to AMBPSO. Among all the three SKF algorithms, the SKFMM produces the highest mean SINR values is the most consistent compared to SKF and OBSKF. SKFMM provides improved consistency of 25.20% compared to SKF and 17.50% compared to OBSKF

5.2 Recommendations

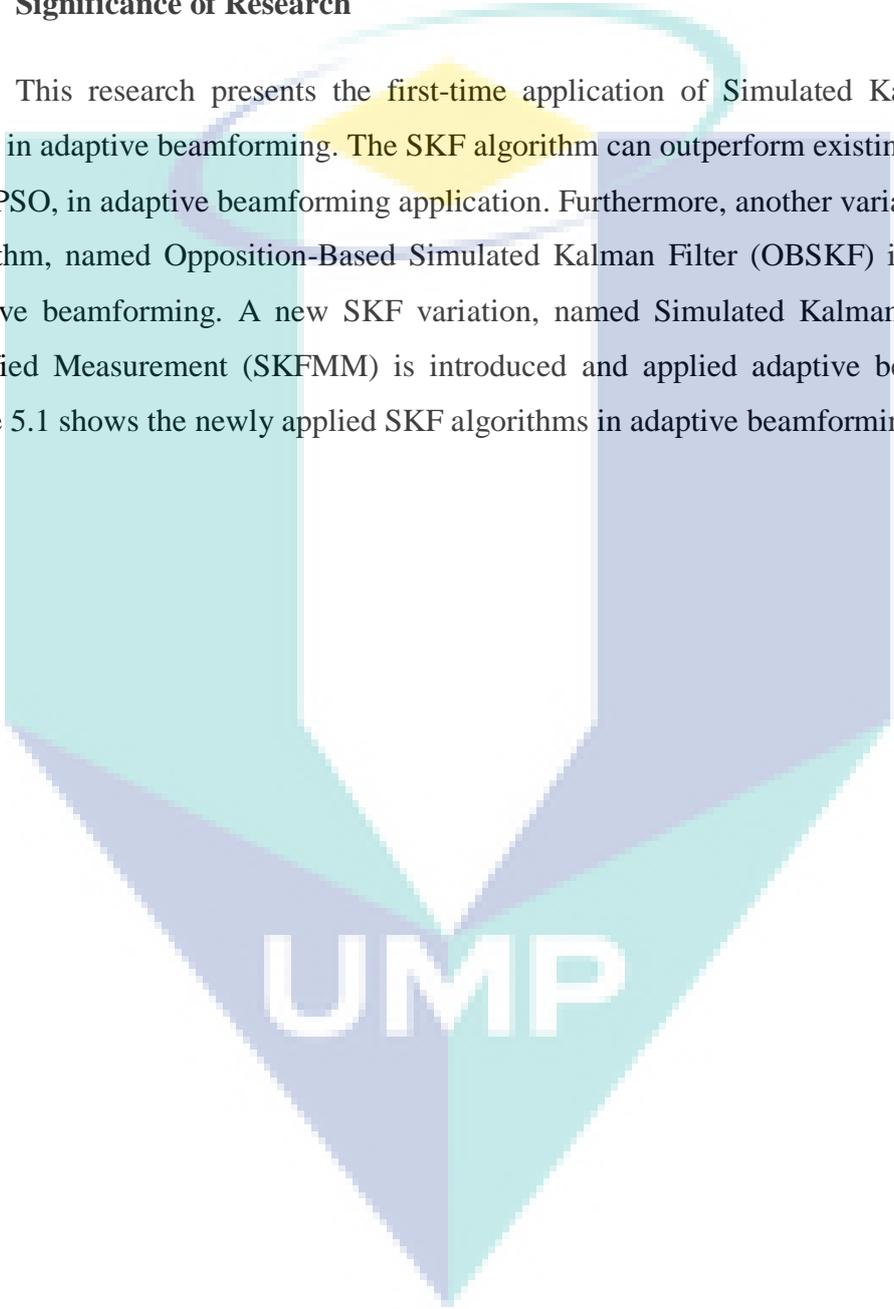
Other than SKF, other algorithms with good exploration capabilities, that has not been applied to adaptive beamforming can be used to apply to improve the performance of adaptive array antenna.

Furthermore, variations of the optimization algorithms can also be used to improve the adaptive beamforming performance.

New algorithms can be developed, specifically to improve the performance of adaptive array antenna. Since optimization algorithms only work in static environment, algorithms can be introduced to solve adaptive array antenna problems in dynamic environment.

5.3 Significance of Research

This research presents the first-time application of Simulated Kalman Filter (SKF) in adaptive beamforming. The SKF algorithm can outperform existing algorithm, AMBPSO, in adaptive beamforming application. Furthermore, another variation of SKF algorithm, named Opposition-Based Simulated Kalman Filter (OBSKF) is applied to adaptive beamforming. A new SKF variation, named Simulated Kalman Filter with Modified Measurement (SKFMM) is introduced and applied adaptive beamforming. Figure 5.1 shows the newly applied SKF algorithms in adaptive beamforming.



UMP

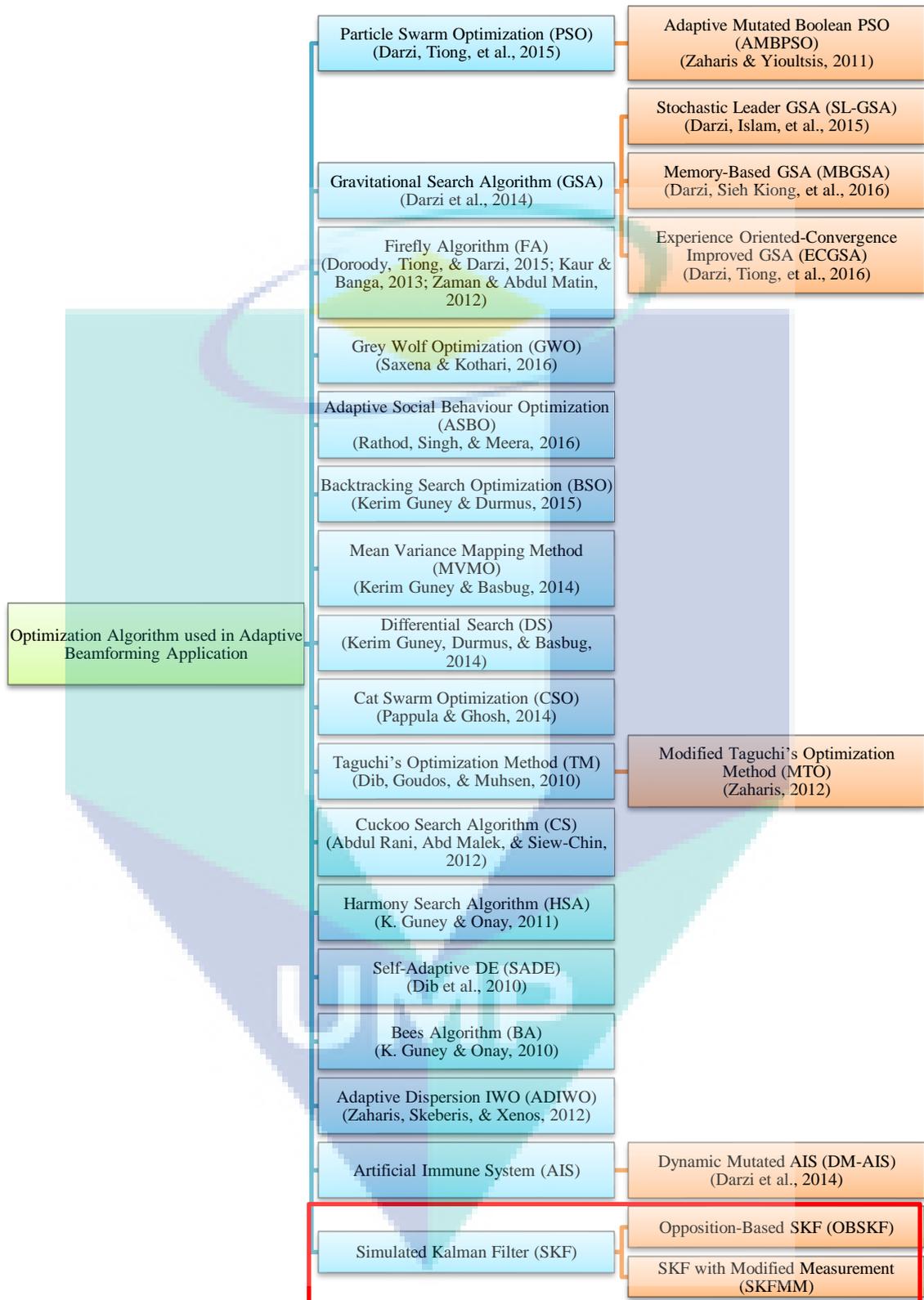


Figure 5.1 Newly applied SKF algorithms for Adaptive Beamforming

REFERENCES

- Ab. Aziz, N. A., Mubin, M., Ibrahim, Z., & Nawawi, S. W. (2015). Statistical Analysis for Swarm Intelligence — Simplified. *International Journal of Future Computer and Communication*, 4(3), 193–197. <https://doi.org/10.7763/IJFCC.2015.V4.383>
- Abdul Rani, K. N., Abd Malek, M. F., & Siew-Chin, N. (2012). Nature-inspired cuckoo search algorithm for side lobe suppression in a symmetric linear antenna array. *Radioengineering*, 21(3), 865–874.
- Allen, B. (2005). *Adaptive Array Systems*.
- Balasesm, S., Tiong, S., & Koh, S. (2011). Beamforming Algorithms Technique by Using MVDR and LCMV. *World Applied Programming*, 2(5), 315–324. Retrieved from <http://waprogramming.com/papers/50af7cd413e266.17413401.pdf>
- Chu, S.-C., Tsai, P., & Pan, J.-S. (2006). Cat Swarm Optimization. *Pacific Rim International Conference on Artificial Intelligence*, 854–858. https://doi.org/10.1007/978-3-540-36668-3_94
- Civicioglu, P. (2012). Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Computers and Geosciences*, 46, 229–247. <https://doi.org/10.1016/j.cageo.2011.12.011>
- Civicioglu, P. (2013). Backtracking Search Optimization Algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15), 8121–8144. <https://doi.org/10.1016/j.amc.2013.02.017>
- Constantine A. Balanis. (2007). *Introduction to Smart Antennas*. Morgan & Claypool. <https://doi.org/10.2200/S00079ED1V01Y200612ANT005>
- Darzi, S., Islam, M. T., Tiong, S. K., Kibria, S., & Singh, M. (2015). Stochastic leader gravitational search algorithm for enhanced adaptive beamforming technique. *PLoS ONE*, 10(11), 1–20. <https://doi.org/10.1371/journal.pone.0140526>
- Darzi, S., Sieh Kiong, T., Tariqul Islam, M., Ismail, M., Kibria, S., & Salem, B. (2014). Null steering of adaptive beamforming using linear constraint minimum variance assisted by particle swarm optimization, dynamic mutated artificial immune system, and gravitational search algorithm. *Scientific World Journal*, 2014, 1–10. <https://doi.org/10.1155/2014/724639>

- Darzi, S., Sieh Kiong, T., Tariqul Islam, M., Rezai Soleymanpour, H., & Kibria, S. (2016). A memory-based gravitational search algorithm for enhancing minimum variance distortionless response beamforming. *Applied Soft Computing Journal*, 47, 103–118. <https://doi.org/10.1016/j.asoc.2016.05.045>
- Darzi, S., Tiong, S. K., Islam, M. T., Ismail, M., & Kibria, S. (2015). Optimal null steering of minimum variance distortionless response adaptive beamforming using particle swarm optimization and gravitational search algorithm. In *ISTT 2014 - 2014 IEEE 2nd International Symposium on Telecommunication Technologies* (pp. 230–235). IEEE. <https://doi.org/10.1109/ISTT.2014.7238210>
- Darzi, S., Tiong, S. K., Tariqul Islam, M., Rezai Soleymanpour, H., & Kibria, S. (2016). An experience oriented-convergence improved gravitational search algorithm for minimum variance distortionless response beamforming optimum. *PLoS ONE*, 11(7), 1–20. <https://doi.org/10.1371/journal.pone.0156749>
- Dib, N., Goudos, S. K., & Muhsen, H. (2010). Application of Taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays. *Progress in Electromagnetics Research*, 102, 159–180. <https://doi.org/10.2528/PIER09122306>
- Doroody, C., Tiong, S. K., & Darzi, S. (2015). SINR improvement using Firefly Algorithm (FA) for Linear Constrained Minimum Variance (LCMV) beamforming technique. *I4CT 2015 - 2015 2nd International Conference on Computer, Communications, and Control Technology, Art Proceeding*, (I4ct), 441–445. <https://doi.org/10.1109/I4CT.2015.7219615>
- Erlich, I., Venayagamoorthy, G. K., & Worawat, N. (2010). A Mean-Variance Optimization algorithm. *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 18–23. <https://doi.org/10.1109/CEC.2010.5586027>
- Faragher, R. (2012). Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal Processing Magazine*, 29(5), 128–132. <https://doi.org/10.1109/MSP.2012.2203621>
- Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1–3), 187–204. [https://doi.org/10.1016/0167-2789\(86\)90240-X](https://doi.org/10.1016/0167-2789(86)90240-X)
- Frank B. Gross. (2015). *Smart Antenna with MATLAB* (2nd ed.). McGraw-Hill.

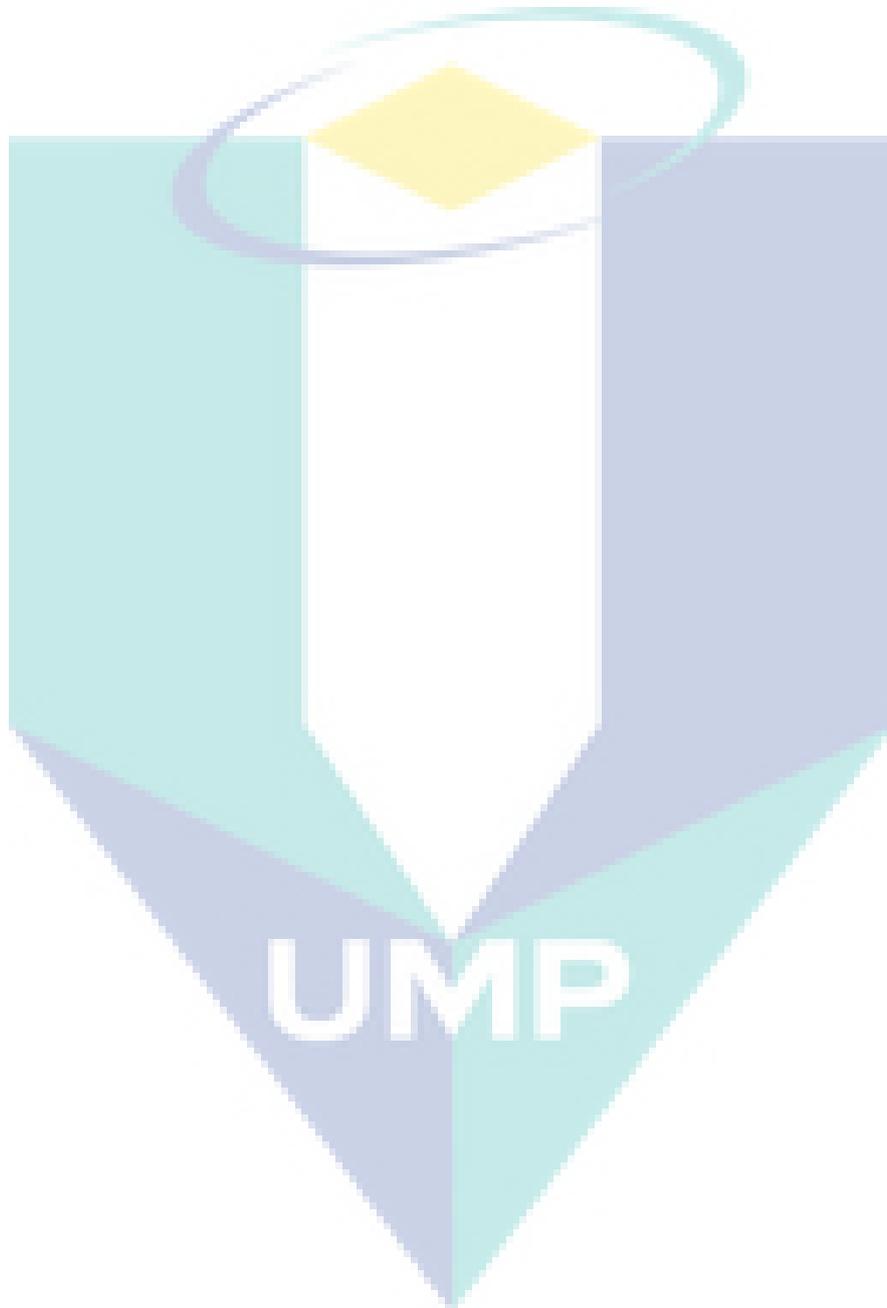
- Genichi Taguchi Yuin Wu, S. C. (2005). *Taguchi's Quality Engineering Handbook. Architecture.*
- Godara, L. C. (2004). *Smart Antennas.* CRC Press LLC.
- Gross, F. B. (2005). *Smart Antennas for Wireless Communications.* McGraw-Hill.
<https://doi.org/10.1036/007144789X>
- Guney, K., & Basbug, S. (2014). Linear Antenna Array Synthesis Using Mean Variance Mapping Method. *Electromagnetics*, 34(2), 67–84.
<https://doi.org/10.1080/02726343.2013.863665>
- Guney, K., & Durmus, A. (2015). Pattern Nulling of Linear Antenna Arrays Using Backtracking Search Optimization Algorithm. *Hindawi Publishing Corporation International Journal of Antennas and Propagation*, 2015, 10.
<https://doi.org/http://dx.doi.org/10.1155/2015/713080>
- Guney, K., Durmus, A., & Basbug, S. (2014). Antenna array synthesis and failure correction using differential search algorithm. *International Journal of Antennas and Propagation*, 2014. <https://doi.org/10.1155/2014/276754>
- Guney, K., & Onay, M. (2010). Bees algorithm for interference suppression of linear antenna arrays by controlling the phase-only and both the amplitude and phase. *Expert Systems with Applications*, 37(4), 3129–3135.
<https://doi.org/10.1016/j.eswa.2009.09.072>
- Guney, K., & Onay, M. (2011). Optimal synthesis of linear antenna arrays using a harmony search algorithm. *Expert Systems with Applications*, 38(12), 15455–15462. <https://doi.org/10.1016/j.eswa.2011.06.015>
- Ibrahim, Z., Aziz, N. H. A., Aziz, N. A. A., Razali, S., Shapiai, M. I., Nawawi, S. W., & Mohamad, M. S. (2015). A Kalman filter approach for solving unimodal optimization problems. *ICIC Express Letters*, 9(12), 3415–3422. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-84947287755&partnerID=40&md5=92e9c78e375b68f2fa1d53501583e2ed>
- K.V. Rop, D. B. O. K. (2012). Performance Analysis of a Rectangular Microstrip Pacht Antenna on Different Dielectric Substrates. *Innovative Systems Design and Engineering*, 3(8), 7–15.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems 1. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D), 35–45.

- Kaur, K., & Banga, V. K. (2013). SYNTHESIS OF LINEAR ANTENNA ARRAY USING FIREFLY ALGORITHM. *International Journal of Scientific & Engineering Research*, 4(8), 601–606.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *1995 IEEE International Conference on Neural Networks (ICNN 95)*, 4, 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
- Maybeck, P. S. (1979). Stochastic models, estimation, and control. *New York*, 1, 1–16. <https://doi.org/10.1109/TSMC.1980.4308494>
- Md Yusof, Z., Ibrahim, I., Satiman, S. N., Ibrahim, Z., Abd Aziz, N. H., & Ab Aziz, N. A. (2015). BSKF : Binary Simulated Kalman Filter. In *Third International Conference on Artificial Intelligence, Modelling and Simulation* (pp. 77–81). <https://doi.org/10.1109/AIMS.2015.23>
- Md Yusof, Z., Ibrahim, Z., Ibrahim, I., Mohd Azmi, K. Z., Ab Aziz, N. A., Abd Aziz, N. H., & Mohamad, M. S. (2016a). Angle modulated simulated kalman filter algorithm for combinatorial optimization problems. *ARPJ Journal of Engineering and Applied Sciences*, 11(7), 4854–4859.
- Md Yusof, Z., Ibrahim, Z., Ibrahim, I., Mohd Azmi, K. Z., Ab Aziz, N. A., Abd Aziz, N. H., & Mohamad, M. S. (2016b). Distance Evaluated Simulated Kalman Filter for Combinatorial Optimization Problems. *ARPJ Journal of Engineering and Applied Sciences*, 11(7), 4904–4910.
- Md Yusof, Z., Satiman, S. N., Mohd Azmi, K. Z., Muhammad, B., Razali, S., Ibrahim, Z., ... Ismail, S. (2015). Solving Airport Gate Allocation Problem using Simulated Kalman Filter. *International Conference on Knowledge Transfer*.
- Mohd Azmi, K. Z. (2017). *Opposition-Based Simulated Kalman Filter and Application in System Identification*. Universiti Malaysia Pahang.
- Muhammad, B., Ibrahim, Z., Ghazali, K. H., Mohd Azmi, K. Z., Ab Aziz, N. A., Abd Aziz, N. H., & Mohamad, M. S. (2015). A new hybrid simulated Kalman filter and gravitational search algorithm for continuous numerical optimization problems. *Unpublished (ARPJ Journal)*, 10(22), 17171–17176. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-84953407417&partnerID=40&md5=fa75fe21f0cf31942720c3a9360133ac>

- Noordin, N. H., Zuniga, V., El-Rayis, A. O., Haridas, N., Erdogan, A. T., & Arslan, T. (2011). Uniform circular arrays for phased array antenna. *LAPC 2011 - 2011 Loughborough Antennas and Propagation Conference*, (November), 1–4. <https://doi.org/10.1109/LAPC.2011.6114031>
- Pappula, L., & Ghosh, D. (2014). Linear antenna array synthesis using cat swarm optimization. *AEU - International Journal of Electronics and Communications*, 68(6), 540–549. <https://doi.org/10.1016/j.aeue.2013.12.012>
- Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The bees algorithm—A novel tool for complex optimisation. *Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems*, 454–459. <https://doi.org/http://dx.doi.org/10.1016/B978-008045157-2/50081-X>
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Rathod, M. L., Singh, M. K., & Meera, A. (2016). Phase Perturbation based Radiation Pattern Synthesis in Antenna Array using ASBO. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*.
- Roslee, M., Subari, K. S., & Shahdan, I. S. (2011). Design of bow tie antenna in CST studio suite below 2GHz for ground penetrating radar applications. *2011 IEEE International RF and Microwave Conference, RFM 2011 - Proceedings*, (December), 430–433. <https://doi.org/10.1109/RFM.2011.6168783>
- Saxena, P., & Kothari, A. (2016). Optimal Pattern Synthesis of Linear Antenna Array Using Grey Wolf Optimization Algorithm. *International Journal of Antennas and Propagation*, 2016. <https://doi.org/10.1155/2016/1205970>
- Saxena, P., & Kothari, A. G. (2014). Performance Analysis of Adaptive Beamforming Algorithms for Smart Antennas. *IERI Procedia*, 10, 131–137. <https://doi.org/10.1016/j.ieri.2014.09.101>
- Shiu, W. Y. (1998). *Noniterative digital beamforming in CDMA cellular communications systems*. <https://doi.org/10.1.1.8.3221>
- Singh, M. K. (2012). A New Optimization Method Based on Adaptive Social Behavior : ASBO, 823–831.

- Souden, M., Benesty, J., & Affes, S. (2010). A study of the LCMV and MVDR noise reduction filters. *IEEE Transactions on Signal Processing*, 58(9), 4925–4935. <https://doi.org/10.1109/TSP.2010.2051803>
- Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2013). *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*.
- Tizhoosh, H. R. (2005). Opposition-Based Learning: A New Scheme for Machine Intelligence. *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, 1, 695–701. <https://doi.org/10.1109/CIMCA.2005.1631345>
- Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M. (2011). Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences*, 181(20), 4699–4714. <https://doi.org/10.1016/j.ins.2011.03.016>
- Welch, G., & Bishop, G. (2006). An Introduction to the Kalman Filter. *In Practice*, 7(1), 1–16. <https://doi.org/10.1.1.117.6808>
- Xu, Q., Wang, L., Wang, N., Hei, X., & Zhao, L. (2014). A review of opposition-based learning from 2005 to 2012. *Engineering Applications of Artificial Intelligence*, 29, 1–12. <https://doi.org/10.1016/j.engappai.2013.12.004>
- Yang, X.-S. (2010). Firefly Algorithm, Stochastic Test Functions and Design Optimisation. <https://doi.org/10.1504/IJBIC.2010.032124>
- Zaharis, Z. D. (2012). a Modified Taguchi'S Optimization Algorithm for Beamforming Applications. *Progress In Electromagnetics Research*, 127(January), 553–569.
- Zaharis, Z. D., Skeberis, C., & Xenos, T. D. (2012). Improved Antenna Array Adaptive Beam-forming with Low Side Lobe Level using a Novel Adaptive Invasive Weed Optimization Method. *Progress In Electromagnetics Research*, 124, 137–150.
- Zaharis, Z. D., & Yioultis, T. V. (2011). A Novel Adaptive Beamforming Technique Applied on Linear Antenna Arrays Using Adaptive Mutated Boolean PSO. *Progress In Electromagnetics Research*, 117, 165–179. <https://doi.org/10.2528/P1er11041904>

Zaman, M. A., & Abdul Matin, M. (2012). Nonuniformly spaced linear antenna array design using firefly algorithm. *International Journal of Microwave Science and Technology*, 2012. <https://doi.org/10.1155/2012/256759>



APPENDIX A
CRITICAL VALUES OF T_0 IN WILCOXON SIGNED RANK TEST

α	0.10	0.05	0.01	α	0.10	0.05	0.01
$n = 5$	1			$n = 28$	130	117	92
$n = 6$	2	1		$n = 29$	141	127	100
$n = 7$	4	2		$n = 30$	152	137	109
$n = 8$	6	4	0	$n = 31$	163	148	118
$n = 9$	8	6	2	$n = 32$	175	159	128
$n = 10$	11	8	3	$n = 33$	188	171	138
$n = 11$	14	11	5	$n = 34$	201	183	149
$n = 12$	17	14	7	$n = 35$	214	195	160
$n = 13$	21	17	10	$n = 36$	228	208	171
$n = 14$	26	21	13	$n = 37$	242	222	183
$n = 15$	30	25	16	$n = 38$	256	235	195
$n = 16$	36	30	19	$n = 39$	271	250	208
$n = 17$	41	35	23	$n = 40$	287	264	221
$n = 18$	47	40	28	$n = 41$	303	279	234
$n = 19$	54	46	32	$n = 42$	319	295	248
$n = 20$	60	52	37	$n = 43$	336	311	262
$n = 21$	68	59	43	$n = 44$	353	327	277
$n = 22$	75	66	49	$n = 45$	371	344	292
$n = 23$	83	73	55	$n = 46$	389	361	307
$n = 24$	92	81	61	$n = 47$	408	379	323
$n = 25$	101	90	68	$n = 48$	427	397	339
$n = 26$	110	98	76	$n = 49$	446	415	356
$n = 27$	120	107	84	$n = 50$	466	434	373

Source: (Ab. Aziz, Mubin, Ibrahim, & Nawawi, 2015)

APPENDIX B
LIST OF PUBLICATION

1. Lazarus, K., Noordin, N. H., Ibrahim, Z., & Abas, K. H. (2016). Adaptive Beamforming Algorithm based on Simulated Kalman Filter. 2016 Asia Multi Conference on Modelling and Simulation, 19–23.
2. Lazarus, K., Noordin, N. H., Zakwan, K., Azmi, M., Hidayati, N., Aziz, A., & Ibrahim, Z. (2016). Adaptive Beamforming Algorithm based on Generalized Opposition-based Simulated Kalman Filter. The National Conference for Postgraduate Research.
3. K. Lazarus et al., “An Opposition-Based Simulated Kalman Filter Algorithm for Adaptive Beamforming,” in 2017 IEEE International Conference on Applied System Inovation, 2017.

UMP

