International Journal of
Intelligent Systems and Applications in Engineering
IJISAE
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Comparative Study of Krill Herd, Firefly and Cuckoo Search Algorithms for Unimodal and Multimodal Optimization

Gobind Preet Singh*[1], Abhay Singh[2]

Received 14th October 2013, Accepted 23th December 2013

Abstract: Today, in computer science, a computational challenge exists in finding a globally optimized solution from an enormously large search space. Various metaheuristic methods can be used for finding the solution in a large search space. These methods can be explained as iterative search processes that efficiently perform the exploration and exploitation in the solution space. In this context, three such nature inspired metaheuristic algorithms namely Krill Herd Algorithm (KH), Firefly Algorithm (FA) and Cuckoo search Algorithm (CS) can be used to find optimal solutions of various mathematical optimization problems. In this paper, the proposed algorithms were used to find the optimal solution of fifteen unimodal and multimodal benchmark test functions commonly used in the field of optimization and then compare their performances on the basis of efficiency, convergence, time and conclude that for both unimodal and multimodal optimization Cuckoo Search Algorithm via Lévy flight has outperformed others and for multimodal optimization Krill Herd algorithm is superior than Firefly algorithm but for unimodal optimization Firefly is superior than Krill Herd algorithm.

Keywords: *Metaheuristic Algorithm, Krill Herd Algorithm, Firefly Algorithm, Cuckoo Search Algorithm, Unimodal Optimization, Multimodal Optimization.*

1. Introduction

In recent times, nature inspired metaheuristic algorithms are being widely used for solving optimization problems, including NP-hard problems which might need exponential computation time to solve in worst case scenario. In metaheuristics methods [1, 9] we might compromise on finding an optimal solutions just for the sake of getting good solutions in a specific period of time. The main aim of metaheuristic algorithms are to quickly find solution to a problem, this solution may not be the best of all possible solutions to the problem but still they stand valid as they do not require excessively long time to be solved. Two crucial characteristics of metaheuristic algorithms are intensification and diversification. The intensification searches around the current best solution and selects the best candidate or solution. The diversification ensures that the algorithm explores the search space more efficiently. Maintaining balance between diversification and intensification is important because firstly we have to quickly find the regions in the large search space with high quality solutions and secondly not to waste too much time in regions of the search space which are either already explored or which do not provide high quality solutions[3].

In this paper, we have used three metaheuristic algorithms Krill Herd Algorithm (KH) [4], Firefly Algorithm (FA) [1] and Cuckoo Search Algorithm (CS) [5].First is the Krill Herd Algorithm which was developed by Amir Hossein Gandomi and Amir Hossein Alavi in 2011. The KH algorithm is based on the simulation of the herding behaviour of Krill individuals. Second is the Firefly Algorithm (FA) which was developed by X.-S.Yang in 2007. It was inspired by the flashing pattern of fireflies. Third algorithm is the Cuckoo search Algorithm which was developed by X.-S.Yang and S.Deb in 2009. It is based on the interesting breeding behaviour of certain species of cuckoos such as brood parasitism.

This paper aims to provide the comparison study of the Krill Herd Algorithm (KH) with Firefly Algorithm (FA) and Cuckoo Search (CS) Algorithm via L´evy Flights against unimodal and multimodal test functions. Rest of the paper is organised as follows. First we will briefly explain the Krill Herd Algorithm, Firefly Algorithm, Cuckoo Search Algorithms and several Mathematical benchmark functions in section (2).Then experimental settings and results will be shown in section (3) and then finally we will conclude the paper.

2. Nature Inspired Algorithms and Optimization

2.1. Krill Herd Algorithm

2.1.1. Krill Swarm's Herding Behavior

Many Research have been done in order to find the mechanism that lead to the development non- random formation of groups by various marine animals [11,12].The significant mechanisms identified are feeding ability, protection from predators, enhanced reproduction and environmental condition [6].

Krills from Antarctic region are one of the best researched marine animals. One of the most significant ability of krills is that they can form large swarms [13, 14].Yet there are number of uncertainties about the mechanism that lead to distribution of krill herd [15].There are proposed conceptual models to explain observed formation of krill heard [16] and result obtained from those models states that krill swarms form the basic unit of organization for this species.

Whenever predators (Penguins, Sea Birds) attack krill swarms, they take individual krill which leads in reducing the krill density. After the attack by predators, formation of krill is a multi- objective process mainly including two Goals: (1) Increasing Krill density

[1] Guru Gobind Singh Indraprastha University– India
[2] Guru Gobind Singh Indraprastha University –India
** Corresponding Author: Email: gobind75@gmail.com*

and (2) Reaching food. Attraction of Krill to increase density and finding food are used as objective function which finally lead the krills to herd around global minima. In this mechanism, all individual krill moves towards the best possible solution while searching for highest density and food.

2.1.2. Krill Herd Algorithm

As Predator remove individual from Krill swarm, the average krill density and distance of krill swarm from the food location decreases. We assume this process as the initialization phase in the Krill Herd Algorithm [4]. Value of objective function for each individual is supposed to be combination of distance from food and highest density of krill swarm. Three essential actions [6] considered by Krills to determine the time dependent position of an individual krill are:

(i) Movement induced by other krill individuals
(ii) Foraging activity
(iii) Random Diffusion

We know that all the optimization algorithm should have searching capability in space of arbitrary dimensionality. Hence we generalize Lanrangian model of krill herding to n dimensional decision space.

$$dX_i/dt = N_i + F_i + D_i \tag{1}$$

Here N_i is the motion induced by other krill individuals, F_i is the foraging motion and D_i stands for physical diffusion for *ith* krill individuals.

2.1.3. Motion Induced by other krill individual

According to research krill individuals move due to the mutual effects by each other so as to maintain high density [6].Movement for the krill individual is defined as:

$$N_i^{new} = N^{max}\alpha_i + \omega_n N_i^{old} \tag{2}$$

In Eq. (2) N^{max} stands for maximum induced speed which is equal to 0.01 (m/s) [6] , α_i is the direction of motion induced which is estimated from target swarm density and local swarm density, ω_n is the inertia weight of the motion induced, N_i^{old} is the last motion induced.

$$\alpha_i = \alpha_i^{local} + \alpha_i^{target} \tag{3}$$

In Eq. (3) α_i^{local} is the local effect due to the neighbors, and α_i^{target} is the target direction effect due to the best krill individual. Attractive or Repulsive effect of the neighbors on an individual krill movement can be formulated as:

$$\alpha_i^{local} = \sum_{j=1}^{NN} \widehat{K}_{i,j}\widehat{X}_{i.j} \tag{4}$$

$$\widehat{X}_{i,j} = X_j - X_i/\|X_j - X_i\| + \varepsilon \tag{5}$$

$$\widehat{K_{i,j}} = K_i - K_j/K^{worst} - K^{best} \tag{6}$$

In Eq. (4) NN is the number of the neighbors. In Eq. (5) and Eq. (6) $K^{worst}$and K^{best} are, the worst and the best fitness values of the krill individuals till now, K_i represents the fitness value of the *ith* krill individual, K_j is the fitness of *jth* neighbor and X represents the related positions.

To choose the neighbor, using actual behavior of Krill individual, a sensing distance (ds) is calculated using

$$d_{s,i} = 1/5N\sum_{j=1}^{N}\|X_i - X_j\| \tag{7}$$

In Eq. (7) $d_{s,i}$ is the sensing distance for the ith krill individual and N stands for number of krill individuals. Based on Eq. (7), two krill individuals are neighbor if the distance between them is less than d_s.

The effect of the individual krill having the best fitness on the ith individual krill is calculated as follow:

$$\alpha_i^{target} = C^{best}\widehat{K}_{i,best}\widehat{X}_{i,best} \tag{8}$$

In Eq. (8) C^{best} is the effective coefficient of the krill with the best fitness and is defined as:

$$C^{best} = 2(rand + I/I_{max}) \tag{9}$$

Where rand is a random value in the range [0, 1], I is the actual iteration number, and $Imax$ is the maximum number of iterations.

2.1.4. Foraging motion

Foraging motion is developed in terms of two main effective parameters. One is the food location and the second one is the previous experience about the food location. This motion can be explained for the ith krill individual as follow:

$$F_i = V_f\beta_i + \omega_f F_i^{old} \tag{10}$$

Where

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{11}$$

In Eq. (10) v_f is the foraging speed, ω_f is the inertia weight of the foraging motion in the range [0, 1] and F_i^{old} is the last foraging motion. In Eq. (11) β_i^{food} and β_i^{best} are the food attractive effect and best fitness of the ith krill so far respectively. Measured values of the foraging speed [7] is 0.02($\underline{ms^{-1}}$).

The center of food is found at first and then food attraction is formulated. The virtual center of food concentration is estimated according to the fitness distribution of the krill individuals, which is inspired from ‘‘center of mass’’. The center of food for each iteration is formulated as:

$$X^{food} = \sum_{i-1}^{N} 1/K_i\, X_i / \sum_{i-1}^{N} 1/K_i \tag{12}$$

Hence, we can evaluate the food attraction for the ith krill individual by following equation:

$$\beta_i^{food} = C^{food}\widehat{K}_{i,food}\widehat{X}_{i,food} \tag{13}$$

In Eq. (13) C^{food} is the food coefficient. As time passes the effect of food in the krill herding decrease and food coefficient is evaluated as:

$$C^{food} = 2(1 - I/I_{max}) \tag{14}$$

The attraction towards food is defined to attract the krill swarm towards global optima. Based on this definition, the krill individuals normally herd around the global optima after some iteration. This can be considered as an efficient global optimization strategy which helps improving the global optima of the KH algorithm.

The effect of the best fitness of the ith krill individual is also handled using the following equation:

$$\beta_i^{best} = \widehat{K}_{i,ibest}\widehat{X}_{i,ibest} \tag{15}$$

In Eq. (15) $K_{ibest}$is the best previously visited position of the ith krill individual.

2.1.5. Physical diffusion

The physical diffusion of all the krills is basically a random process. We can express this motion in terms of maximum diffusion speed and a random directional vector. We can formulate it as follows:

$$D_i = D^{max}\delta \qquad (16)$$

In Eq. (16) D^{max} is the maximum diffusion speed [8] and δ is the random directional vector and it includes random values in range [-1, 1]. As krills position gets better, random motion is also reduced. Thus, another term is added to the physical diffusion formula to consider this effect. The effects of the motion induced by other krill individuals and foraging motion gradually decrease increase in iterations. The physical diffusion is a random vector hence it does not steadily reduces with increase in number of iterations due to which another term is added to Eq. (16). This term, linearly decreases the random speed with time and works on the basis of a geometrical annealing schedule:

$$D_i = D^{max}(1 - I/I_{max})\delta \qquad (17)$$

2.1.6. Motion Process of the KH Algorithm

The defined motions frequently change the position of a krill individual toward the best fitness. The foraging motion and the motion induced by other krill individuals contain two global and two local strategies. KH a powerful algorithm as all these work in parallel. The formulations of these motions for the ith krill individual show that, if fitness value of each of the above mentioned effective factor likeK_j,K^{best}, K^{food}, K_i^{best} is better i.e. less than the fitness of the ith krill, it has an attractive effect else it is a repulsive effect. We can notice from the above formulations that better fitness has more effect on the movement of ith krill individual. The position vector of a krill individual during the interval t to t + Δt is given by the following equation:

$$X_i(t + \Delta t) = X_i(t) + \Delta t\, dX_i/dt \qquad (18)$$

Δt should be carefully set according to the optimization problem because this parameter works as a scale factor of the speed vector. Δt completely depends on the search space and it can be obtained simply by the following formula:

$$\Delta t = C_t \sum_{j=1}^{NV}(UB_j - LB_j) \qquad (19)$$

In Eq. (19) NV is the total number of variables and LB_j and UB_j are lower and upper bounds of the jth variables respectively. It is empirically found that $C_t$is a constant
Number between [0, 2]. Lower the values of C_t more carefully the krill individuals will search.

2.1.7. Crossover

To improve the performance of the algorithm, genetic reproduction mechanisms are incorporated into the algorithm. One such algorithm is crossover. Crossover is a genetic operator used to vary the programming of chromosomes from one generation to the next. In this Algorithm, an adaptive vectored crossover scheme is employed.
We can control crossover by a crossover probability, Cr, and actual crossover can be performed in two ways: (1) binomial and (2) exponential. The binomial scheme performs crossover on each of the d components or variables/parameters. By generating a uniformly distributed random number between 0 and 1, the mth component ofX_i, $x_{i,m}$, is determined as:

$$x_{i,m} = \begin{cases} x_{r,m}, & rand_{i,m} < Cr \\ x_{i,m}, & else \end{cases} \qquad (20)$$

$$Cr = 0.2\hat{K}_{i,best} \qquad (21)$$

In Eq. (20) r ∈ {1, 2,. . ., i _ 1, i + 1,. . .,N}. With this new crossover probability, the crossover probability for the global best is equal to zero and it increases with decrease in fitness.

2.2. Firefly Algorithm

2.2.1. Behavior and nature of Fireflies

Fireflies are the creatures that can generate light inside of it. Light production in fireflies is due to a type of chemical reaction. The primary purpose for firefly's flash is to act as a signal system to attract other fireflies. Although they have many mechanisms, the interesting issues are what they do for any communication to find food and to protect themselves from enemy hunters including their successful reproduction. There are around two thousand firefly species, and most of them produce short and rhythmic flashes. The pattern observed for these flashes is unique specific species. The rhythm of the flashes, rate of flashing and the amount of time for which the flashes are observed together forms a pattern that attracts both the males and females to each other. Females of a species respond to individual pattern of the male of the same species.
The light intensity at a particular distance from the light source follows the inverse square law. That is as the distance increases the light intensity decreases. Furthermore, the air absorbs light which becomes weaker and weaker as there is an increase of the distance. There are two combined factors that make most fireflies visible only to a limited distance that is usually good enough for fireflies to communicate each other. The flashing light can be formulated in such a way that it is associated with the objective function to be optimized. This makes it possible to formulate new metaheuristic algorithms.

2.2.2. Firefly algorithm

The firefly (FA) algorithm [1, 9, 10] is a metaheuristic algorithm, inspired by the flashing behavior of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies.
Xin-She Yang formulated this firefly algorithm by taking three assumptions [1]

i. All fireflies are unisexual, so that one firefly will be attracted to all other fireflies;
ii. Attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted by (and thus move to) the brighter one; however, the brightness can decrease as their distance increases;
iii. If there are no fireflies brighter than a given firefly, it will move randomly.

2.2.3. Light Intensity and Attractiveness

Two core issues in firefly algorithm are (i) The variation of light intensity, (ii) The formulation of the attractiveness.
For simplicity, it is assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function of the optimization problems. In the simplest case for maximum optimization problems, the brightness I of a firefly for a particular location x could be chosen as $I(x) \propto f(x)$. Even so, the attractiveness β is relative, it should be judged by the other fireflies. Thus, it will differ with the distance r_{ij} between firefly *i* and firefly *j*. In addition, light intensity decreases with the distance from its source, and light is also absorbed by the media, so we should allow the attractiveness to vary with the varying degree of absorption.
Since a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, attractiveness β of a firefly can be defined as

$$\beta(r) = \beta_o e^{-\gamma r^m}, (m \geq 1 \qquad (22)$$

In Eq. (22), r or r_{ij} is the distance between the *ith* and *jth* fireflies. β_o Is the attractiveness at r = 0 and γ is a fixed light absorption coefficient. The distance between any two fireflies *ith* and *jth* at x_i and x_j is the Cartesian distance and can be calculated as:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \quad (23)$$

In Eq. (23), x_{ik} is the *k*th component of the *i*th firefly (x_i). The movement of ith firefly, to another more attractive (brighter) *jth* firefly, is determined by

$$x_i = x_i + \beta_o e^{-\gamma r_{i,j}^2}(x_j - x_i) + \alpha(rand - \quad 0.5) \quad (24)$$

In Eq. (3) the second term is due to the attraction while the third term is the randomization with α being the randomization parameter. Rand is a random number generator uniformly distributed in the range of [0, 1]. For most cases in the implementation, β_o = 1 and α ∈ [0, 1]. Furthermore, the randomization term can easily be extended to a normal distribution N (0, 1) or other distributions.

2.3. Cuckoo Search Algorithm via Lévy Flight

2.3.1. Cuckoo's breeding behaviour

Cuckoo is an interesting bird species, known not only for the beautiful sound they make, but also for their aggressive reproductive strategy. Cuckoos are extremely diverse group of birds with regards to breeding system. Many Cuckoo species follow the strategy of brood parasitism by using host individuals either of the same or different species to raise the young of the own. Cuckoo species such as Anis and Guira lay their eggs in communal nest though they may remove others eggs to increase the survival probability of their own eggs. Some host birds can engage in direct conflict with the intruding cuckoos. On recognition of parasitic eggs, the host may kick the parasites eggs out, or build a new nest. Female parasitic cuckoos who are specialized in mimicry lay eggs that closely which resemble the eggs of their host which reduces the probability of their eggs being abandoned.

Parasitic cuckoos often choose a nest where host bird have just laid their eggs .The cuckoo egg hatches earlier as compared to the host's, and the cuckoo chick grows faster than them;. In most cases the chick evicts the eggs laid by host species, which increases the cuckoo chick's share of food provided by its host bird. Some cuckoo chick can even replicate the call of host chicks to gain access to more feeding opportunity.

2.3.2. Lévy flight

A Lévy flight is a random walk in which the step-lengths have a probability distribution that is heavy-tailed. Research works have shown that flight behavior of many animals and insects demonstrated the typical characteristics of Lévy flights [17, 18, 19, 20]. Fruit flies or Drosophila melanogaster, explore their landscape using a series of straight flight paths punctuated by a sudden 90 degree turn, leading to a Lévy -flight-style intermittent scale free search pattern was shown in a study conducted by Reynolds and Frye. Many researches shows that Lévy flights interspersed with Brownian motion can describe the animals' hunting patterns [24, 25]. Even light can be related to Lévy flights [23]. Latterly, such behavior has been applied to optimization and optimal search, and preliminary results show its promising capability [18, 20, 21, 22].

2.3.3. Cuckoo Search Algorithm

Each egg in the nest represents solution, and Cuckoo egg represents new solution. The aim is to use the new and potentially better solutions (Cuckoos) to replace not-so-good or inferior solution in the nests. In the simplest form, each nest has one egg. The algorithm [5] can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions. Cuckoo search is based on three idealized rules which states that

i. Each Cuckoo lays one egg, which represents a set of solution coordinates, at a time, and dumps it in a random nest.
ii. A fraction of the nests containing the best eggs, or solutions, will be carried over to the next generation.
iii. The number of nests is fixed and there is a probability that a host can discover an alien egg. If this happens, the host can either discard the egg or the nest and this result in building a new nest in a new location.

When generating new solutions x^{t+1} for the *ith* Cuckoo, Lévy Flight is performed.

$$x_i^t = x_i^{t+1} + \alpha \oplus Levy(\lambda) \quad (25)$$

In equation (25), α > 0 is the step size which should be related to the scales of the problem of interest. In most cases, we can use α = O(L/10) where L is the characteristic scale of the problem of interest. The above equation is essentially the stochastic equation for a random walk. The product ⊕ means entry wise multiplications.

The Lévy flight essentially provides a random walk whose random step length is drawn from a Lévy distribution

$$\text{Lévy} \sim u = t^{-\lambda}, (1 < \lambda < 3) \quad (26)$$

This has an infinite variance with an infinite mean. Here the steps essentially form a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by Lévy walk around the best solution obtained so far, this will speed up the local search. To ensure that that the system will not be trapped in a local optimum, a substantial fraction of the new solutions should be generated by far field randomization whose locations should be far enough from the current best solution.

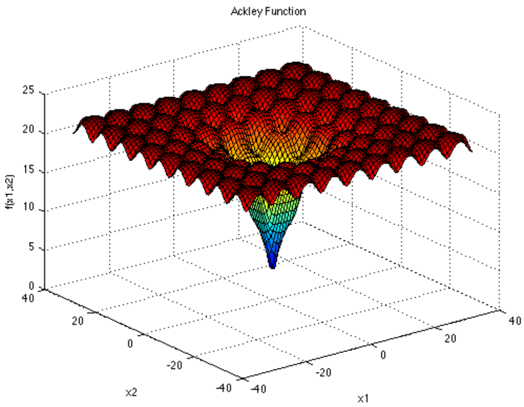
2.4. Testing Optimization Functions

In Literature [26] there are many benchmark test functions which are designed to test the performance of optimization algorithms. In this paper we will compare and validate above mentioned algorithms against these benchmark functions. Seventeen functions [27,28,29] including many multimodal functions are used in this paper in order to compare and verify efficiency and convergence of all three above mentioned Nature Inspired Algorithms. Certain test functions used in our simulations are as follows:

Ackley Function is multimodal function widely used for testing optimizat1ion algorithms.

$$\boldsymbol{f(x) = -20\, exp\left[-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right] - exp\left[\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)\right] + (20 + e)}$$

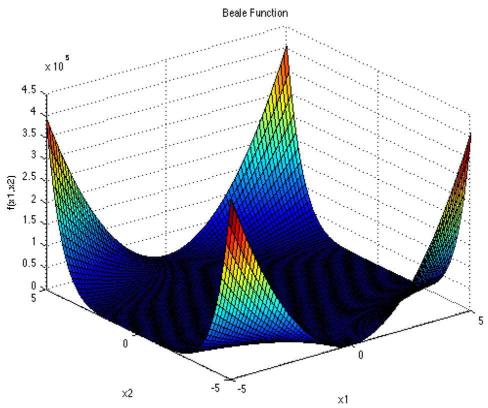
With a global minimum $f(x_*) = 0$ at $x_* = (0,0,\dots,0)$ in the range of x_i ∈ [-32.768, 32.768], for all i = 1,2, …, d.



Beale function is 2- dimensional multimodal function, with sharp peaks at the corners of the input domain.

$$f(\mathbf{x}) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$$

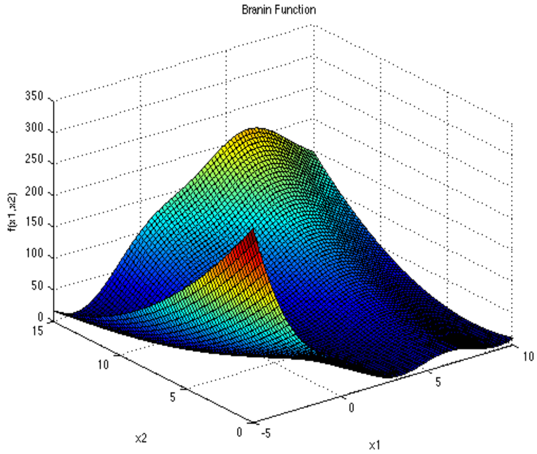
Which has minimum $f(x_*) = 0, at\ x_* = (3{,}0.5)$ in range $x_i \in$ [-4.5, 4.5], for all i = 1, 2.



Branin function is multimodal with three global minima. The recommended values of a, b, c, r, s and t are: a = 1, b = 5.1 ⁄ (4π2), c = 5 ⁄ π, r = 6, s = 10 and t = 1 ⁄ (8π).

$$f(\mathbf{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s$$

With a global minimum $f(x_*) = 0.397887$ at $x_* = (-\pi, 12.275), (\pi, 2.275)\ and\ (9.42478{,}2.475$ in the range $x_i \in$ [-5, 10] and x2 ∈ [0, 15].



Colville is a unimodal test function.

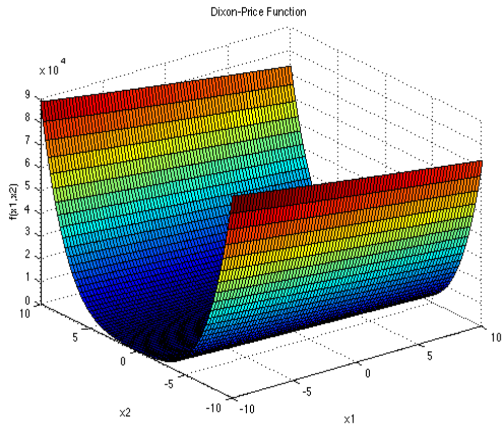
$$f(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (x_1-1)^2 + (x_3-1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2-1)^2 + (x_4-1)^2) + 19.8(x_2-1)(x_4-1)$$

Which has minimum $f(x_*) = 0, at\ x_* = (1{,}1{,}1{,}1)$ in range $x_i \in$ [-10, 10], for all i = 1, 2, 3, 4.

DIXON-PRICE's unimodal test function

$$f(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^{d} i\,(2x_i^2 - x_{i-1})^2$$

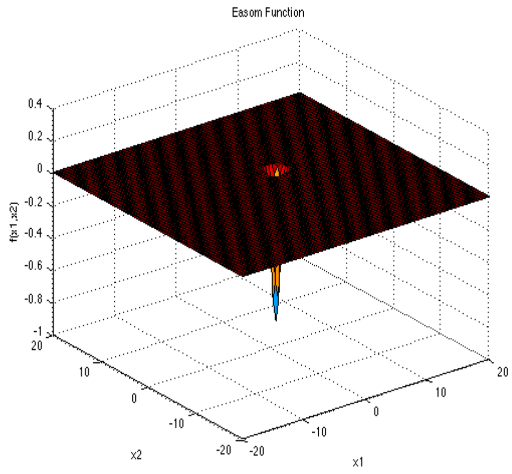
With a global minimum $f(x_*) = 0$ at $x_i = 2^{-\frac{2^i - 2}{2^i}}$ bin the range of $x_i \in$ [-10, 10], for all i = 1, 2, …, d.



Easom function has several local minima. It is unimodal, and the global minimum has a small area relative to the search space.

$$f(\mathbf{x}) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1 - \pi)^2 - (x_2 - \pi)^2\right)$$

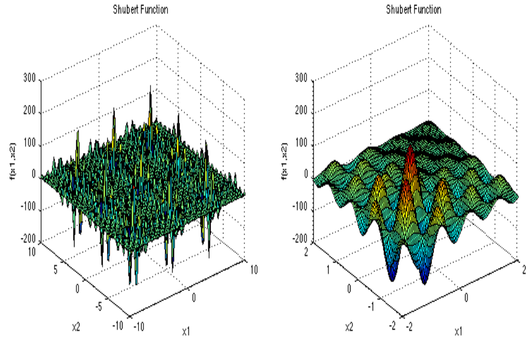
Which has minimum $f(x_*) = -1, at\ x_* = (\pi, \pi)$ in range $x_i \in$ [-100,100], for all i = 1, 2,



Shubert function an multimodal test function has several local minima and many global minima. Its equation is

$$f(\mathbf{x}) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$$

Which has minimum $f(x_*) = -186.7309$, in range $x_i \in [-10, 10]$, for all i = 1, 2,

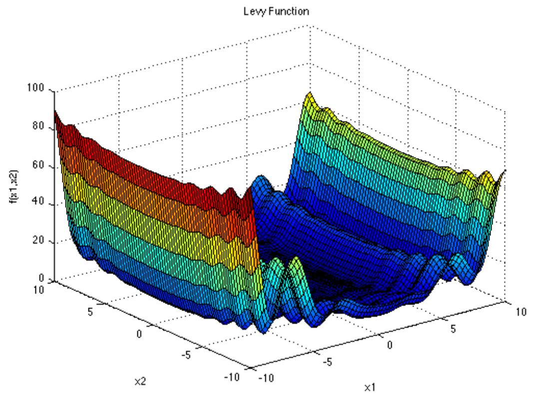


Levy Test function is multimodal function. Its equation is

$$f(\mathbf{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2\left[1 + 10\sin^2(\pi w_i + 1)\right] + (w_d - 1)^2\left[1 + \sin^2(2\pi w_d)\right], \text{ where}$$

$$w_i = 1 + \frac{x_i - 1}{4}, \text{ for all } i = 1, \ldots, d$$

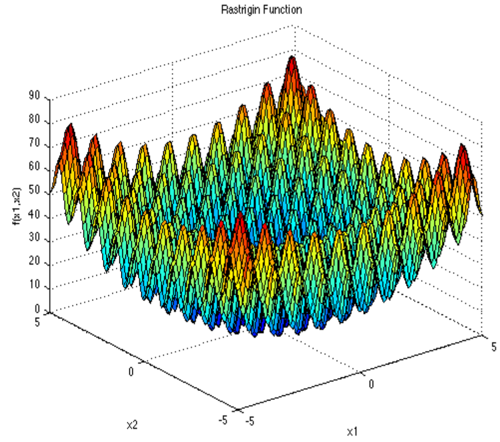
With a global minimum $f(x_*) = 0$ at $x_i = (1, \ldots, 1)$ which is evaluated in the range of $x_i \in [-10, 10]$, for all $i = 1, 2, \ldots, d$.



Rastrigin function has several local minima. It is highly multimodal, but locations of the minima are regularly distributed.

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$$

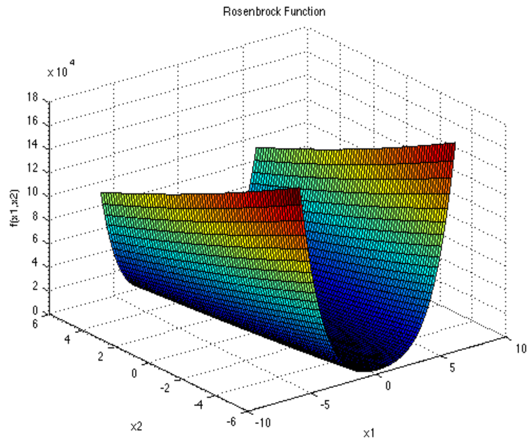
With a global minimum $f(x_*) = 0$ at $x_* = (0, 0, \ldots, 0)$ evaluated in the range of $x_i \in [-5.12, 5.12]$, for all $i = 1, 2, \ldots$, d.



Rosenbrock function is unimodal, and the global minimum lies in a narrow, parabolic valley.

$$f(\mathbf{x}) = \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$$

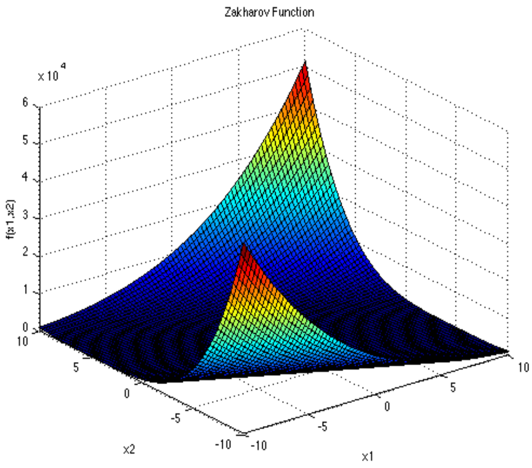
With a global minimum $f(x_*) = 0$ at $x_i = (1, \ldots, 1)$ which is evaluated in the range of $x_i \in [-5, 10]$, for all $i = 1, 2, \ldots$, d.



Zakharov function has no local minima except the global one. It's a unimodal function and its equation is

$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^4$$

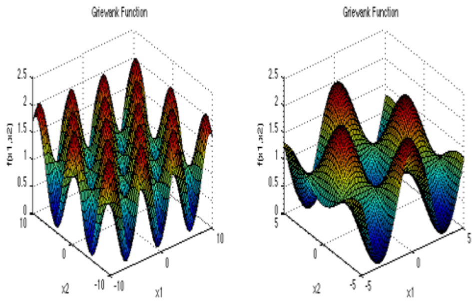
With a global minimum $f(x_*) = 0$ at $x_i = (0, \ldots, 0)$ which is evaluated in the range of $x_i \in [-5, 10]$, for all $i = 1, 2, \ldots$, d.



Griewank function has many widespread local minima, which are regularly distributed. It is a multimodal test function.

$$f(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

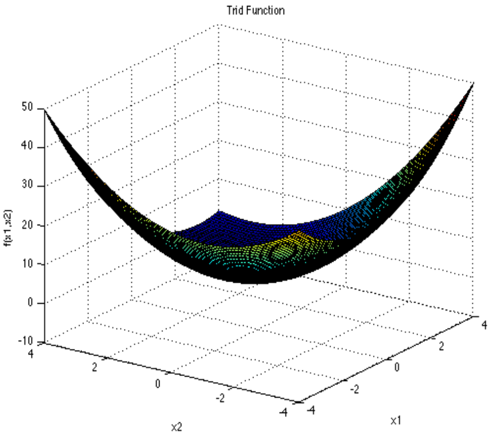
With a global minimum $f(x_*) = 0$ at $x_i = (0, \dots, 0)$ which is evaluated in the range of $x_i \in$ [-600, 600], for all i = 1, …, d.



Trid function has no local minimum except the global one. It is a unimodal function.

$$f(\mathbf{x}) = \sum_{i=1}^{d} (x_i - 1)^2 - \sum_{i=2}^{d} x_i x_{i-1}$$

With a global minimum $f(x_*) = -50$ for d=6 and $f(x_*) = -200$ at d=10 which is evaluated in the range of $x_i \in [-d^2, d^2]$, for all i = 1,2, …, d.



Powell Function is a unimodal function

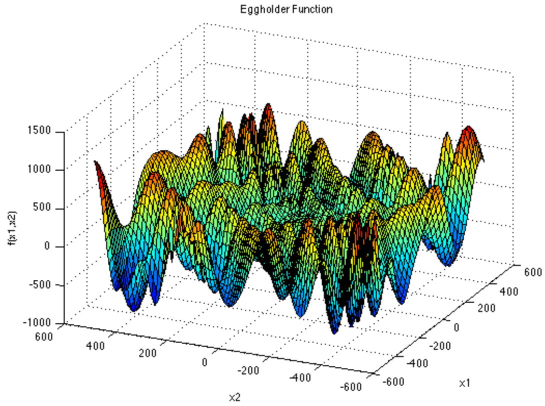
$$f(\mathbf{x}) = \sum_{i=1}^{d/4} \left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4\right]$$

This function is usually evaluated on the region $x_i \in$ [-4, 5], for all i = 1, …, d. having minima $f(x_*) = 0$ at $x_i = (3, -1, 0, 1, \dots .3, -1, 0)$

Eggholder function is a difficult function to optimize, because of the large number of local minima. It is multimodal test function.

$$f(\mathbf{x}) = -(x_2 + 47) \sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - x_1 \sin\left(\sqrt{|x_1 - (x_2 + 47)|}\right)$$

Which has minimum $f(x_*) = -959.6407, at\ x_* = (512, 404.2319)$ in range $x_i \in$ [-512, 512], for all i = 1, 2.



3. Implementation and Numerical Experiments

In this section we will compare the performance of Krill Herd algorithm Firefly algorithm, Cuckoo search algorithm for various benchmark test functions. The benchmarks function include both unimodal and multimodal with both low and high dimensional problems are described in Section 2.4 and for evaluation all computational procedures described above has been implemented in MATLAB™ computer program. In order to compare these algorithms we have carried out extensive simulations and each algorithm has been run 50 times so as to carry out meaningful analysis. The maximum number of function evaluations is set as 10,000 for high dimensional functions and 1000 for low dimensional functions.

Here for Krill Herd Algorithm C_t is set to 0.5 and the inertia weights (ω_n, ω_f) are equal to 0.9 at beginning of search and it linearly decreases to 0.1 at the end. For Firefly algorithm certain constants are fixed as $\alpha = 0.5$, $\beta = 0.2$ and $\gamma = 1$ for simulation. For cuckoo search algorithm probability for host bird is fixed as $p_\alpha = 0.25$. For simulation we have used various population sizes from n = 10 to 150, and found that for most problems, it is sufficient to use n = 15 to 50. Therefore, we have used a fixed population size of n = 50 in all our simulations for comparison.

Now we will divide this section in two parts comprising comparison of algorithms for unimodal test function in first section and comparison of algorithms for multimodal test functions in another. For both the section we will be comparing KH algorithm, FA Algorithm and CS Algorithm via Lévy Flights on the basis of three criteria i.e. Optimization fitness (efficiency), Convergence and processing time.

3.1. Optimization for Unimodal Test Function

In this section we have compared KH algorithm, FA Algorithm and CS Algorithm via Lévy Flights on Eight Unimodal benchmark functions popular for optimization. Unimodal functions are those function which have only single local minima and these function easier to optimize.

3.1.1. Optimization fitness

Here we have calculated the mean fitness value using the above mentioned algorithms for all unimodal test functions mentioned above. Optimized fitness result where global optima is reached are summarized in Table 1.

Table 1. Comparison of Optimization Fitness for Unimodal Test Functions

Function/ Algorithms	**KH Algorithm**	**FFA Algorithm**	**CS Algorithm**
Dixon-Price(d=20)	0.6975	0.6668	0.6667
Rosenbrock (*d*=20)	31.89	14.49	2.63e-14
Zakharov(*d*=20)	5.973	3.19e-06	2.77e-28
Powell(*d*=20)	0.0299	0.002269	5.27e-09
Trid(*d*=20)	9.34e+2	1.52e+03	1.52e+03
Beale	2.92e-11	5.49e-10	1.34e-30
Easom	-0.96	-0.86	-1
Colville	-1.40e+09	-1.56e+07	-7.57e+17

Here we can see that Cuckoo Search algorithm has outperformed both Krill Herd and Firefly algorithm. For all Unimodal test function, fitness value of CS algorithm is much closer to global optima as compared to other two algorithms. But if we just compare the other two algorithms i.e. Firefly and Krill herd algorithm, their result are very close to each other, but on average, results obtained using Firefly algorithm are slightly better than results obtained using Krill Herd algorithm. As per results from Table 1 it is visible that performance of Krill Herd algorithm is better for low dimensional functions and as we move from low dimensional function to high dimensional functions fitness value for krill herd decreases i.e. distance from global minima increases. According to results in Table 1 performance of Firefly algorithm is better than Krill Herd algorithm for high dimensional functions but for low dimensional function result using Krill Herd algorithm are better than Firefly algorithm. Also in Krill Herd algorithm we have varied dimensions from d= 5 to 20 and observed that as we increases the dimensions, fitness value for function decreases.

3.1.2. Processing time

Here we will compare above mentioned algorithms on basis on processing time. Processing time is basically time consumed by algorithm to process single simulation. It includes time consumed by fixed number of iteration to solve the problem.

Table 2. Comparison of processing time in seconds for Unimodal Test Functions

Function/ Algorithms	KH Algorithm	FFA Algorithm	CS Algorithm
Dixon-Price(*d*=20)	141.73	125.56	19.69
Rosenbrock (*d*=20)	137.18	123.44	21.58
Zakharov (*d*=20)	137.66	125.66	22.92
Powell(*d*=20)	172.44	138.95	63.35
Trid(*d*=20)	155.12	146.04	30.72
Beale	13.034	11.18	1.64
Easom	12.69	12.13	2.46
Colville	13.33	12.92	2.59

From Table 2 it is quite easily visible that time taken or consumed by Cuckoo search algorithm is much less than the other two algorithms We can also compute from the Table 2 that time consumed by Firefly algorithm is less than Krill Herd Algorithm although difference is not much, In term of processing time Cuckoo search algorithm again outperform other two algorithms.

3.1.3. Convergence

In this section convergence plots of the benchmark functions for three different algorithm i.e. Krill Herd, Firefly, Cuckoo Search are compared for fixed number of iteration i.e. 10,000 iterations for high dimensional function and 1000 iteration for low dimensional function. Here we will estimate which algorithm gives potentially better and quicker convergence towards optimality. Below In Figure 1-8 Convergence Graph is plotted for all above mentioned Unimodal benchmark functions.

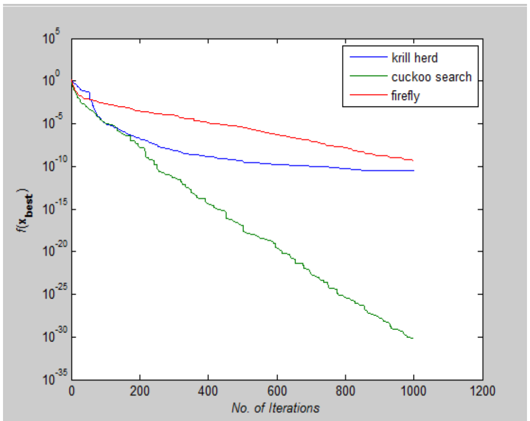


Fig. 1. Convergence Plot for Beale Function

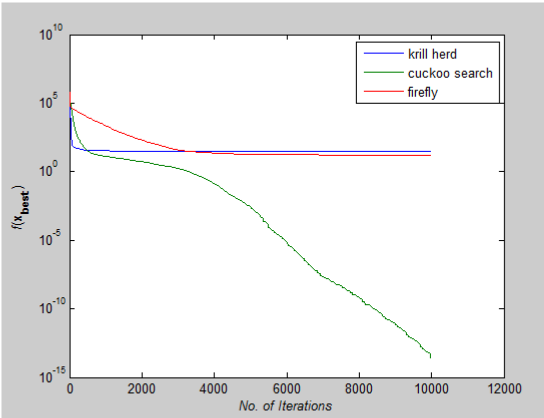


Fig. 2. Convergence Plot for Rosenbrock Function

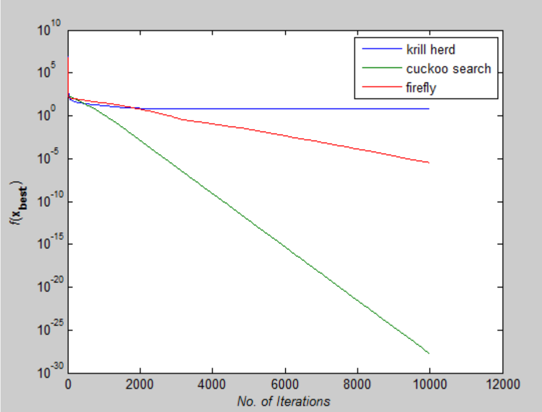


Fig. 3. Convergence Plot for Zakharov Function

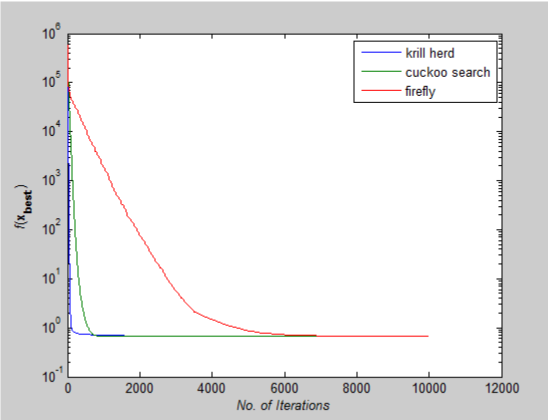


Fig. 4. Convergence Plot for Dixon-Price Function

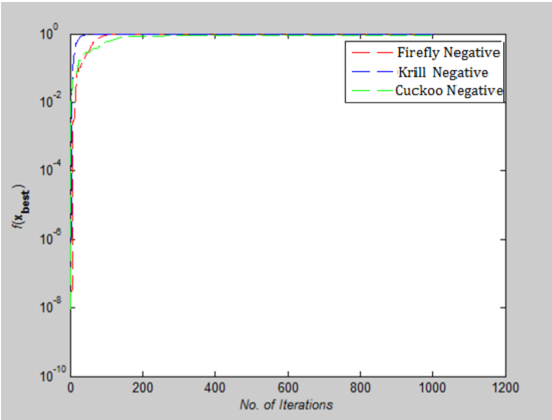


Fig. 5. Convergence Plot for Easom Function

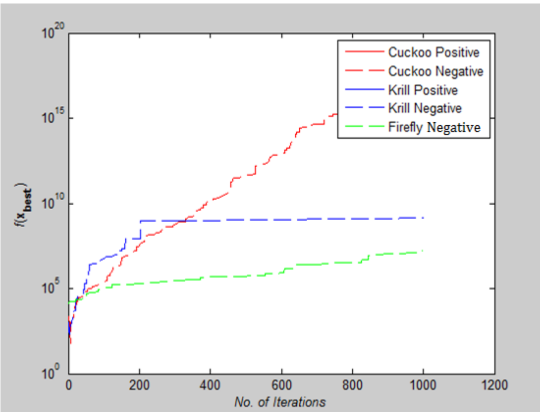


Fig. 6. Convergence Plot for Colville Function

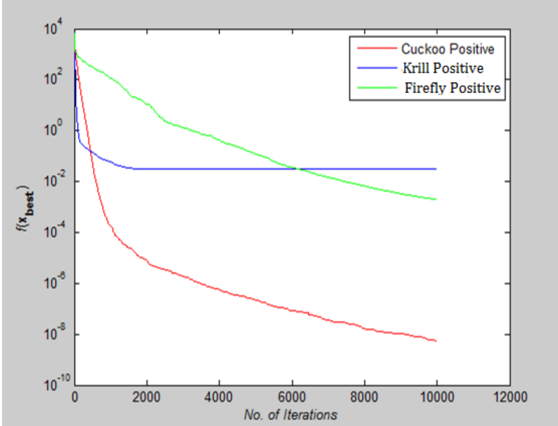


Fig. 7. Convergence Plot for Powel Function

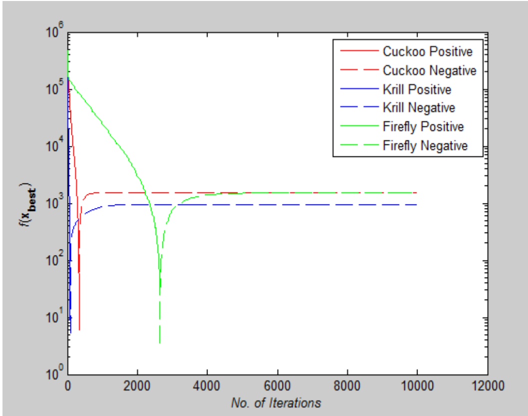


Fig. 8. Convergence Plot for Trid Function

In Fig.5, Fig.6 and Fig.8 we have plotted the absolute value of the fitness function. For Trid, Easom, Colville function, value of global minima is negative and so to plot them on convergence graph we took their absolute vale.

From Fig 1-8 we can interpret that the algorithm that quickly converges to its optimal solution is Krill Herd Algorithm. When we compare them on the number of iteration, Krill Herd Algorithm takes least number of iteration to converge whereas solution of other two algorithm are better in terms of fitness value. It can also be seen from Fig. 1-8 that for most of the test functions, the other two algorithms i.e. Firefly Algorithm and Cuckoo search algorithm do not converge till 10,000 iterations and for functions for which these two algorithm converges before 10,000 iterations, it is the cuckoo search algorithm which converges quickly than firefly for more of function as compare to number of function for which firefly algorithm converges faster than cuckoo search algorithm.

3.2. Optimization for Multimodal Test Function

In this section we have compared KH algorithm, FA Algorithm and CS Algorithm via Lévy Flights on Seven Multimodal benchmark functions popular for optimization. Multimodal functions are those function which have many number local minima and these function comparatively more difficult to optimize.

3.2.1. Optimization fitness

Here we have calculated the mean fitness value using the above mentioned algorithms for all multimodal test functions mentioned in Section 2.4. Optimized fitness result where global optima is reached are summarized in Table 3.

Table 3. Comparison of Optimization Fitness for Multimodal Test Functions

Function/ Algorithms	KH Algorithm	FFA Algorithm	CS Algorithm
Ackley(*d*=20)	1.19e-05	7.36e-03	4.44e-15
Levy(*d*=20)	0.066	2.28e-06	1.14e-06
Rastrigin (*d*=20)	8.69	20.36	2.84
Griewank (*d*=20)	1.37e-09	5.89e-04	0
Branin	0.3979	0.3981	0.3981
Shubert	-186.7309	-186.7309	-186.7309
Eggholder	-893.0205	-930.2513	-959.6407

From Table 3 we can see that Cuckoo Search algorithm has outperformed both Krill Herd and Firefly algorithm in multimodal optimization as well. For all Multimodal test function, fitness value of CS algorithm is much closer to global optima as compared to other two algorithms. But if we together compare the other two algorithms i.e. Firefly and Krill herd algorithm for multimodal test functions, result for them are in contradiction with results in previous section for unimodal test functions. In multimodal optimization for most of the low dimensional function ,result for both the algorithms are almost equivalent, for most of the time both are able to find the global optima but for high dimensional multimodal functions fitness value obtained using Krill Herd is better than fitness value obtained using Firefly Algorithm. Results in Table 3 are in contradiction with results in Table 1 as in unimodal test function optimization fitness for high dimensional function using firefly was better than krill herd but in multimodal function optimization fitness using krill herd algorithm is better than firefly algorithm for high dimensional function.

3.2.2. Processing time

In this section we will compare Krill Herd, Firefly, Cuckoo Search algorithms on basis on processing time for multimodal optimization functions. Processing time is basically time consumed by algorithm to process single simulation. It includes time consumed by fixed number of iteration to solve the problem.

Table 4. Comparison of processing time in seconds for Multimodal Test Functions

Function/ Algorithms	KH Algorithm	FFA Algorithm	CS Algorithm
Ackley(*d*=20)	138.06	123.49	24.78
Levy(*d*=20)	169.37	129.03	38.86
Rastrigin(*d*=20)	139.55	126.31	24.58
Griewank(*d*=20)	137.18	123.44	21.58
Branin	13.36	11.28	1.91
Shubert	13.83	11.33	2.56
Eggholder	13.04	11.23	2.83

Results in Table 4 are quite similar with the results in Table 2. In multimodal optimization function as well, time taken or consumed by Cuckoo search algorithm is much less than the other two algorithms It can also be interpreted from the Table 4 that time consumed by Firefly algorithm is less than Krill Herd Algorithm although difference is not much, In term of processing time Cuckoo search algorithm again outperform other two algorithms.

3.2.3. Convergence

Here convergence plots of the benchmark functions for three different algorithm i.e. Krill Herd, Firefly, Cuckoo Search are compared for fixed number of iteration i.e. 10,000 iterations for high dimensional function and 1000 iteration for low dimensional functions. Here we will estimate which algorithm gives potentially better and quicker convergence towards optimality. Convergence Graph for all above mentioned Multimodal benchmark functions are plotted in Figure 9-15.

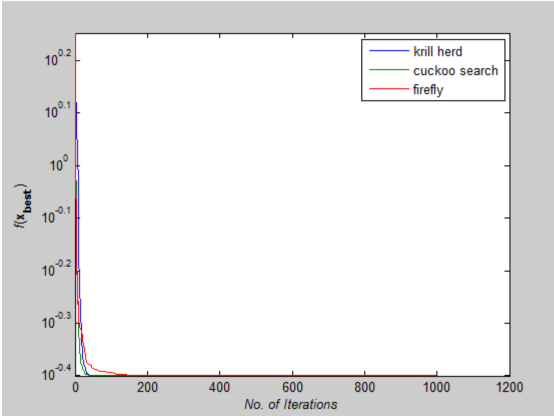


Fig. 9. Convergence Plot for Branin Function

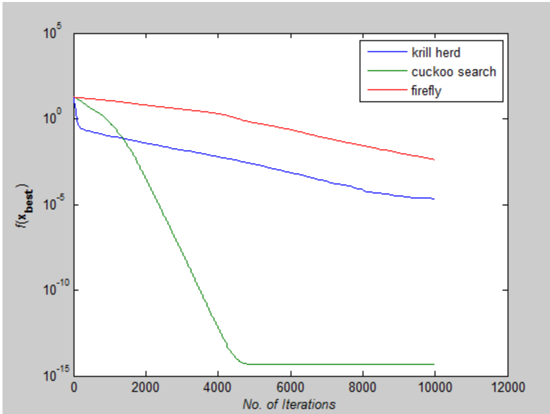


Fig. 10. Convergence Plot for Ackley Function

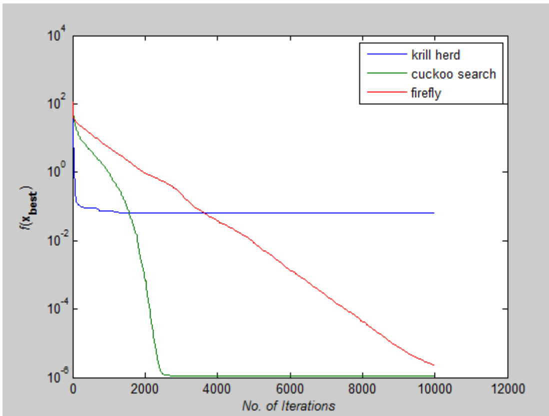


Fig. 11. Convergence Plot for Levy function

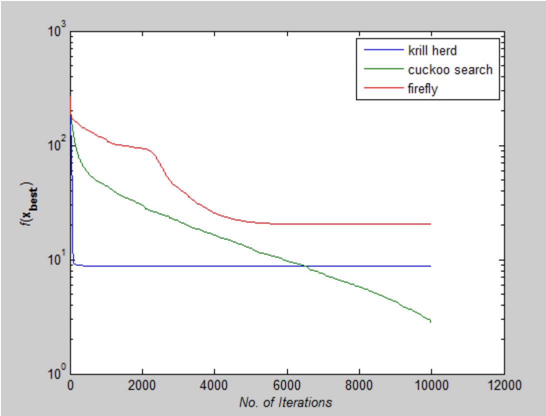


Fig.12 . Convergence Plot for Rastrigin Function

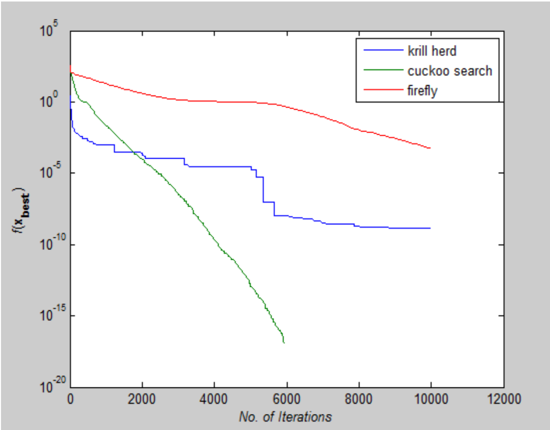


Fig. 13. Convergence Plot for Griewank Function

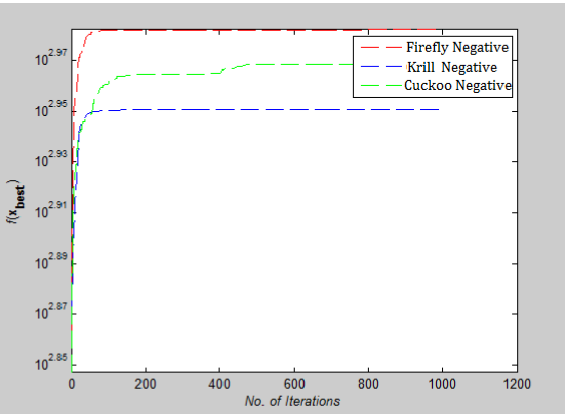


Fig. 14 . Convergence Plot for Eggholder Function

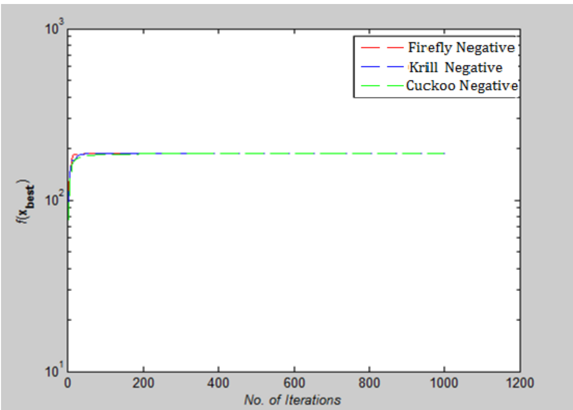


Fig. 15. Convergence Plot for Shubert Function

In Fig.14, and Fig.15 we have plotted the absolute value of the fitness function. In multimodal optimization Shubert and Eggholder function value of global minima is negative and so to plot them on convergence graph we took their absolute vale.

From Fig 9-15 we can interpret that although for many function all the algorithms are not able to converge before 10,000th iteration but for test functions like Rastrigin, Branin, Griewank, Levy and Eggholder, Krill herd algorithm is the fastest to converge to its optimal solution. When we compare them on the number of iteration, Krill Herd Algorithm takes least number of iteration to converge. If we see Fig.10 it is visible that for Ackley function Cuckoo Search Algorithm is the fastest to converge to its optimal solution others are not able to converge before 10,000th iteration. Also in Fig. 11 Cuckoo Search Algorithm is second fastest after Krill Herd algorithm to converge. From Fig 9-15 we can interpret that Firefly algorithm don not converge to its optimal solution for any of the high dimensional functions till 10,000th iteration. For multidimensional functions it is the Krill herd algorithm which is fastest to converge to its optimal solution, then after Krill Herd algorithm it is cuckoo search algorithm to converge to its optimal solution and at last is Firefly algorithm.

4. Conclusion

In this paper we have compared latest metaheuristic algorithms such as Krill Herd Algorithm, Firefly Algorithm and Cuckoo Search algorithm via Lévy Flights on basis of three criteria i.e. optimization fitness (efficiency), time processing and convergence. Results obtained by simulation of theses algorithms on unimodal and multimodal test functions shows that Cuckoo search algorithm is superior for both unimodal and multimodal test function in terms of optimization fitness and time processing whereas when comparison comes down to line between Krill Herd Algorithm and Firefly Algorithm, KH Algorithm is superior than FFA algorithm for multimodal optimization of both high and low dimensional functions whereas for unimodal optimization FFA algorithm is superior than KH Algorithm for High dimensional function but KH algorithm is superior for low dimensional functions but in terms of time processing FFA Algorithm is surpasses KH Algorithm for both unimodal and multi modal optimization. When we compare these algorithms on basis of convergence Krill Herd is the fastest of all to converge to its optimal solution after which comes the Cuckoo search algorithm and at last comes the Firefly algorithm which do not converge for most of the function to it optimal solution.

During simulation we also noticed that as we increase the dimension, fitness value or optimization fitness for KH algorithm decreases although it outperformed FFA algorithm for high dimensional multimodal function but on average as dimension increases optimization fitness for KH algorithm decreases.

As all these powerful optimization strategy are able to optimize both unimodal and multimodal test function effectively hence we can easily extend them to study multi objective optimization applications with various constraints and even to NP-hard problems.

References

[1] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms", Luniver Press, 2008.

[2] Christian Blum, Maria Jos´e Blesa Aguilera, Andrea Roli, Michael Sampels, Hybrid Metaheuristics, An Emerging Approach to Optimization, Springer, 2008 .

[3] Christian Blum, and Maria Jos´e Blesa Aguilera. Metaheuristics in Combinatorial Optimization: Overview

and Conceptual Comparison, Springer, 2008.

[4] Amir Hossein Gandomi, Amir Hossein Alavi. Krill herd: A new bio-inspired optimization algorithm, Elsvier, 2012.

[5] X.-S. Yang, S. Deb, “Cuckoo search via L´evy flights”, in: Proc. Of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, USA, pp. 210-214 (2009).

[6] Hofmann EE, Haskell AGE, Klinck JM, Lascara CM. Lagrangian modelling studies of Antarctic krill (Euphasia superba) swarm formation. ICES J Mar Sci 2004;61:617–31.

[7] Price HJ. Swimming behavior of krill in response to algal patches: a mesocosm study. Limnol Oceanogr 1989;34:649–59.

[8] Morin A, Okubo A, Kawasaki K. Acoustic data analysis and models of krill spatial distribution. Scientific Committee for the Conservation of Antarctic Marine Living Resources, Selected Scientific Papers, Part I; 1988. p.311–29.

[9] Sh. M. Farahani, A. A. Abshouri, B. Nasiri, and M. R. Meybodi, “A Gaussian Firefly Algorithm”, International Journal of Machine Learning and Computing, Vol. 1, No. December 2011.

[10] Xin-She Yang, Chaos-Enhanced Firefly Algorithm with Automatic Parameter Tuning, International Journal of Swarm Intelligence Research, December 2011.

[11] Flierl G, Grunbaum D, Levin S, Olson D. From individuals to aggregations: the interplay between behavior and physics. J Theor Biol 1999;196:397–454.

[12] Okubo A. Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. Adv Biophys 1986;22:1–94.

[13] Hardy AC, Gunther ER. The plankton of the South Georgia whaling grounds and adjacent waters, 1926–1927. Disc Rep 1935;11:1–456.

[14] Marr JWS. The natural history and geography of the Antarctic krill (Euphausia superba Dana). Disc Rep 1962;32:33–464.

[15] Nicol S. Living krill, zooplankton and experimental investigations. Proceedings of the international workshop on understanding living krill for improved management and stock assessment marine and freshwater behaviour and physiology 2003;36(4):191–205.

[16] Murphy EJ, Morris DJ, Watkins JL, Priddle J. Scales of interaction between Antarctic krill and the environment. In: Sahrhage D, editor. Antarctic Ocean and resources variability. Berlin: Springer-Verlag; 1988. p. 120–30.

[17] Brown C., Liebovitch L. S., Glendon R., Lévy flights in Dobe Ju/’hoansi foraging patterns, Human Ecol., 35, 129-138 (2007).

[18] Pavlyukevich I., Lévy flights, non-local search and simulated annealing, J. Computational Physics, 226, 1830-1844 (2007).

[19] Pavlyukevich I., Cooling down Lévy flights, J. Phys. A:Math. Theor., 40, 12299-12313 (2007).

[20] Reynolds A. M. and Frye M. A., Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search, PLoS One, 2, e354 (2007).

[21] Shlesinger M. F., Zaslavsky G. M. and Frisch U. (Eds), Lévy Flights and Related Topics in Phyics, Springer, (1995).

[22] Shlesinger M. F., Search research, Nature, 443, 281- 282 (2006).

[23] Barthelemy P., Bertolotti J., Wiersma D. S., A Lévy flight for light, Nature, 453, 495-498 (2008).

[24] Viswanathan, G. M. et al. Optimizing the success of random searches. Nature 401, 911–914(1999)

[25] Bartumeus, F. et al. Optimizing the encounter rate in biological interactions: Lévy versus Brownian strategies. Phys. Rev. Lett. 88, 097901 (2002)

[26] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evolut Comput 1999;3:82–102.

[27] www.sfu.ca/~ssurjano/optimization.html

[28] www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm

[29] MominJamil,Xin-SheYang, Hans-Ju¨rgenZepernick. ” Test Functions for Global Optimization: A Comprehensive Survey” in Swarm Intelligence and BioInspired Optimization, Elsevier, Part I; 2013.p. 193-222

International Journal of
Intelligent Systems and
Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Development Of HealthCare System For Smart Hospital Based On UML and XML Technology

Magdy Shayboub[1] Ali Mahmoud*[2]

Received 26th December 2013, Accepted 22th January 2014

Abstract: The convergence of information technology systems in health care system building is causing us to look at more effective integration of technologies. Facing increased competition, tighter spaces, staff retention and reduced reimbursement, today's traditional hospitals are looking at strategic ways to use technology to manage their systems called smart hospital. The concept of the smart hospital is a useful system for any hospital; about adding intelligence to the traditional hospital system by covering all resources and locations with patient information. Patient's information is an important component of the patient privacy in any health care system that is based on the overall quality of each patient in the health care system. The main commitment for any health care system is to improve the quality of the patient and privacy of patient's information. Today, there is a need of such computer environment where treatment to patients can be given on the basis of his/her previous medical history at the time of emergency at any time, on any place and anywhere. Pervasive and ubiquitous environment and UML (unified modeling language) can bring the boon in this field. For this it's needed to develop the ubiquitous health care computing environment using the UML with traditional hospital environment. This paper is based on the ubiquitous and pervasive computing environment based on UML and XML(The Extensible Markup Language) technology, in which these problems has been tried to improve traditional hospital system into smart hospital in the near future. The key solution of the smart hospital is online identification of all patients, doctors, nurses, staff, medical equipments, medications, blood bags, surgical tools, blankets, sheets, hospital rooms, etc. In this paper efforts is channeled into improving the knowledge-base ontological description for smart hospital system by using UML and XML technology, Our knowledge is represented in XML format from UML modeling(class diagram). Our smart hospital provides access to its system by using a smart card. Finally, the former try to improve health care delivery through development and management of acute care hospital designed; both physically and operationally, for more efficiency and increased patients safety.

Keywords: *UML, Smart Hospital (SH), Ontology, XML, health care system.*

1. Introduction

With more than 90 percent of hospital administrators involved in constructing a new building or renovating an existing facility to meet the ever-increasing demands for space today, traditional hospital executives have to look closer at their work flow processes earlier in the program, in order to capitalize on the latest program technology to optimize clinical, financial and administrative processes. And it involves more than advanced healthcare information systems. It also includes assistance technology such as medical smart card, advanced nurse call, and advanced patient tracking.

There are many organizational units or departments in the traditional hospital, from which, it is necessary for them that there should be good coordination in each other. Even the available health care ontology automation software also does not provide such coordination among them. These software are limited up to the hospital works but do not provide the interconnectivity with other hospitals and blood banks etc. Thus, these traditional hospitals cannot share information in-spite of the good facilities and services.

[1] *Computer Science Department, Faculty of Computers and Informatics, Ismailia 41522, Suez Canal University, Egypt*
[2] *Computer Science Department, Faculty of Computers and Information systems Taif University, Kingdom of Saudi Arabia (KSA)*
* *Corresponding Author: Email: magdy01sh@yahoo.com*

Many changes and developments in health care environment in the last decades are due to new technologies such as portable devices (Laptop, Mobile) and wireless computing. On the one hand, where the main aim of traditional hospital is to provide better services and facilities to the patient, his/her proper care brings success to the hospital's name. Along with this, traditional hospitals also add many new facilities and services with existing facilities and services in one place for their patient. Having all facilities and services in the same place, hospitals are able to provide sufficient care to the patient at any place and time.

The major problems with the health care environments are related to the information storage and retrieval of the patient's data and other entities of the health care system. These problems are further categorized below [1-3]:¬-

- One problem is when there is information gap among the medical professionals, users/patients and various data source.
- Another problem is that in there is a need to present and organize the information flow among the hospital members and other entities so that information can be accessed at any time and any place.
- Other problems are related to the various types of data used and no common format for holding it in a common way.

2. Concept and design ontology

The concept of smart hospital has been designed from the ground up to achieve the following goals:

- Safety and clinical quality
- Productivity
- Ease of use for patients, doctors, families and caregivers
- Service excellence
- Optimal use of technologies- medical, information and consumer

There has been much development in the concept of ontology process since the last decade and many good thinkers gave its meaning and its various definitions [1, 3, 6]. It is a set of primitive concepts that can be use for representing a whole domain or part of it that defines a set of objects, relations between them and subsets of those in the respective domain. It is also a man-made framework that supports the modeling processes of a domain in such a way that collection of terms and their semantic interpretation is provided. Our knowledge design in our paper [2, 3] is represented in XML format. In artificial intelligence [13] the term-Ontology is an explicit specification of a conceptualization, where ontology is defined as:

- a vocabulary ; the set of terms used for modeling.
- a structure of the statements in the model.
- the semantic interpretation of these terms.

Ontologies have become ubiquitous [7, 9] in information systems. They constitute the Semantic Web's [4] backbones, facilitate e-commerce, and serve such diverse application fields as bioinformatics and medicine. Many times the meaning of the word 'Ontology' is taken as a branch of philosophy [4] that is the science, of the kinds and structures of objects, properties, events, processes and relations in every area of reality. Sometimes, it is used as a synonym of 'metaphysics' and having broader sense which refers to the study of what might exist and which of the various alternative possible ontologies is in fact true of reality. In simple term, Ontology can be defined as a collection of Classes, Sub-classes that makes the relationship among them and represent the ontology design with knowledge base. Our knowledge base of ontology design is represented in XML format.

3. Smart Hospital

The smart hospital accomplishes these goals by taking integrated and current information and communication technologies and combining them with:

- Careful design of the facility to be accessible and efficient,
- Initial engineering and continual redesign of clinical and business processes to operate reliably and safely,
- Constant emphasis on patient and family service and satisfaction, and
- Fervent attention to providing a superior workplace for physicians and staff.

The end result is that smart hospital patients and families experience coordinated, safe, and high quality care in an information-rich, service-minded and easily accessible environment. Physicians affiliated with the smart hospital experience an efficient and clinically focused facility in which to practice state-of-the art evidence-based medicine [24-27]. Caregivers and technicians at the smart hospital devote time and energy to doing what they do best caring for patients. Figure 1a, show an use-case diagram of a simple smart hospital. It is a type of hospital that is able to share the domain's knowledge with same or other domain [16, 18] and fulfill the requirement of the ubiquitous and pervasive computing [6] environment. The smart hospital offers a number of advantages:¬

- It provides a beneficial strategy for the better education and training simulation among the health care professionals.
- It ensures the higher levels of competence, confidence and critical thinking skills.
- It helps to manage the complex and changing health care system.
- It also supports the faculty for developing and evaluating new educational models, modalities, and teaching-learning strategies at no risk to patients.
- It also helps to integrate the better combination of ICT technologies, product and services.

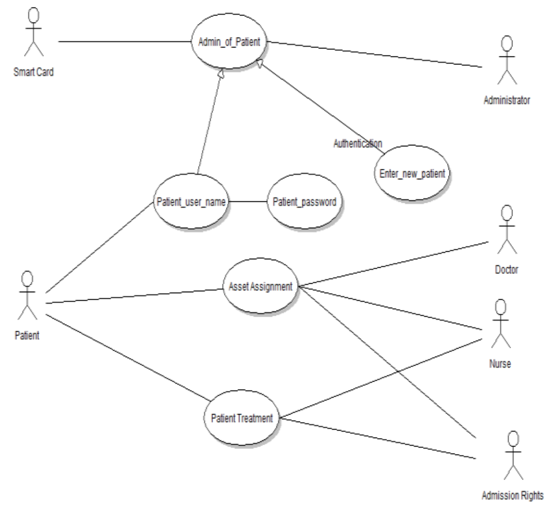


Figure 1a. Use-case diagram

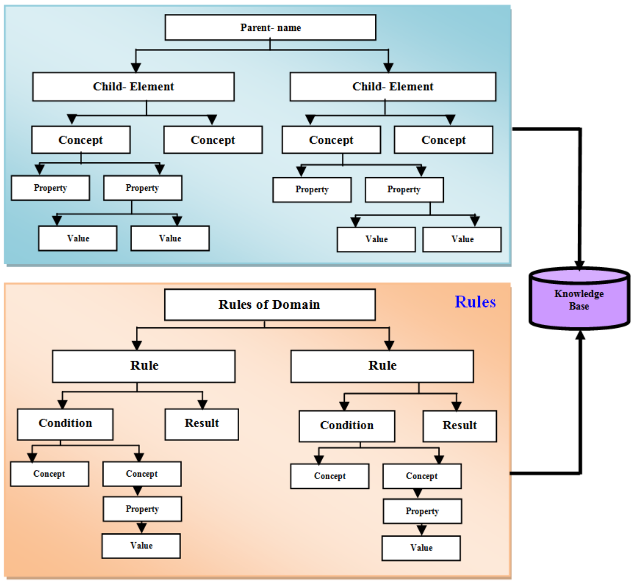


Figure 1b. Overall knowledge structure

4. Ontology Design for Smart Hospital (SH)

The ontology of health care system is a major component where end user interacts with it and the information encompasses a conceptual component i.e. information that plays a role in hospital care outcomes, including errors and difficulties. To deal with the events, Deployment of SH in a particular hospital setting will involve developing the middleware to relate the ontological knowledge base [1, 10] with existing information systems and by creating instances of ontological categories that is based on the information in the smart hospital databases [16]. Our knowledge base [1] is represented in XML format from UML modeling. Knowledge representation has been defined as "A set of syntactic and semantic conventions that makes it possible to describe things [2-5]. The syntax of a representation specifies a set of rules for combining symbol to form expressions in the representation language as shown in figure 1b. The semantics of a representation specify how expressions so constructed should be interpreted (i.e. how meaning can be derived from a form). In the proposed system, the knowledge representation methodology uses XML format. Where, two elements of knowledge, facts and model rules are represented using XML format as shown in figure 2. The overall knowledge structure is:


```
?XML version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" XMLns=http://schemas.microsoft.com/developer/msbuild/2003
ToolsVersion="3.5"> <PropertyGroup>
  <ProductVersion>9.0.21022</ProductVersion>
  <SchemaVersion>2.0</SchemaVersion>
  <RootNamespace>Untitled</RootNamespace>
  <AssemblyName>Untitled</AssemblyName>
  <ProjectGuid>{}</ProjectGuid>
  <OutputType>Library</OutputType>
 </PropertyGroup>
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <DebugSymbols>true</DebugSymbols>
  <DebugType>full</DebugType>
  <Optimize>false</Optimize>
  <OutputPath>bin\Debug\</OutputPath>
  <DefineConstants>DEBUG;TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
 </PropertyGroup>
 <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <DebugType>pdbonly</DebugType>
  <Optimize>true</Optimize>
  <OutputPath>bin\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
 </PropertyGroup> <ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.XML" />
 </ItemGroup> <ItemGroup> <Compile Include="PatientSmartCard.cs" />
  <Compile Include="Patient.cs" />
```


Figure 2. XML Sample of developed facts in our knowledge.

5. Smart Hospital Schema

The Smart Hospitals schema uses wireless technology to facilitate communication internally and externally. These schemes can allow a legal user to login to remote server and access its facilities. Its consumer technology improves the flow of information to customers using text messaging pagers, PDAs, tablet PCs, and customer internet access. Electronic medical records, bar coding and integrated vital signs are among the additional technologies that result in such operational efficiencies as:

- Reduced documentation time for nurses, allowing them to spend more time giving patient care.
- Immediate access to diagnostic test results:
- Access to patient medical information by physicians from office, home or elsewhere.
- Improved patient safety through automated checking of medication administration.
- Space savings resulting from digital storage of "films" and other medical records.
- Daily access and feedback to management and physicians of clinical information, financial results and patient satisfaction.

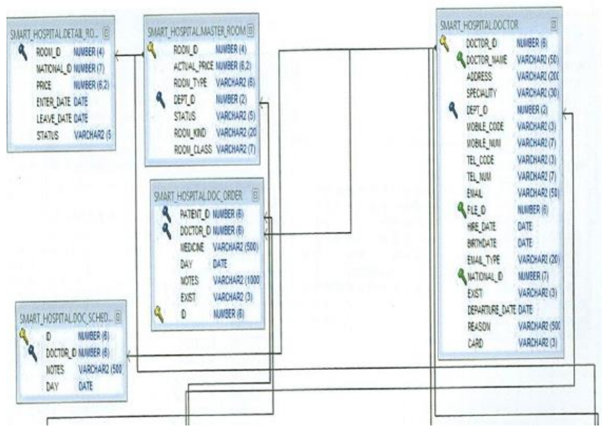


Figure 3a. Smart Hospital Schema 1.

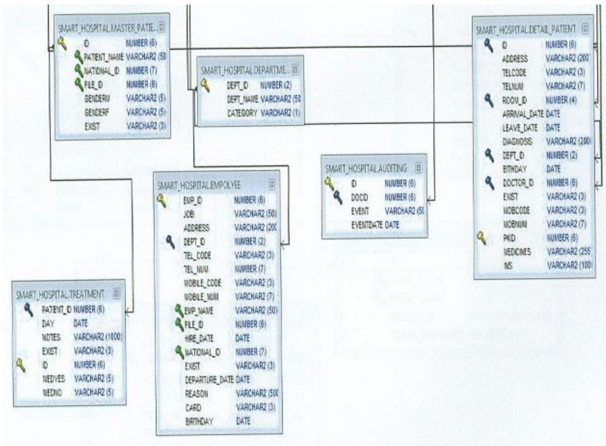


Figure 3b. Smart Hospital Schema 2

In the above figure 3a,3b and 3c show the components of Smart hospital schemas, which are events, actions, person, policies, alerts etc. For example, in the SH different type of objects is taken such as-agents, policies, record, drugs, place and equipment etc.:

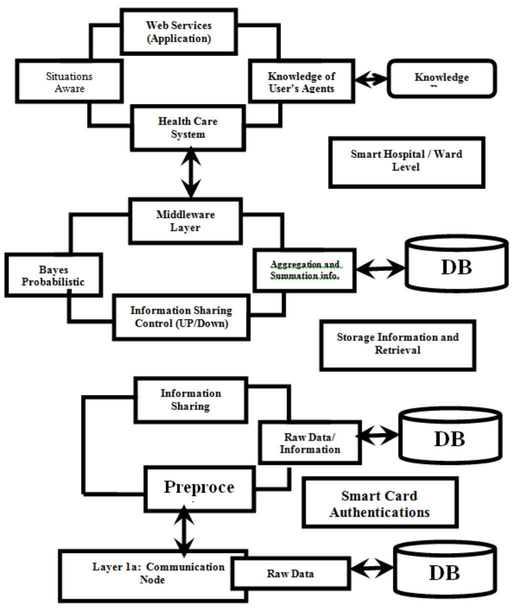


Figure 4. Middleware Architecture of patient data collection system for SH

Figure 4, illustrates A middleware architecture of a patient data collection system for SH ontology. So, this ontology is able to describe which action and event is performed in what time and what place. This is also useful to alert the different type of domain time to time with different type of alters such as-medical condition alert, medical conflict alert, regulation action and regulatory conflict alert. The major benefits of ontology in health care environment are To find out the common understanding of the structure of information among hospital entities or software agents and share it [10-15].

- Domain assumptions can be made explicit.
- To separate domain knowledge from the operational knowledge.
- To analyze domain knowledge

Often ontology of the domain is not a goal in itself. Developing ontology is defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases [1] built from ontologies as data. For example, we develop ontology of patient, doctor and nurse and appropriate combinations of patient with doctor and nurse.

6. Methodlogy and Approach

A simple approach used to develop the ontology/knowledge base is the iterative technique that is used to identify the various super classes and sub classes and its properties which is based on the simple knowledge/ontology engineering [16-18] methodology. This methodology is described as shown in figure 5:

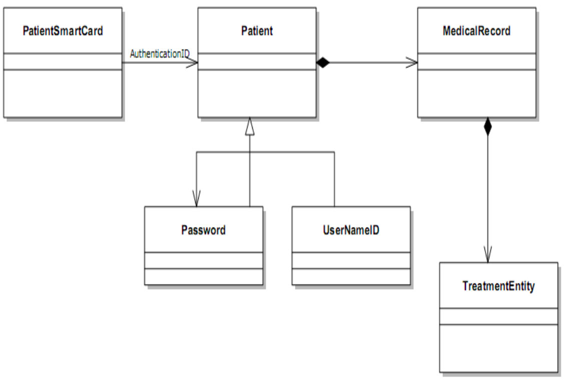


Figure 5. Class Diagram of a Patient Record in SH

6.1. Our Database Methodology:

The Oracle forms and Report Builder are used primarily by application developer. The results of these efforts are very rarely used to the end users like Doctors and Patients, it takes so much of time to explain the syntax of SQL even to do simple queries. However some advanced users will be interested in knowing other DBA tools. They need to memorize the commands and function names as the query system will not provide any prompting. The enterprise manager is a tool created to help DBAs monitor the database and perform basic tasks. It's a graphical interface that runs in a browser over the internet. Consequently, it is relatively easy to use, even for beginning DBAs, and makes the database controls accessible from more locations.

The main drawback is that only a DBA may use the enterprise manager. If you have installed a copy of the oracle DBMS in your machine, you can use the system account which has the DBA role. If you are limited to accessing a shared computer in a lab setting, the DBA will not want to give the DBA role to students; so you probably will not be able to run the exercises in this book that use the enterprise manager. So, Our database tables as shown below.

1. Auditing:

ID	DOCID	EVENT	EVENTDATE
41	100027	He/She Search Patient Treatment	6/26/2009 12:11:00 AM

2. Company:

COMPANY_ID	COMPANY_NAME	ADDRESS	TELCODE	TELNUM	EXIST
1	nike	1	2	3	YES

3. Department :

DEPT_ID	DEPT_NAME	CATEGORY
28	Rooms	E

4. Detail_billp:

ID	FKID	MEDICINE	PRICE	QUANTITY	TOTAL

5. Detail_medicine:

ID	MED_ID	COST_PRICE	SAL_PRICE	QUANTITY	START_DATE	EXPIRE_DATE	COMPANY_ID	SUPPLIER_ID	EXIST
63	60	1	1	3	6/20/2009	6/20/2010	2	2	YES

6. Detail_patient:

ID	ADDRESS	TELCODE	TELNUM	ROOM_ID	ARRIVAL_DATE	LEAVE_DATE	INS
300022	Ismailia,nfysha,al kilo 6	064	3029202	118	3/30/2009 2:13:54 AM		

DIAGNOSIS	DEPT_ID	BITHDAY	DOCTOR_ID	EXIST	MOBCODE	MOBNUM	PKID	MEDICINES
entrobious vermicularis eggs in stool	16	3/6/2005	100003	YES	010	896788	22	

7. Detail_room:

ROOM_ID	NATIONAL_ID	PRICE	ENTER_DATE	LEAVE_DATE	STATUS
85	2102965	120	6/25/2009 11:49:00 AM		FULL

8. Doc_order:

PATIENT_ID	DOCTOR_ID	MEDICINE	DAY	NOTES
300003	100004	iron supplement 1 amp everyday for two weeks	4/6/2009 11:37:00 PM	follow up for antenatal care health education

	EXIST	ID
for 1.good nutrition and rest 2.following hyginic habbits 3.risk signs	YES	2

9. Doc_schedule:

ID	DOCTOR_ID	NOTES	DAY
1	100026	go to hospital	6/25/2009 1:05:00 PM

10. Doctor:

DOCTOR_ID	DOCTOR_NAME	ADDRESS	SPECIALITY	DEPT_ID	MOBILE_CODE	MOBILE_NUM
100006	shaimaa asem mohamed	Cairo, el maady ,block 5	internal medicine	17	019	2341987

TEL_CODE	TEL_NUM	EMAIL	FILE_ID	HIRE_DATE	BIRTHDATE	EMAIL_TYPE	NATIONAL_ID	EXIST	DEPARTURE_DATE	REASON	CARD
064	3202040	sh.mohamed	100006	3/28/2009	2/17/1982	@Gmail.com	1701250	YES			YES

11. Employee:

EMP_ID	JOB	ADDRESS	DEPT_ID	TEL_CODE	TEL_NUM	MOBILE_CODE	MOBILE_NUM
31	room admistrator	Ismailia,al sheikh zaid, no 123	28	064	3892198	014	8182712

EMP_NAME	FILE_ID	HIRE_DATE	NATIONAL_ID	EXIST	DEPARTURE_DATE	REASON	CARD	BIRTHDAY
nancy mohamed mohsen	31	4/13/1999	2938283	YES			YES	6/13/1978

12. Master_billp:

ID	PHARMACY_ID	DATEE	TIMEE	BILLTYPE

13. Master_medicine:

ID	PAR_CODE	MED_NAME	MED_USE	TEMP_QUANTIT
1	2324567876	haematon	iron	23

14. Master_patient:

ID	PATIENT_NAME	NATIONAL_ID	FILE_ID	GENDERM	GENDERF	EXIST
300001	alia mohamed nageb	1301001	300001	False	True	YES

15. Master_room:

ROOM_ID	ACTUAL_PRICE	ROOM_TYPE	DEPT_ID	STATUS	ROOM_KIND	ROOM_CLASS
138	200	Double	17	FULL	Room In Patient	Class A

16. Supplier:

SUPPLIER_ID	SUPPLIER_NAME	ADDRESS	MOBCODE	MOBNUM	TELCODE	TELNUM	EXIST
1	soda	1	2	2	1	1	YES

6.2. Knowledge Base Methodology

No specified methods or approaches are still developed for the development of ontology. Our methodology depends on an iterative technique to ontology development. All the steps in this technique are revised and refined in the process of iterative technique to evolve the ontology. The process in iterative design is likely to be continued through the entire development lifecycle of the ontology. Based on the various literature survey, the proposed steps for the processing of developing ontology are:¬

(i) Finding the domain and scope of the ontology: The first step of the development of ontology is to determine and define its domain and scope. During the determination and definition of it, we must have to consider the following four questions so that we can be able to easily determine it:

- What is the domain that the ontology will cover?
- What are we going to use the ontology?
- What types of questions should the information in the ontology provide answers?
- Who will use and further develop the ontology?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model. Figure 5, show the class diagram of a Patient Record in smart hospital and the XML format from this figure as shown:


```
Project DefaultTargets="Build" XML ns="http://schemas.microsoft.com/developer/msbuild/2003">
<PropertyGroup>  <ProductVersion>8.0.50727</ProductVersion>  <SchemaVersion>2.0
</SchemaVersion>  <RootNamespace>Untitled</RootNamespace>  <AssemblyName>Untitled
</AssemblyName>  <ProjectGuid>{}</ProjectGuid>  <OutputType>Library</OutputType>
</PropertyGroup> <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
<DebugSymbols>true</DebugSymbols>  <DebugType>full</DebugType>  <Optimize>false</Optimize>
<OutputPath>bin\Debug\</OutputPath>  <DefineConstants>DEBUG;TRACE</DefineConstants>
<ErrorReport>prompt</ErrorReport>  <WarningLevel>4</WarningLevel> </PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
<DebugType>pdbonly</DebugType>  <Optimize>true</Optimize>
<OutputPath>bin\Release\</OutputPath>  <DefineConstants>TRACE</DefineConstants>
<ErrorReport>prompt</ErrorReport>  <WarningLevel>4</WarningLevel> </PropertyGroup>
<ItemGroup>  <Reference Include="System" />  <Reference Include="System.Data" />
<Reference Include="System.XML " /> </ItemGroup> <ItemGroup>
<Compile Include="PatientSmartCard.cs">  <Compile Include="Patient.cs" />
<Compile Include="MedicalRecord.cs" />  <Compile Include="Password.cs" />
<Compile Include="UserNameID.cs" />  <Compile Include="TreatmentEntity.cs" />
</ItemGroup> <Import Project="$(MSBuildBinPath)\Microsoft.CSharp.targets" /> </Project>
```


(ii) Consider reusing existing ontology: The second step to consider is about the existing ontology. The benefit of considering the existing ontology is about what someone else has done and checking if we can refine and extend existing sources for our particular domain and task. Reusing existing ontology may be a requirement if our system needs to interact with other applications that have already committed to particular ontology or controlled vocabularies.

(iii) Enumerate important terms in the ontology-preparing vocabulary: The third step is to write down a list of all terms that are used in the system. We need to enumerate all properties that the concepts may have, or whether the concepts are classes or slots.

(iv) Define the classes and the class hierarchy: The forth step is to define the classes and it is hierarchies. There are several possible approaches to develop a class hierarchy. These are: A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts. A middleware development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts. A mix development process is a combination of the top-down and bottom-up approaches. Here, it is defined as the more salient concepts first and then generalize and specialize them appropriately.

(v) Define the properties of classes: The fifth step is to define the properties of the class. Once we have defined some of the classes, we must describe the internal structure of concepts. For each property in the list, we must determine which class it describes. In general, there are several types of object properties that can become in development of ontology [20- 27] .

(B) Define the facets of the rules: The sixth step is to define the new facets of the rules in our knowledge base. These facts can be different from about facts which describe in these features such as the value type, allowed values, the number of the values and other features of the values. The last step is to create the individual instances of classes in the hierarchy. Defining an individual instance of a class requires: Choose a class, Create an individual instance of that class, and Filling in the Fact values. The following figures [6.a],[6.b] shows the use of class diagram for patient treatment and Asset Assignment.

(B**) Define the facets of the rules:** The sixth step is to define the new facets of the rules in our knowledge base. These facts can be different from about facts which describe in these features such as the value type, allowed values, the number of the values and other features of the values. The last step is to create the individual instances of classes in the hierarchy. Defining an individual instance of a class requires: Choose a class, Create an individual instance of that class, and Filling in the Fact values. The following figures [6.a],[6.b] shows the use of class diagram for patient treatment and Asset Assignment.

This UML class diagram provides the graphical representation of visualization, specifying, constructing and documenting the artifacts[19, 22]. The sample of the developed facts in our knowledge is shown in figure 7.


```
<DiagTestVal> <ConceptVal Name="Total Bilirubin /> <ValueVal Val="Normal / Increased /> <ValueVal Val="Increased />
</ConceptVal> <ConceptVal Name="Conjugated Bilirubin /> <ValueVal Val="Increased /> <ValueVal Val="Normal />
</ConceptVal> <ConceptVal Name="Unconjugated Bilirubin /> <ValueVal Val="Increased />
<ValueVal Val="Normal / Increased /> <ValueVal Val="Normal /> </ConceptVal> </DiagTestVal>
>DiagConcept< >ResultConcept Name="Prehepatic" NoTrueFinding="1<" >TestConcept Cpt="Total Bilirubin"
Val="Normal / Increased</" >TestConcept Cpt="Conjugated Bilirubin" Val="Increased</"
>TestConcept Cpt="Unconjugated Bilirubin" Val="Increased</" >TestConcept Cpt="Urobilinogen" Val="Increased</"
>TestConcept Cpt="Urine Color" Val="Normal (urobilinogen</"( >TestConcept Cpt="Stool Color" Val="Normal</"
>TestConcept Cpt="Alkaline Phosphatase Levels" Val="Normal</" >TestConcept Cpt="Alkaline Transferase
and Aspartate Transferase Levels" Val="Normal</" >TestConcept Cpt="Conjugated Bilirubin in Urine
" Val="Not Present</" />ResultConcept< >ResultConcept Name="Hepatic" NoTrueFinding="4<
>TestConcept Cpt="Total Bilirubin" Val="Increased</" >TestConcept Cpt="Conjugated Bilirubin" Val="Normal</"
>TestConcept Cpt="Unconjugated Bilirubin" Val="Normal / Increased</" >TestConcept Cpt="Urobilinogen
" Val="Normal / Increased</" >TestConcept Cpt="Urine Color" Val="Dark (urobilinogen+conjugated bilirubin</"
( >TestConcept Cpt="Stool Color" Val="Normal</" >TestConcept Cpt="Alkaline Phosphatase Level
s" Val="Increased</" >TestConcept Cpt="Alkaline Transferase and Aspartate Transferase Levels" Val="Increased</" >
TestConcept Cpt="Conjugated Bilirubin in Urine" Val="Present</" />ResultConcept<
```


Figure 7. Sample of developed Rules in our knowledge.

In our domain knowledge [1] the facts is represented as shown in figure 7 where the concept "Total Bilirubin" is a one of the diagnostic test for "Jaundice" and the concept has possible values are " Normal / Increased " and " Increased ". The property of each concept here is default as "Value". The knowledge can be formulated as shown in the following simple statements: IF the ‘traffic light’ is green THEN the action is go, as for example: IF the ‘traffic light’ is red THEN the action is stop. These statements represented in the IF-THEN form are called production rules or just rules. The term ‘rule’ in artificial intelligence, which is the most commonly type of knowledge representation, can be defined as IF-THEN structure that relates given information or facts in the IF part to some action in the THEN part. A rule provides some description of how to solve a problem. Rules are relatively easy to create and understand. Any rule consists of two parts: the IF part, called the antecedent (premise or condition) and the THEN part called the consequent (conclusion or action). The basic syntax of a rule is: IF <antecedent> THEN <consequent> . The rules in XML format have a different structure with the previous meaning but in different format. Sample of rule built in the proposed HCS system is shown below; it can be interpreted as following:

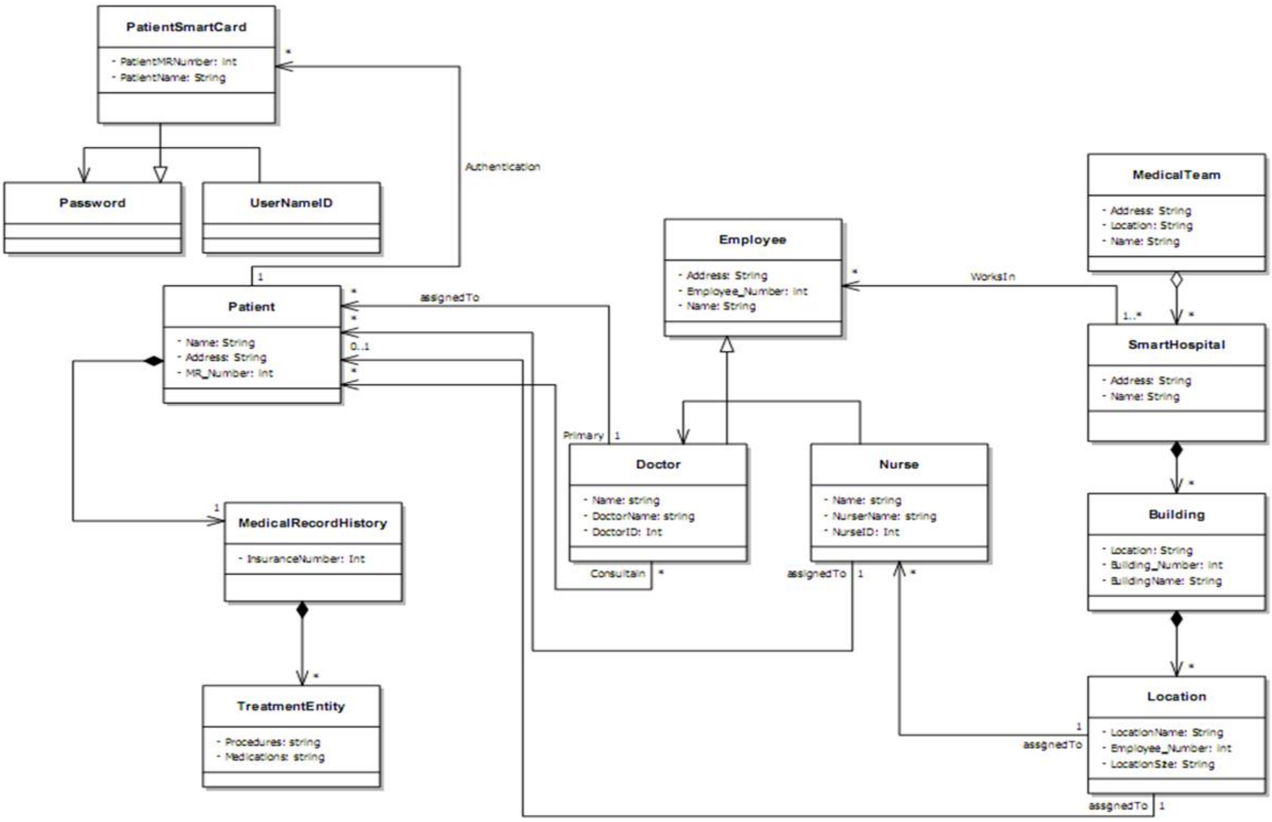


Figure 6(a). Class diagram for patient Treatment.

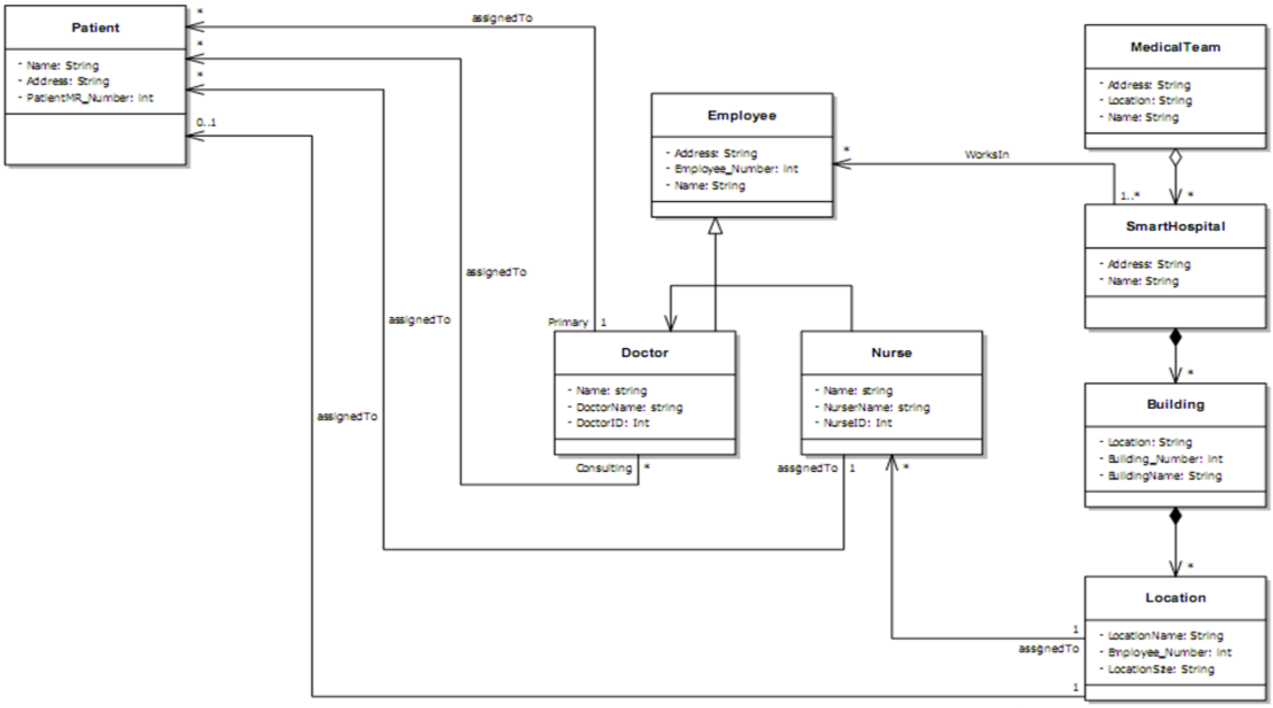


Figure 6(b). Class diagram for Asset Assignment.

<DiagConcept>; represents the root in the domain of the Jaundice. The node of "ResultConcept" represents a rule consequent and has attribute "Name" its value takes the consequent as " Prehepatic ". The child nodes "TestConcept" represent the decision rule for each part in Jaundice diagnosis that has two attributes are " Cpt ", and "Val". For example the antecedent of rule is " Total Bilirubin = Normal / Increased ".
The attribute "NoTrueFinding" represents the number of rule antecedent selected.

7. Implementation Model of Smart Hospital

It is a useful system for any hospital; especially, Smart hospital concerning with all aspects and issues related to a hospital including patients, doctors, employees, treatment and departments. By using internet, doctors can access the system from any place around the world as shown in an assembled pictures below. Figure 8 and 9 illustrates the collection steps for Implementation of our smart Hospital model by using medical smart card to develop the ontology system in modern hospital (smart hospital). Our smart hospital provides access to its system by using a smart card through three levels to maintain:

•.Write.

•.Modify.

•.Full control .

• Smart Card

For logging into our program, you need to have a smart card. A personal smart card will be generate by administrator, and given to the user. Each user has some permissions in logging into the program, some users can read only, some can write only, and others can do anything (full control). After you taking your personal card, you can insert it in your pc or laptop, and then loggings into the program by enter your code.

8. Conclusion

Today's hospitals are looking at strategic ways to use technology to manage their systems called smart hospital. The concept of the smart hospital is about adding intelligence(smart) to the traditional hospital system by covering all resources and locations with patient information. Patient's information is an important component of the patient privacy in any health care system that is based on the overall quality of each patient in the health care system.

Figure 8. Implementation of our smart Hospital model.

The main commitment for any health care system is to improve the quality of the patient and privacy of patient's information. For this, it is needed to develop the ubiquitous health care computing environment using the XML technology with traditional hospital environment. Our goal is achieved by combining elements of

facility design, process engineering, customer services programs, increasing patients safety and deployment of clinical and information technologies and internet to give access the system from any place around the world.

This paper is based on the UML and XML technology to design and development of the ontology system, can be solved these

Figure 6(b). Class diagram for Asset Assignment.

problems has been tried to improve traditional hospital system into smart hospital. The key solution of the smart hospital is online identification of all patients, doctors, nurses, staff, Medical equipments, medications, blood bags, surgical tools, blankets, sheets, hospital rooms, etc. In this paper, Efforts are to improve the knowledge-base ontological description for smart hospital system by using UML and XML technology, Our knowledge is represented in XML format from UML modeling(class diagram). Finally, we are implementing our model in smart hospital by using a medical smart card to improve the performance of health care system.

References

[1] Sanjay Anand and Akshat Verma " Development of Ontology for Smart Hospital and Implementation using UML and RDF, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, p.p. 206-212, September 2010.

[2] Khaled M Fouad, Magdy Shayboub Ali Mahmoud and Yasser N. "Agent of Healthcare system based on XML Technology "International Journal of Computer Information Systems(IJCIS), Vol. 2, No. 1, pp.35-46, January 2011.

[3] Magdy Shayboub Ali Mahmoud "Improve of Health Care Systems for Smart Hospitals Based on UML and XML", International Journal of Computer and Information Technology(IJCIT), Volume 02– Issue 03, pp. 484-491, May 2013.

[4] Bardram JE. and Christensen HB. "Pervasive Computing Support for Hospitals: An overview of the Activity-Based Computing Project," IEEE Pervasive Computing, Vol. 16, No. 1, pp. 44-51, 2007.

[5] Berners-Lee, Tim, James H, & Ora L. "The Semantic Web," Scientific American, May 2001.

[6] Horrocks I; Patel-Schneider PF.; and van Harmelen F. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002).

[7] Grainger M; Lee J "Ontology – Applications and Design. Communications of the ACM", Vol. 45, No. 2, February 2002.

[8] Weiser M. "Ubiquitous Computing". IEEE Computer, 26(10):71-72, 1993.

[9] Erdmann M and Studer R. "Ontology as conceptual models for XML documents". In Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99). Knowledge Science Institute, University of Calgary, 1999. http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Erdmann1/erdmann.pdf.

[10] Dean M; Schreiber G; Bechhofer S; van Harmelen F; Hendler J; Horrocks I; McGuinness DL; Patel-Schneider PF; and Stein L A (2004). OWL web ontology language reference. W3C Recommendation. http://www.w3.org/TR/owl-ref/

[11] Alani H, Kim S, Millard DE, Weal MJ, Hall W, Lewis PH and Shadbolt NR. "Automatic Ontology-Based Knowledge Extraction fromWeb Documents",IEEE Intelligent Systems,Vol. 10,No1, pp.14-2, 2003..

[12] Guided Tour of Ontology; John F. Sowa, http://www.jfsowa.com/ontology/guided.htm

[13] Gandon F. Ontology Engineering: a survey and a return on experience, Research Report of INRIA n°4396, France, March 2002.

[14] AIWatch-TheNewsletter of Artificial Intelligence.Vol. 4-6.AI Intelligence, Oxford,UK.

[15] Uschold M and Gruninger M. Ontologies: Principles, methods and applications. Knowledge Engineering Review, Vol. 11:2, 93-136, 1996. Also available as AIAI-TR-191 from AIAI, the University of Edinburgh.

[16] Bruno B. "Modélisation linguistique et modélisation logique des ontologies: l'apport de l'ontologie formelle." In Proceedings of IC 2001, pp 349-368 Plate-forme AFIA, Grenoble 25-28 juin 2001.

[17] Press Release -The Smart Hospital at The University of Texas at Arlington School of Nursing becomes a Laerdal Center of Excellence in Simulation. Available at-http://www.uta.edu/nursing/simulation/pr-laerdal.pdf

[18] Victor Foo Siang Fook, Siew Choo Tay, Maniyeri Jayachandran, Jit Biswas, Daqing Zhang, "An Ontology-based Context Model in Monitoring and Handling Agitation Behaviour for Persons with Dementia," percomw, pp.560-564, Fourth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006

[19] Bardram J. Applications of context-aware computing in hospital work – examples and design principles, Proceedings of the ACM Symposium on Applied Computing. 2004.

[20] Favela et al., "Activity Recognition for Context-Aware Hospital Applications: Issues and Opportunities for the Deployment of Pervasive Networks," Mobile Networks and Applications, vol. 12, nos. 2–3, pp. 155–171, 2007.

[21] M.D. Rodríguez, J. Favela, A. Preciado y A. Vizcaino,. "Agent-Based Ambient Intelligence for Healthcare," AI Comm., vol. 18, no. 3, pp. 201– 216, 2005.

[22] E. B. Moran et al., "Mobility in Hospital Work: Towards a Pervasive Computing Hospital Environment," Int'l J. Electronic Healthcare, vol. 3, no. 1, pp. 72–89, 2006.

[23] V Stanford "Beam Me Up, Doctor McCoy," IEEE Pervasive Computing, vol. 2, no. 3, pp. 13–18, 2003.

[24] D Sánchez, M Tentori and J Favela "Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments," Proc. 8th Mexican Int'l Conf. Current Trends in Computer Science (ENC 07), IEEE CS Press, pp. 33–40, 2007.

[25] Manhattan Research inc., "Physicians in 2012: The Outlook for On Demand, Mobile, and Social Digital Media". A Physician Research Module Report, 2009, New York, USA.

[26] Dairazalia Sánchez, Monica Tentori, and Jesús Favela, Cicese "Activity Recognition for the Smart Hospital", IEEE INTELLIGENT SYSTEMS, April 8, 2009. Pp. 50-57.

[27] Changrui Xiao and Arthur Yu "Medical Smart Card System for Patient Record Management", Bears Breaking Boundaries Science, Technology, and Energy Policy White Paper Competition, 2009, pp.1-12.

International Journal of
Intelligent Systems and
Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

A Fuzzy Logic Controller with Tuning Output Scaling Factor for Induction Motor Control Taking Core Loss into Account

Mohammad Abdul Mannan[1*], Toshiaki Murata[2], Junji Tamura[2]

Received 31th December 2013, Accepted 22th March 2014

Abstract: This paper presents a design of a fuzzy logic controller (FLC) with tuning output scaling factor for speed control of indirect field oriented induction motor (IM) taking core loss into account. The variation of output scaling factor of FLC depends on the normalized output of FLC. Firstly the speed control of IM taking core loss into account is presented by using FLC with fixed scaling factors (FLC-FSF). Secondly the speed controller based on suggested FLC with tuning output scaling factor (FLC-TOSF) is proposed. The performance of the proposed FLC-TOSF for speed control of IM are investigated and compared to those obtained using FLC-FSF at different operating conditions and variation of parameters. A comparison of simulation results shows that the convergence of actual speed to reference speed is faster by using the proposed FLC-TOSF.

Keywords: *Core loss, field oriented induction motor, fuzzy logic controller, fuzzy logic controller with tuning output scaling factor*.

1. Introduction

The field oriented control (FOC) [1] of induction motor (IM) used in industrial and process applications for high dynamics performances based on conventional PI controller [2]. Fuzzy logic control (FLC) is used in order to overcome the problems of conventional PI controller based FOC of IM [3 - 6]. But the FOC has been developed by neglecting core loss [1-6]. The decoupling control of rotor flux and torque has been implemented by decomposing the stator currents components [7]. The effects of core loss in AC drive have been investigated and it has been clarified that some compensation is required for FOC to overcome the perturbations of core loss [8]. The flux controlling stator current components and electromagnetic torque control stator current components are interfered with each other due to the effects of core loss. The effects of core loss in IM have been investigated and an indirect FOC (IFOC) in terms of magnetizing current components has been proposed in [9]. But the proposed IFOC is valid only for steady state condition.

Based on the proposed IFOC in [9], the conventional PI controller has been proposed in [10] for both transient and steady state operations of an IM taking core loss into account. The gains of PI controller have been changed when the load torque changed [10] by which it has proven that the proposed PI controller is not robust under the variation of load torque. Consequently, fuzzy logic based high performance control of IM taking core loss into account has been proposed in [11]. It is assumed that the FLC is a best controller in terms of dynamics response and best disturbance rejection [12, 13].

The FLC has been designed where the scaling factors, membership function of the linguistic variable and the rules are kept constant. But these parameters can be altered on-line in order to improve and to modify the controller performance [14, 15].

In this paper, we suggested a novel design of a FLC with tuning the output scaling factor (FLC-TOSF) by the normalized output of FLC-FSF for speed control of IM taking core loss into account. The improvements and the effectiveness of the proposed FLC-TOSF are verified by using simulation. The faster convergence of actual speed to the reference speed is achieved around the operating rated conditions and under the variations of load torque and parameters of IM. The special merits of this controller is that it does not require the knowledge of a mathematical model of the plant for the variation of output scaling factor and it can be easily implemented.

2. MODEL OF IM TAKING CORE LOSS INTO ACCOUNT

The equivalent circuit of an IM taking core loss into account in a synchronously rotating reference frame is shown in **Fig. 1** [9]. The voltage and flux linkage can be expressed by the following equations.

$$\boldsymbol{v}_1 = R_1\boldsymbol{i}_1 + d\boldsymbol{\Phi}_1/dt + j\omega_e\boldsymbol{\Phi}_1 \quad (1)$$

$$0 = R_2\boldsymbol{i}_2 + d\boldsymbol{\Phi}_2/dt + j\omega_s\boldsymbol{\Phi}_2 \quad (2)$$

$$R_c\boldsymbol{i}_c = d\boldsymbol{\Phi}_m/dt + j\omega_e\boldsymbol{\Phi}_m \quad (3)$$

$$\boldsymbol{\Phi}_1 = L_1\boldsymbol{i}_1 + \boldsymbol{\Phi}_m;\ \boldsymbol{\Phi}_2 = L_2\boldsymbol{i}_2 + \boldsymbol{\Phi}_m;\ \boldsymbol{\Phi}_m = L_m\boldsymbol{i}_m \quad (4)$$

The mechanical modelling part of IM is given by

$$d\omega_m/dt = -(D/J)\,\omega_m + (P_n/J)(T_e - T_L) \quad (5)$$

[1] *Dept. of EEE, American International University – Bangladesh, 83/B Road 14 Kemal Ataturk Avenue, Banani, Dhaka*
[2] *Dept. of EEE, Kitami Institute of Technology,165Koen-cho, Kitami, Hokkaido, 090-8507 Japan*
** Corresponding Author: Email: mdmannan@aiub.edu*

with

$$T_e = P_n(L_m/L_2)(i_{mq}\Phi_{2d} - i_{md}\Phi_{2q}) \quad (6)$$

where, symbols R_1, R_2, R_m indicate stator, rotor and core loss resistances. L_1, L_2, L_m indicate stator and rotor leakage and mutual inductances. P_n indicates number of pole pair. ω_e, ω_m, ω_s indicate synchronous, rotor and slip speeds. $\boldsymbol{v}_1$indicates stator voltage vector. $\boldsymbol{i}_1$, $\boldsymbol{i}_2$,$\boldsymbol{i}_m$, $\boldsymbol{i}_i$ indicate stator, rotor, magnetizing and core loss current vectors. $\boldsymbol{\Phi}_1$, $\boldsymbol{\Phi}_2$, $\boldsymbol{\Phi}_g$ indicate stator, rotor and air gap flux vectors. T_e, T_L indicate electrical and load torques. J, D indicate total inertia and damping factor. Suffix d and q indicate direct axis and quadrature axis components.

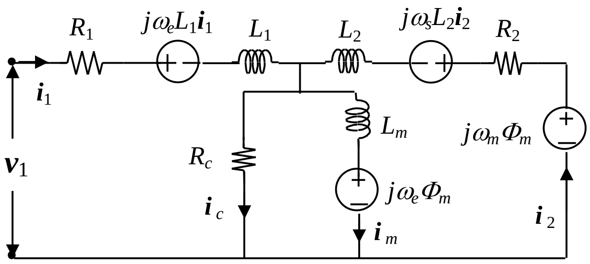


Fig. 1. Equivalent circuit of IM taking core loss into account.

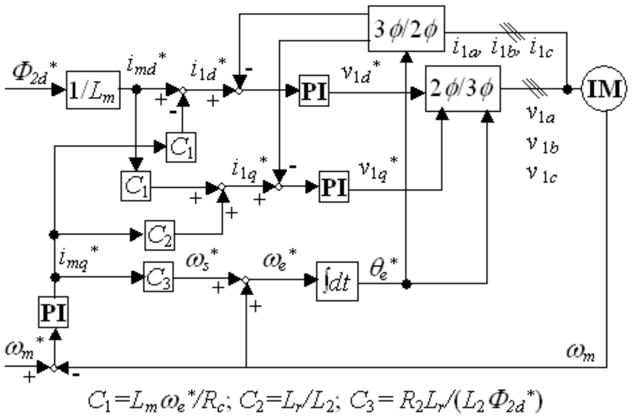


Fig. 2. Control structure of IM taking core loss into account based on PI controller

According to the IFOC theory [9], the rotor flux and electromagnetic torque are decomposed into i_{md} and i_{mq} components. Consequently, the constraint of IFOC can be written by the following expression.

$$\Phi_{2d} = \text{Constant}; \quad \Phi_{2q} = 0 \quad (7)$$

Hence, the rotor flux and electromagnetic torque are decoupled from each other, and can be separately controlled as desired. Then the behaviour of IM can be described by the following simplified model equations [9]:

$$\Phi_{2d} = L_m i_{md} \quad (8)$$

$$T_e = Pn(L_m/L_2) i_{mq}\Phi_{2d} \quad (9)$$

$$\omega_s = (R_2 L_m/L_2)(i_{mq}/\Phi_{2d}) \quad (10)$$

According to the theory of IFOC, the whole control system structure based on PI controller is shown in **Fig. 2** [10]. There are three PI controllers are used one for speed control and the others for stator currents control. Due to the drawbacks of conventional PI controller, a FLC was proposed for replacing PI speed controller.

3. DESIGN OF A FLC WITH TUNING OUTPUT SCALING FACTOR FOR IM SPEED CONTROL

The discrete time form of PI speed controller (see Fig. 2) can be written by the following equation.

$$\Delta i_{mq}^{*}(k) = K_p \Delta e(k) + K_i T_s e(k) \quad (11)$$

where, $e(k) = \omega_m^{*}(k) - \omega_m(k)$;

$\Delta e(k) = e(k+1) - e(k)$;

$\Delta i_{mq}^{*}(k) = i_{mq}^{*}(k+1) - i_{mq}^{*}(k)$, superscript "*"indicates the desired or reference value. K_p and K_i indicate proportional and integral constant. k and T_s indicate sampling instant and sampling time.

Control structure of a standard FLC is shown inn **Fig. 3**. The structure of a standard FLC can be seen as a traditional PI controller, so the design of FLC is explained in the following:

3.1. Fuzzification

Like a PI controller, the speed error $e(k)$ and its variation $\Delta e(k)$ are considered as input linguistic variables and the change of magnetizing q-axis current $\Delta i_{mq}^{*}(k)$ is considered as the output linguistic variable. For convenience, the inputs and output of the FLC are scaled with three different coefficients K_e, $K_{\Delta e}$ and K_i (see Fig. 3). These scaling factors play an important role for FLC in order to achieve a good behaviour in both transient and steady state conditions. These scaling factors can be constants or variable.

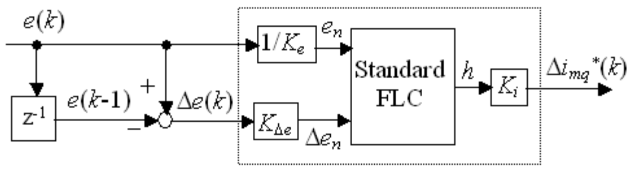


Fig. 3 Control structure of a standard FLC.

In this work, seven membership functions with overlap, of triangular shape are used for each input and output variable. The membership functions of linguistic variable are shown in **Fig. 4**. The linguistic variables are represented by NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium) and PB (Positive Big). The grade of input membership functions can be obtained as follows.

$$\mu(x) = [w - 2|x - m|]/w \quad (12)$$

where, $\mu(x)$ is the value of grade of membership, w is the width and m is the coordinate of the point at which the grade of membership is 1, x is the value of the input variable.

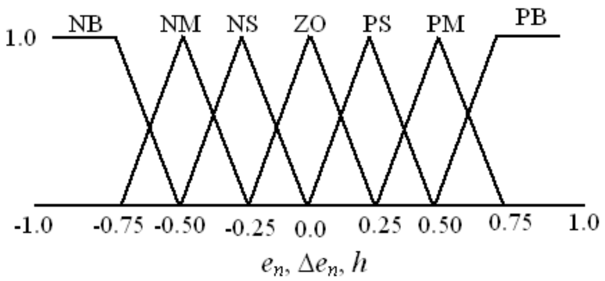


Fig. 4 Membership functions of input and output of FLC.

3.2. Rule Base

By using the scaling factors of input variables are normalized within the range –1 to 1 and then the range of output variable of FLC is also within –1 to 1. Here, we defined e_n,Δe_n and h as normalized speed error, change of speed error and change of magnetizing q-axis current (see Fig 3). The fuzzy mapping of the input variables to the output is represented by fuzzy IF-THEN rules of the following form:

IF < e_n is ZO> and <Δe_n is NB> THEN <h is NM>

IF < e_n is NB> and <Δe_n is PB> THEN <h is ZO>

Since the number of membership function for each input is seven, there are total 49 rules to achieve desired speed trajectory. The entire rule base is given in **Table 1**.

Table 1. Fuzzy rule table

Δi_{mq}^n		Δe_ω^n						
		NB	NM	NS	ZO	PS	PM	PB
e_ω^n	**NB**	NB	NB	NM	NM	NS	NS	ZO
	NM	NB	NM	NM	NS	NS	ZO	PS
	NS	NM	NM	NS	NS	ZO	PS	PS
	ZO	NM	NS	NS	ZO	PS	PS	PM
	PS	NS	NS	ZO	PS	PS	PM	PM
	PM	NS	ZO	PS	PS	PM	PM	PB
	PB	ZO	PS	PS	PM	PM	PB	PB

3.3. Inference and Defuzzification

From the rule base in Table 1, the inference engine provides fuzzy value of h, and then crisp numerical value of Δi_{mq}^* is obtained by using defuzzification procedure. The most popular method on inference and defuzzification is Mamdani's max-min (or sum-product) composition with centre of gravity method. In this work, we used the Mamdani type fuzzy inference and defuzzification method. The centre of gravity method [15] is used for defuzzification to obtain h. The normalized output function is given as

$$h = \sum_{i=1}^{N} \mu_i C_i / \sum_{i=1}^{N} \mu_i \tag{13}$$

where, N is total number rules, μ_i is the membership grade for ith rule and C_i is the coordinate corresponding to the maximum value of the respective consequent membership function [$C_i \in$ {-0.75, -0.5, -0.25, 0.0, 0.25, 0.5, 0.75}]. After finding out h, the actual desired first difference magnetizing current, Δi_{mq}^*, can be found out by product of scaling factor K_i as shown in Fig. 3.

3.4. Proposed Output Scaling Factor Variation

In sub- sections 3.1 to 3.3 described to design a FLC with fixed scaling factors of input and output variables. The FLC-FSF provides good performances at rating operating conditions and for load rejection. We proposed FLC-TOSF where the output factor is tuned in on-line to improve the performance of FLC-FSF in terms of the convergence of actual speed to the reference speed. In the proposed control system, the tuning of output factor does not depend on any mathematical model. The tuning of output factor is changed according to the normalized output of FLC-FSF. Let K_{it} is the modified scaling factor and its can be expressed by the following equation.

$$K_{it} = K_i + (1-h) \tag{14}$$

According to equation (14), the scaling factor K_{it} is used instead of K_i in Fig. 3.

4. SIMULATION RESULTS

In order to verify the performance of the proposed FLC-TOSF for speed control of IM drive, simulations were carried out. The ratings and parameters of the IM model are listed in **Table 2**.

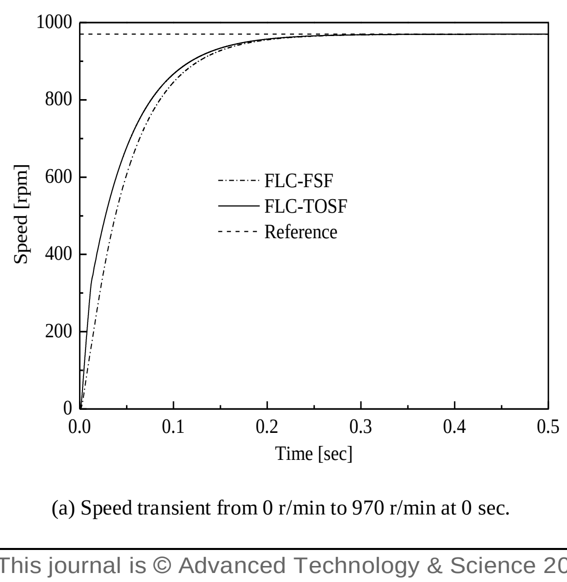
The value of sampling period is chosen 75 μsec. For FLC-FSF, the scaling factors are chosen as K_e = 304.74, $K_{\Delta e}$ = 2.1 rad/sec and K_i = 0.0162. For PI current control, the PI constants are chosen as K_p = 3.0 and K_i = 1500.0. The mentioned scaling factors and PI gains are selected by trial and error method to achieve as good as performance of FLC-FSF and PI controller.

Table 2. Ratings and Parameters of IM

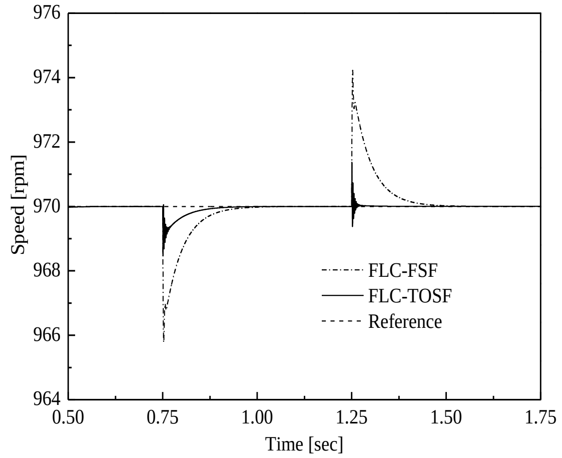
Ratings	**Parameter values**
1.1 kW, 200/√3 voltage/phase, slip=0.03, 6 pole, 50 Hz	R_1 =0.2842 Ω, R_2 =0.2878 Ω, R_c=329.667Ω L_s = 28.3 mH, L_r = 28.8 mH, L_m = 26.8 mH, J = 0.0179 Kg-m^2, D = 0.0

Fig. 5 shows the transient and steady state responses of speed regulation of IM for various operating conditions. Fig. 5 (a) and (b) show the transient responses of speed where speed is changed from 0 r/min to 970 r/min (rated value) at t = 0 sec and the load torque is changed 50% to 100% of its rated value at t = 0.75 sec. The 50% load torque is reduced again at t = 1.25 sec. It is comprehended from Fig. 5 (a) that the convergence of actual speed to the reference speed is faster by using the proposed FLC-TOSF. Fig. 5(b) shows that the transient performance of FLC-FSF is improved by using the proposed FLC-TOSF. Fig. 5 (c) also shows that the transient response of speed regulation can be achieved by using the proposed FLC-TOSF.

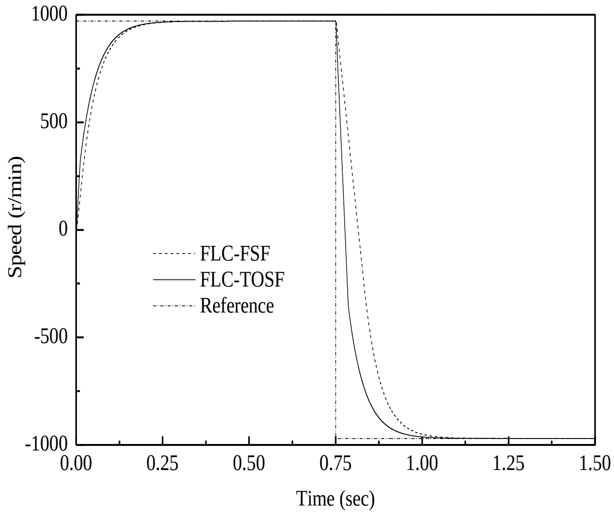
In order to show the robustness of FLC-FSF and FLC-TOSF against parameters variation (R_2 and J), simulation results, are presented as shown in Figs. 6 and 7. It can be seen that non-overshoot speed response is obtained against the large deviations of rotor resistance and inertia by using the FLC-FSF or FLC-TOSF. The transient response of FLC-FSF is also improved by using the proposed FLC-TOSF.



(a) Speed transient from 0 r/min to 970 r/min at 0 sec.

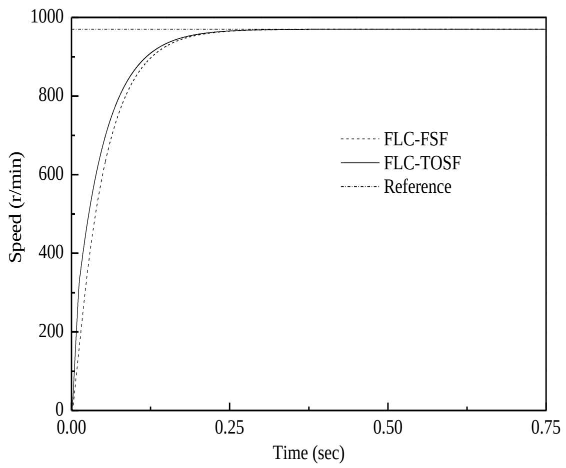


(b) Speed transient response by applying 100% rated torque at 0.75sec and 50% rated torque at 1.25 sec.

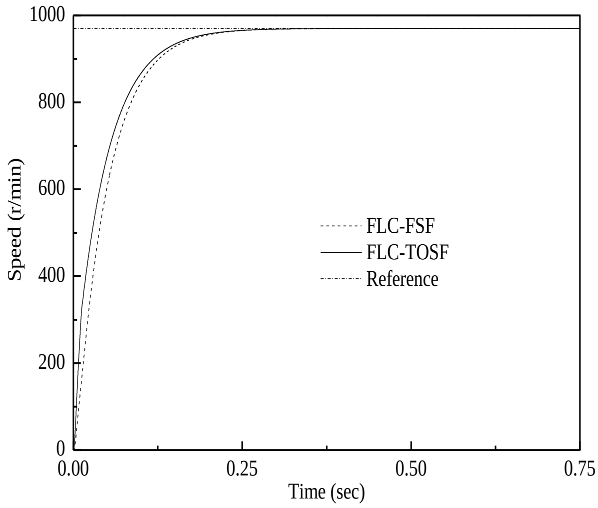


(c) Speed transient from 0 r/min to 970 r/min followed by speed reversion from 970 r/min to –970 r/min at 0.75 sec.

Fig. 5. Simulation results of the speed regulation of IM for both FLC-FSF and FLC-TOSF.

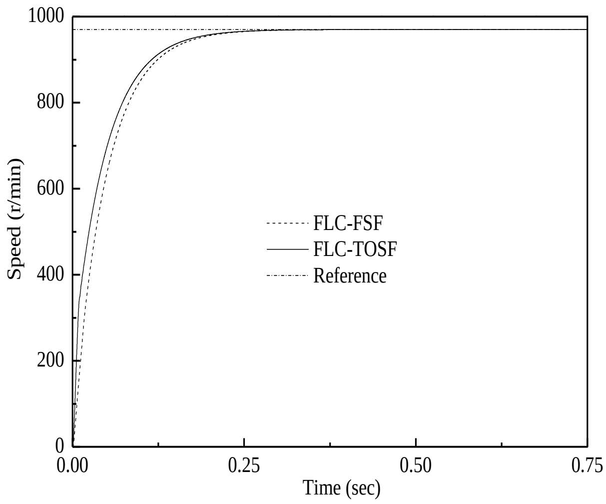


(a) ΔR_2 = +50% of rated R_2

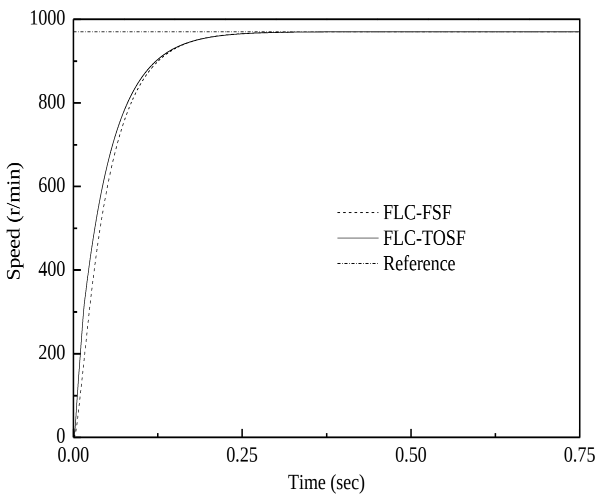


(b) ΔR_2 = -50% of rated R_2

Fig. 6. Simulation results of the speed regulation of IM considering R_2 variation.



(a) ΔJ = -50% of rated J



(b) ΔJ = +100% of rated J

Fig. 7. Simulation results of the speed regulation of IM considering J variation.

5. CONCLUSION

A novel design of a fuzzy logic controller with on-line tuning output scaling factor for the speed control of IM taking core loss into account is fully explained. Moreover, the achievement of the proposed controller for various operating conditions and parameters variation was investigated by simulation study. It has been demonstrate how a simple the transient response can be improved by tuning output scaling factor. It can be concluded that the convergence of actual speed to the reference speed of a FLC with fixed scaling factor can be improved by tuning output scaling factor.

References

[1] P. Vas, Vector Control of AC Machines, Clarendon Press Oxford, U. K., 1990.

[2] Dong-Choon Lee, et al.: "High performance Current Regulator for Field-Oriented Controlled Induction Motor Drive," IEEE Trans. on Industry applications, Vol. 30, pp. 1247–1257, 1994.

[3] M. Nasir Uddin, Tawfik S. Radwan and M. Azizur Rahman, "Performances of Fuzzy-Logic-Based Indirect Vector Control for Induction Motor Drive," IEEE Ind. Applications, vol.38, No. 5, pp. 1219 - 1225, Sep./Oct. 2002.

[4] Y. Srinivasa Kishore Babu and G. Tulasi Ram Das, "Improvement indirect torque control of induction motor using fuzzy logic duty ratio controller", ARPN Journal of Engineering and Applied Sciences, Vol. 5, No.4, pp. 68-74, 2010.

[5] P. Tripura and Y. Srinivasa Kishore Babu, "Fuzzy Logic Speed Control of Three Phase Induction Motor Drive," World Academy of Science, Engineering and Technology, Vol. 60, pp. 1371 - 1375, 2011.

[6] B. Bhushan, M. Singh, and Prem Prakash, "Performance Analysis of Field Oriented Induction Motor using Fuzzy PI and Fuzzy Logic based Model Reference Adaptive Control," International Journal of Computer Applications, Vol. 17, No. 4, pp. 5 – 12, March 2011.

[7] Bimal K. Bose, "Modern Power Electronics and Ac Drives," Prentice Hall, 2001.

[8] I. Boldea and S.A. Nasar, "Unified Treatment of Core Losses and Saturation in the Orthogonal-axis Model of Electric Machines," IEE Proc., Vol.134, Pt. B, No.6, pp.355 - 363, 1987.

[9] Emil Levi, "Rotor Flux Oriented Control of Induction Machines Considering the Core Loss," Electrical Machines and Power Systems, Vol.24, No.1, pp. 37-50, 1996.

[10] M. A. Mannan, T. Murata, J. Tamura and T. Tsuchiya, "Indirect Field Oriented Control for High Performance Induction Motor Drives Using Space Vector Modulation with Consideration of Core Loss," in Proc.PESC'03, Acapulco, Mexico, pp. 1449–1454, 15-19 June, 2003.

[11] M. A. Mannan, T. Murata, J. Tamura and T. Tsuchiya, "Fuzzy-Logic-Based High-Performance Control of Induction Motor Taking Core Loss into Account," in Proc.ICEM'04, Cracow, Poland, Paper No. 256, 5 - 8 September, 2004.

[12] B N Kar; K B Mohanty; M Singh, Indirect vector control of induction motor using fuzzy logic controller, in Proc. ICEEE"11 (International conference on Environment and Electrical Engineering), pp. 1-4, 2011.

[13] D. A. Linkens and H. O. Nyongesa, "Genetic Algorithams for Fuzzy Control Part 2: Online System Development and Application," IEE Proc. Control Theory Application, Vol. 142, No. 3, pp. 177-185, 1995.

[14] Aldo Balestrino, Alberto Landi and Luca Sani, "CUK Converter Global Control Via Fuzzy Logic and Scaling Factors," IEEE Ind. Applications, vol.38, No. 2, pp. 406 - 413, Mar./Apr. 2002.

[15] D. Driankov, H. Hellendoorn and M. Reinfrank, An Introduction to Fuzzy Control, Springer-Verlag, 1993.

International Journal of
Intelligent Systems and
Applications in Engineering
IJISAE
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Application of Angle-Modulated Particle Swarm Optimization Technique in Power System Controlled Separation WAP

Almoataz Youssef Abdelaziz[1*], Walid El-Khattam, Mohammed

Received 29th December 2013, Accepted 28th January 2014

Abstract: One of the recommended preventive plans against the wide area disturbances is WAP, Wide Area Protection, through controlled system splitting or separation. In this paper, authors are proposing three simple algorithms that are intended to be operating online, analyze data from wide-area PMUs placed in different parts of the grid, process the system state and lines' status and issue disconnecting actions to certain lines in the grid to form islands with minimum imbalances of power between generation and loads. First a simple approach is introduced which performs an extensive search for proper splitting strategies. The authors then present modified approaches which could make the whole system act within much shorter times. All the presented algorithms are analyzed and comments are made on ways to enhance their performances. This will emphasize on how such WAP systems would be designed and developed and if necessary tailored to fit specific systems or applications.

Keywords: *Controlled Separation, Particle Swarm Optimization, System Splitting, Wide Area Protection.*

1. Nomenclature

AMPSO: Angle-Modulated Particle Swarm Optimization
PMU: Phasor Measurement Unit
PSO: Particle Swarm Optimization
WAP: Wide Area Protection

2. Introduction

Power systems today are made up of thousands and thousands of different components. They are widely and, in some cases, massively spread over vast lands and territories. Hence reliability and security of their operation is no longer an easy target. Challenges in maintaining healthy operation are magnified and elevated by the power system size. Despite huge advancements in different power system components, operations, and protection technologies, today's power systems are more vulnerable to blackouts than ever before. One of the recommended preventive plans against the wide area disturbances and the blackouts is Wide Area Protection. With the rapidly growing capabilities in computer and communication technologies, opportunities are now being available for the introduction of advanced wide-area protection and control systems which shows a great potential. Such systems would receive wide-span information, e.g. system-wide voltages, angles, active and reactive power flows, etc., and analyze them, indicating whether the system is on the urge of a transformation into an unstable state, and thus, issuing wide-span, coordinated actions that will save the system from proceeding to total collapse, or even, mitigate the wide-area disturbance effects upon the system.

System splitting, also known as controlled separation, is to split the interconnected transmission network, deliberately and on purpose, into islands of load with matching generation at proper splitting points by opening a selection of transmission lines and ties. After which, load shedding and sometimes generation rejection should follow in order for the load and generation to remain balanced within balance, keeping the majority of the system intact and hence avoid cascading instabilities or even partial or total blackouts. The study of previous blackouts and outages suggest that if proper system splitting strategies along with suitable load shedding and minimized generator rejection had been performed within short time, some blackouts could have been avoided and mitigated. In this Paper the authors are proposing three simple real-time algorithm that are intended to be operating online, analyze data from wide-area PMUs placed in different parts of the grid, process the system state and lines' status and issue disconnecting actions to certain lines in the grid to form islands with minimum imbalances of power between generation and loads, without violating thermal and overloading constraints of the ties left intact within each island. Performances of the three different algorithms and techniques, utilized for finding suitable solutions for the controlled separation problem, are explored and compared. A conclusion is drawn in light of this comparison and suggested ideas for further future work are presented.

3. The Controlled Separation Problem

System splitting, also known as controlled separation, is to split the interconnected transmission network, deliberately and on purpose, into islands of load with matching generation at proper splitting points by opening a selection of transmission lines and ties. After which, load shedding and sometimes generation rejection should follow in order for the load and generation to remain balanced within balance, keeping the majority of the system intact and hence avoid cascading instabilities or even partial or total blackouts. The study of previous blackouts and

[1] *Ain Shams University, Faculty of Engineering, Electrical Power & Machines Department, Egypt*
** Corresponding Author: Email: almoatazabdelaziz@hotmail.com*

outages suggest that if proper system splitting strategies along with suitable load shedding and minimized generator rejection had been performed within short time, some blackouts could have been avoided and mitigated [1-3]. Operating states of the power system have been defined before as [4]:

- Normal State.
- Alert State.
- Emergency State.
- In-Extremis.
- Restoration.

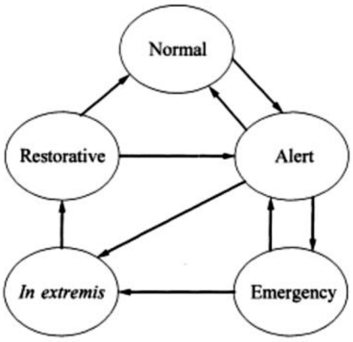


Fig. 1. Power System Operating states [4]

The system should be designed to act in the emergency state and help stop and save the situation from devastating. A practical controlled separation scheme needs to address the following three critical problems [5]:

- Where to separate? (i.e. separation points to form stable islands)
- When to separate? (i.e. separation timing)
- What to do after separation? (i.e. post islanding actions like load shedding and/or generator rejection)

When to act exactly or what triggers the system should be a threatening instability or signs derived from the input data received by the system, but it is not in this paper's scope. However, a lot of research is going on and shall be put into this "prediction of instabilities" scope. So is the case with the post-separation actions. The main concern in this paper with how to act and the separation techniques. Following the decision that separation is needed, the following describes the set of problems and/or constraints that the controlled separation has to take into consideration in order to derive proper and relatively stable islands. The most important thing is that those islands are proven to be stable and persistent after the separation in order to maintain most of the system's integrity, minimize the loads that experienced outages and ease the restoration process. Hence the solution that is found suitable should satisfy a set of constraints, formulated and discussed in numerous previous references [1], [6]:

3.1. Separation and synchronization constraint

Asynchronous groups of generators must be separated into different islands and generators in each island must be synchronous with each other. Following a serious contingency, some generators tend to swing together, while others tend to swing against those. Uncontrolled islanding will be inevitable, due to protection systems operating against and to prevent out-of-step oscillations [7, 8]. A suggested solution would be to determine the different groups of generators that swing together and start forming separate islands containing those coherent generators and disconnected from the other asynchronous groups' islands. Bottom-line: The first constraint would be identified as that buses of synchronized generators shall be connected in the same island and separate from the other generators that are swinging against them. Several previous researches [9], [10] have discussed this issue and in this paper. Here, it is assumed that the coherent generator groups are previously identified and set.

3.2. Power balance constraint

In order to ensure that the resulted islands remain intact and stable, each of the islands should have, theoretically, equal generated active power and load power consumption. Of course, this is impossible as some areas would have concentrated generation and others would have a majority of loads. Therefore, the constraint is revised to allow a small pre-set imbalance between generation and load of active power, or in other words power generation shall be roughly equal to power load. The imbalance will not be allowed to evolve into a major instability by further measures, either load shedding in case of load surplus or generator rejection in case of surplus in generation. Reactive power is not considered as constraint, as it assumed that the local measures are enough to solve this issue, i.e. capacitor banks, Static Var Compensators will compensate for additional reactive power demand and under-voltage load shedding would save the island from collapsing due to deficiency in reactive power supplies.

3.3. Rated Value & limit constraints

If the main aim is to maintain most of the system's integrity through maintaining the integrity and stability of the sub-islanded grids, then the transmission lines must not be loaded over their thermal and steady-state capacity limits. The grid degrades through cascading disconnections of the ties carrying its power flows, by protection systems, due to overloads and short circuits. To prevent that, it has to be ensured that overloading, if it occurs, is within the thermal limit for each tie.

4. Direct Searching Method

The following flowchart, in Fig. 2, shows simply the stages of the algorithm. The main and major benefit of this algorithm is its absolute simplicity, it could be implemented on any type of system and achieve results. However its performance is relatively more time consuming.

1. Initialization:

At first, the algorithm initializes the system parameters, i.e. the number of buses, number of lines, connectivity matrix that contains the lines and ties between all buses in the grid, groups of coherent generators, the injected powers of all buses meaning the difference between the generated and consumed powers, the allowed margin of power imbalance in each island, thermal limits and thermal overloading coefficients of all lines, impedances of branches.

2. Calculating the number of possible grid configurations:

The number of possible candidate solutions that the algorithm is intended to test and study equals to 2 to the power of number of the lines in the grid.

$$m = 2^{number\_of\_lines} \quad (1)$$

Where m = number of possible configurations of the grid.

3. Ensuring that no bus is left unconnected:

It should be noted that all load buses are included and connected to an island. This mitigates the blackout minimizing the amount of load loss over the whole grid.

4. Ensuring that no incoherent generator buses are connected:

This ceases the instabilities, resulting from the incoherent generators swinging against each other, from developing.

5. Ensuring that all coherent generator buses are connected to each other:

This will minimize the loss of generation and loss of supply to load by keeping most of the integrity of the generator buses.

6. Conduct Power imbalance calculation:

For each island the difference between the power generation and power load is calculated to make sure it is within the preset limit.

7. Conduct Approximate DC power load flow:

To make sure all lines left intact within each island will not be severely overloaded and hence result in further disconnection and degradation of the grid.

8. Add possible solution:

If candidate solution is verified to satisfy all the prior constraints then it will be added to the set of possible separation solutions.

The algorithm was engineered and programmed on Matlab and using a 1.73GHz dual core processor with installed 2 GB of RAM and running a 32-bit Windows 7 Operating system. So far, the algorithm was tested on 5, 9 and 14 bus standard systems. Here, only running the algorithm on the IEEE 14-bus standard test network is previewed. The following Table I shows the parameters, results and times:

Table I - Parameters, Results and Times for IEEE 14-bus system (total online solution)

Number of lines	20 lines
Number of candidate configurations	2^20=1048576
Time to explore all candidate configurations	7.5 minutes (450 seconds)
Time taken to find the first possible separation technique	39.3 seconds
Number of possible separation techniques found	256

The results show that the program, yet simple, but will take extensive time to find the possible solution.

This will not be appropriate to provide a suitable separation technique in short time to save the system.

This will not be appropriate to provide a suitable separation technique in short time to save the system. Doing part of the job as offline work will be the solution. The algorithm was further developed so that the stages 2-6 can be performed offline and their results will be ready prior to any instability, the system will have a much less number of candidate solutions to explore. For the purpose of this simulation, generation power at bus-2 is increased to 200MW.

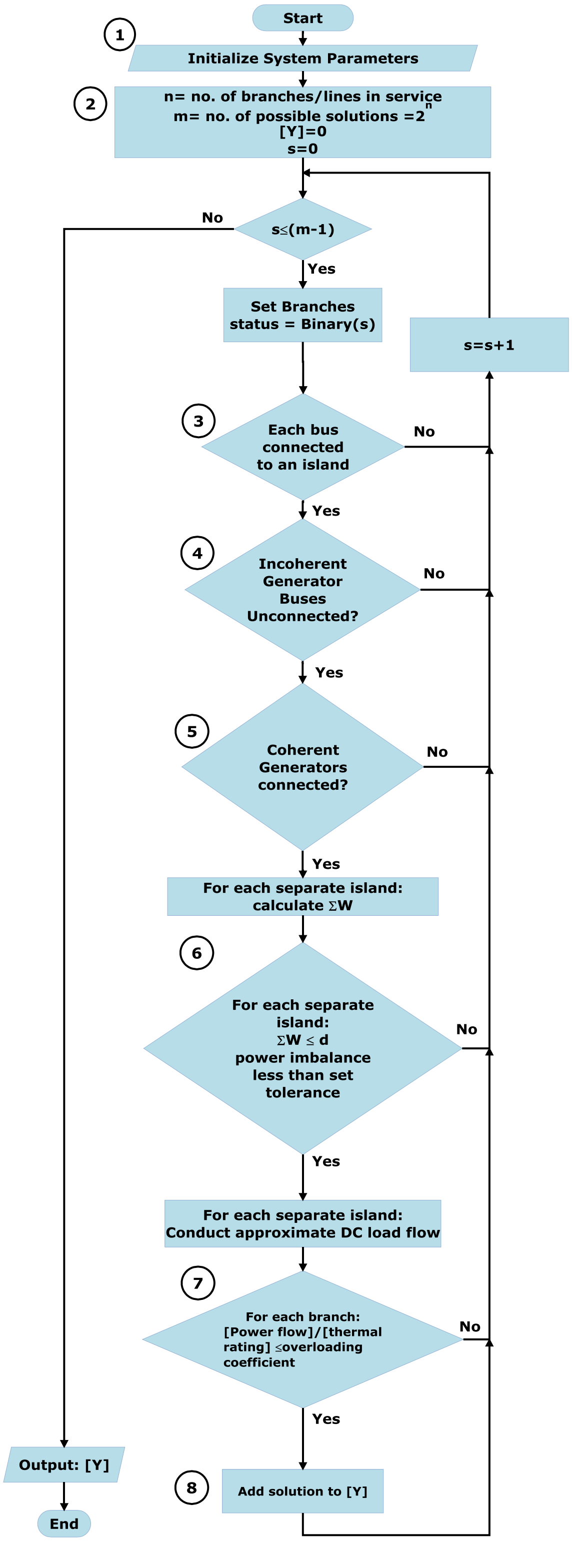


Fig. 2. Direct Searching Controlled Separation Algorithm Flowchart

Table II shows the Parameters, Results and times for the algorithm running on the same IEEE 14-bus standard system, but after doing most of the job, that is doable, offline.

Table II - Parameters, Results and Times for IEEE 14-bus system (partially online solution)

Number of candidate configurations (that already satisfy steps 3-6)	484
Time to explore all candidate configurations	0.5304 seconds
Time taken to find the first possible separation technique	0.2496 seconds
Number of possible separation techniques found	256

As shown the result of the system is exceptional and getting part of the algorithm processing to be done offline saved a lot of time for getting the final suitable separation solution. The time taken by the online processing is very suitable to be taken by a real-time application and will help save the system in appropriate short time.

5. Angle Modulated Particle Swarm Method

Search spaces of controlled separation problems are huge. In these cases, population based searches are a very valid proposed solution. Particle Swarm Optimization (PSO) is a population based algorithm first introduced by Dr. Russel C. Eberhart and Dr. James Kennedy in 1995 [11]. As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions [12].

Particle Swarm has two primary operators: Velocity update and Position update. During each generation each particle is accelerated toward the particles previous best position and the global best position. At each iteration, a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times or until a minimum error is achieved. The dimensionality of a problem influences the computational complexity in converging to a valid solution. Angle modulation is applied to reduce complexity of binary problems by generating a bit string to solve binary problems using PSO to evolve the function coefficients of a trigonometric model. Instead of evolving a high dimensional but vector, angle modulation reduces the problem to a four-dimensional problem defined in continuous space. Experimental results show that the angle modulation method is faster than the standard Binary PSO and that accuracy is improved for most benchmark functions used [13].

The Angle Modulated PSO (AMPSO) is a PSO algorithm that employs a trigonometric function as a bit string generator. The function is derived from a technique used in the field of signal processing from the telecommunications industry and is based on angle modulation. The standard PSO is then applied to optimize the simpler 4-dimensional tuple (a, b, c and d) instead of evolving the actual bit string. After the PSO iteration, the parameters are substituted back into equation. The resultant function is then sampled at the evenly spaced intervals to generate a bit for each interval, with the set of all generated bits representing the binary vector solution to the original problem. The benefit of the AMPSO is that a larger dimensional binary space can be represented by a smaller 4-dimensional continuous space. Results from the optimization are produced in a shorter period time, as only 4 parameters need to be optimized instead of the original n-dimensions. Previous literature [14, 15] have explored relying on the same technique for some part of the solution, however because of the random nature of this optimization technique the generated solution may result in some isolated buses. Thus the former researchers added post-processing stages that added to the time and resulted in solutions that were not purely the result of the AMPSO.

In this research, it is targeted to use only AMPSO and tried to decrease the time through some programming modifications. The algorithm here only explores solutions through solving only the power imbalance constraints. The algorithm for the AMPSO based controlled separation algorithm is as explained and shown:

5.1. Initialization:

At first, the algorithm initializes the system parameters, i.e. the number of buses, number of lines, connectivity matrix that contains the lines and ties between all buses in the grid, groups of coherent generators, the injected powers of all buses meaning the difference between the generated and consumed powers, the allowed margin of power imbalance in each island, the coefficients of the trigonometric angle modulation function and fitness function.

Processing:

In every iteration, for each particle:

i. Substitute current positions in the four dimensions into the trigonometric angle modulation function to get lines positions:

$$g(x) = \sin\big(2\pi \times (x-a) \times b \times \cos\big(2\pi \times c \times (x-a)\big)\big) + d$$

ii. For each line, to get its status:
If g(x)>0 then g(x)=1.
If g(x)<0 then g(x)=0.

iii. Evaluate solution to get the actual number of islands resulted, Coherency Index and the Fitness Function [11], [13].

a. Coherency Index is the sum of each island available generation divided by its pre-decided generation that was supposed to be grouped all as a percentage over number of islands.

b. Fitness Function [11], [12]:

$$F = \left(\sum_{i=1}^{n_{isl}} \sum_{j=1}^{m_j} L_{ij} \times P_{ij}\right) \times \left(1 + e^{\frac{(n_d - n_a)^2}{k}}\right) \times Coherency\ Index$$

c. If solution is best solution globally, then update a, b, c and d as best global positions found so far.

d. If solution is best solution in the neighbourhood, then update a, b, c and d as best neighbourhood positions found so far.

e. Update a, b, c and d new velocities and new positions according to PSO equations:

xid(t) = f (xid(t - 1), vid(t - 1), pid, pgd)

vid(t) = vid(t - 1) + C1 x ϕ1(pid - xid(t - 1)) + C2 x ϕ2(pgd - xid(t - 1))

xid(t) = xid(t - 1) + vid(t)

if vid > Vmax then vid = Vmax

else if vid < -Vmax then vid = -Vmax

5.2. Output:

Produce a record of all solutions and present top set of solutions

found during the iterations.
The algorithm was engineered and programmed on Matlab and using an Intel Core i7 2.1 GHz processor with installed 4 GB of RAM and running a 62-bit Windows 7 Operating system. So far, the algorithm was tested on 5, 9, 14 and 30 buses standard systems. Here, only running the algorithm on the 9-bus standard test network is previewed. The following Table III shows a sample of the results and times achieved by this algorithm:

Table III. Parameters, Results and Times for 9-bus system (total online solution)

Number of lines	9 lines
Number of candidate configurations	2^9=512
Time to complete all 15 iterations of 20 particles (300 steps)	3.12 seconds
Time taken to find the best separation technique	0.156 seconds

In comparison with previous literature, this algorithm has reached the best result during the second iteration, where every iteration is of 20 particles each. The best solution was found by the 16th particle in the second iteration. In previous literature [14, 15], the same best solution was claimed during the fifth iteration using the same population size; number of particles, which is 20. During a lot of simulations, results also showed that it reached the same best solution during the first iteration, by the 16th particle, however this was achieved by a population size of 30 particles. Despite that, it is not the best solution in respect to time taken as the time taken was 0.166 seconds. As shown, due to the random nature of the optimization problem, results may differ in each run. In this research, the same algorithm was run for the 14 bus and 30 bus IEEE systems but unfortunately did not perform successfully. Due to the random nature of the problem in binary space dimensions, this technique has only shown its big success in solving smaller grids. This still can be utilized if the bigger systems are clustered into smaller grids by compressing groups of buses or regions into one bus and considering the ties that only connect then to the other groups/regions as the only transmission lines studied.

6. New Online Method- "Sweeping and Keeping"

The Bioinformatics Toolbox™ product extends the MATLAB® environment to provide an integrated software environment for genome and proteome analysis [16]. In this research a Bioinformatics toolbar function, called graphtraverse, is used to analyze the relationship between the buses and present it as a path for the second phase to operate on as a guide for processing. The algorithm only concentrates on solving the controlled separation problem by trying to satisfy the power imbalance constraint in the islands.
Initializing the algorithm:
In this preliminary step the algorithm initializes the system data by constructing the system connectivity matrix, in addition to forming the matrix that contains generators that are planned to be joined together in each same island. It will also realize how many islands are required by setting.
Sweeping:
The algorithm here uses a pre-programmed function to traverse through the grid starting from each island's main generation bus. This function will travel through grid exploring and recording the buses that it traverses. It will use Breadth-First method which means that the order of buses will be according to closest from origin bus.

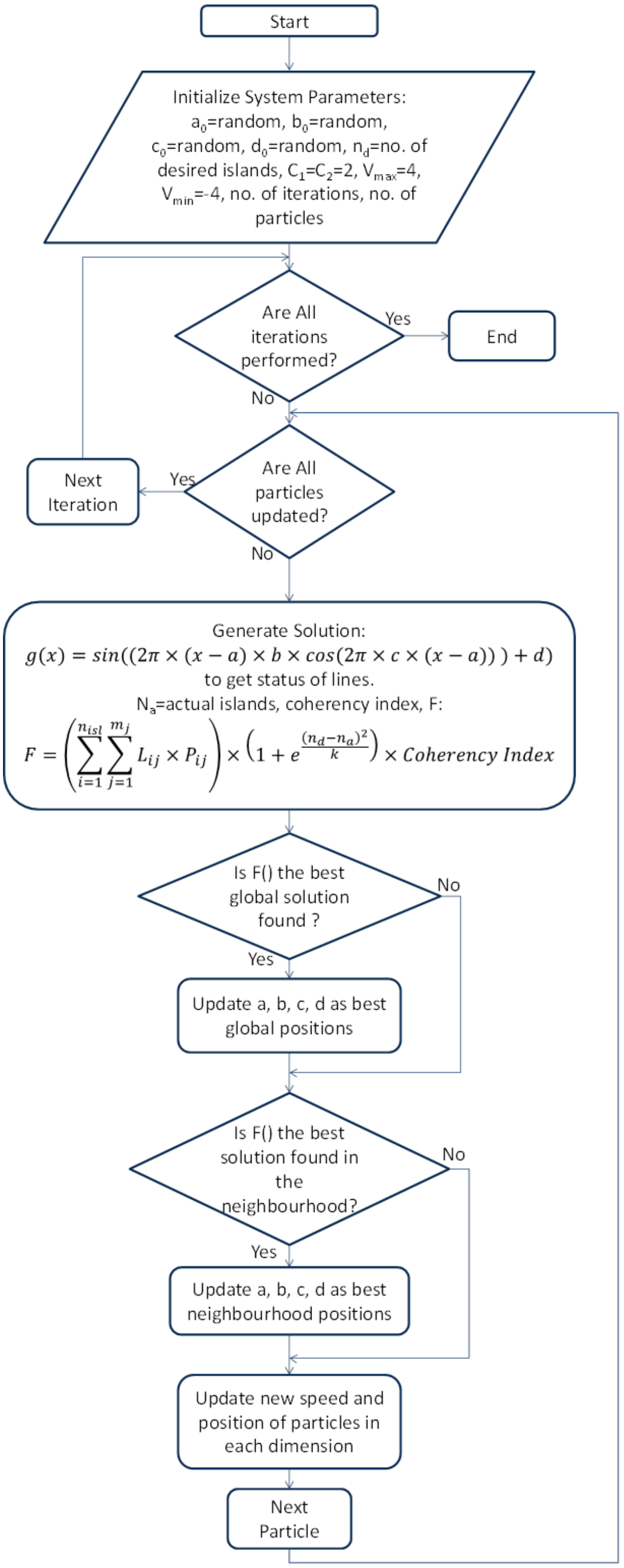


Fig. 3. Angle Modulated Particle Swarm Optimization-based Controlled Separation Algorithm Flowchart

This will create trails, according to number of required separated islands, for each agent of the second part of the algorithm to follow and decide whether each studied bus will be part of the same island or of another.
Keeping:
According to the number of islands, there will be agents or parts of the program loop, each concerned with its respective island and will follow the trail got from the prior part of the program, the sweeping, and will compare this bus power to the imbalance

of its island. If the bus power doesn't violate the power imbalance constraint, if added to this island, then it is added and connected to the island. All its transmission lines that are affiliated with the first island will be set to "ON" and hence will not be disconnected by the system. All its transmission lines that are affiliated with other islands will be set to "OFF" and hence will be disconnected by the system and tripping signals will be ordered to transmit to the line's involved circuit breakers.

In case the bus power will violate the power imbalance constraint the algorithm will skip it for the mean time and will proceed with the next bus in the sequence. This bus will then be left to other agents, of the other islands, to consider and study and if there is a possibility it will add it.

This part is looped until all buses are covered. If a bus or more remains unconnected then the algorithm will study which island could that bus be connected to, to explore all these options then decide the connection according to which island will be the best choice to minimize the imbalance.

The program terminates after all buses have been decided to which islands each is connected. The algorithm was engineered and programmed on Matlab and using an Intel Core i7 2.1 GHz processor with installed 4 GB of RAM and running a 62-bit Windows 7 Operating system. So far, the algorithm was tested on 9 and 30 buses standard systems.

1) Running the algorithm on the 9-bus standard test network is previewed. The following Table IV shows a sample of the results and times achieved by this algorithm for solving the 9-bus system:

Table IV - Parameters, Results and Times for 9-bus system (sweeping and keeping)

Number of lines	9 lines
Number of candidate configurations	2^9=512
Time taken to find the best separation technique	0.0312 seconds

As shown, for solving the 9-bus system, it took the algorithm 0.0312 seconds! This time is off course much shorter than all the previous literature results [13, 14]. It is even shorter in duration than the results achieved by the prior two techniques explained and presented in this research. By far this is the fastest and most successful result, known to the researchers, for the 9-bus system.

2) Running the algorithm on the IEEE 14-bus standard test network is previewed. It has to be noted that the system was modified for the purpose of this simulation test, generation of bus 2 was increased to 200MW. The following Table V shows a sample of the results and times achieved by this algorithm for solving the IEEE 14-bus system:

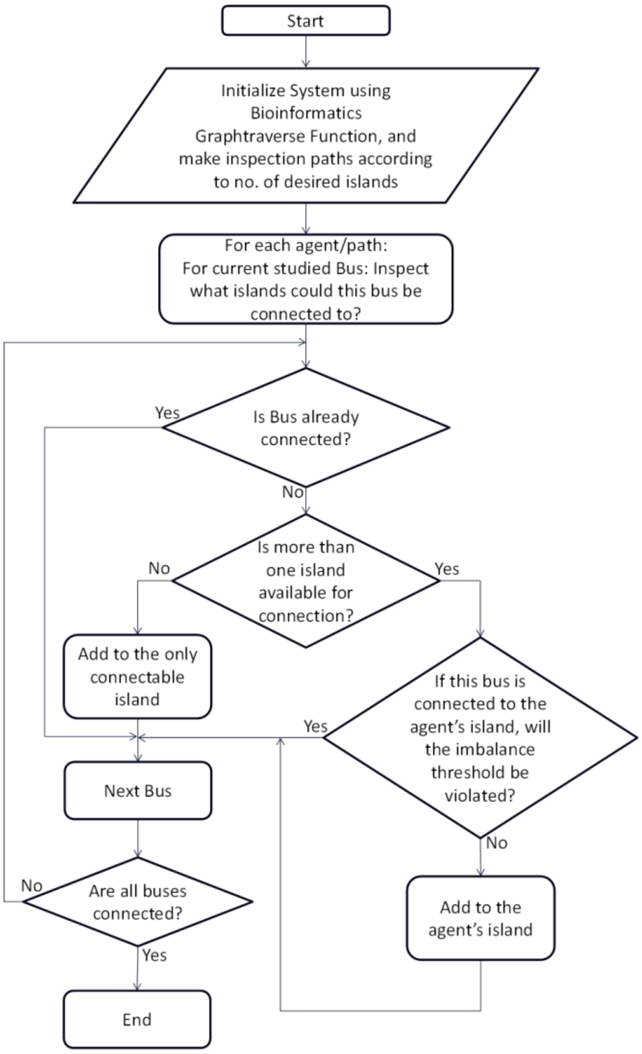


Fig. 4. New Sweeping and Keeping Controlled Separation Algorithm Flowchart

Table V - Parameters, Results and Times for 14-bus system (sweeping and keeping)

Number of lines	20 lines
Number of candidate configurations	2^20=1048576
Time taken to find the best separation technique	0.0468 seconds

3) Running the algorithm on the IEEE 30-bus standard test network is previewed. The following Table VI shows a sample of the results and times achieved by this algorithm for solving the IEEE 30-bus system:

Table VI Parameters, Results and Times for 30-bus system (sweeping and keeping)

Number of lines	41 lines
Number of candidate configurations	2^41=2.1990 x(10^12)
Time taken to find the best separation technique	0.3276 seconds

As shown, for solving the 30-bus system, it took the algorithm 0.3276 seconds. Again, this time is much shorter than all the previous literature results [1], [5], [6], [14] and [15] which were 2.5 seconds and 1 minute, noting also that the first two references were studying other constraints like post-separation violation of overloading limits and the last two references were outputting a group of solutions. The proposed algorithm only considers power imbalance constraints in the islands and presents one solution which is ideally the best in that regard. By far this is the fastest and most successful result, known to the researchers, for the 30-bus system.

7. Conclusions and Suggested Future Work

The work presented so far, suggests and leads to the road of applying the same algorithms to the IEEE 30-bus and bigger standard systems, but preliminary experiments show that bigger systems will result in a vast search space, that will let the first algorithm takes a very big time looking for suitable solutions. For example the 30 bus has 41 lines, which means that the search space will equal to $2^{41}=2.2x10^{12}$ possible configurations of the grid. Previous researches have looked into this matter and proposed solutions for this problem using different techniques like OBDD [1], [6] and BPSO [14]. Due to the random nature of the problem in binary space dimensions, the second algorithm/technique has only shown its big success in solving smaller grids. This still can be utilized if the bigger systems are clustered into smaller grids by compressing groups of buses or regions into one bus and considering the ties that only connect then to the other groups/regions as the only transmission lines studied. There is a vast variety of work that can be suggested to be performed in the future, targeting development of the controlled separation problem solutions and using other optimization techniques, in particular. Vast arrays of nature and animal behavior-based optimization techniques are already available. Some of these types are possible to implement for binary problems like the controlled separation problem.

Another research path may be into the possibilities and ways of how bigger systems may be automatically clustered and grouped to form smaller grids that can be operable by this algorithm developed so far. Buses could be grouped according to actual geographic locations, i.e. near-by buses will always stay together, or as another example, they could be grouped according to prior load flow and stability studies. This will make the problem far simpler and will save a lot of processing time. Then, the goal is to implement the solution online and present it as a real-time WAP. The third algorithm is very simple and can be easily implemented on any computer. It is also a potential platform for further development and introduction of more logic and decision making. Other constraints could be taken into consideration like performing a very quick and approximate DC load flow, similar to what is done by the first technique, and verify whether the solution will not cause any further cascading due to overloading of intact ties. Another step or stage could be added to the algorithm that decides the amount of load shedding according to input system data and available load feeders, their consumption and their priorities. This will help eliminate the imbalance and keep the system stable until the urgent case is cured and connections are restored. In this paper, researchers were trying to target the near perfect solution that will be implemented online and process data and act real-time in short time to save the system. The system could be expanded by including more modules to provide load shedding orders by finding the optimum number of load feeders to disconnect, taking into consideration the priority of their loads.

References

[1] Q. Zhao, K. Sun, D. Zheng, J. Ma and Q. Lu, "A study of system splitting strategies for island operation of power system: A two-phase method based on OBDDs" IEEE Transactions on Power Systems, Volume 18, no.4, November 2003.

[2] F. M. Elkady and A. Y. Abdelaziz, "Voltage stability assessment of electrical power systems using artificial neural networks," Journal of Engineering and Applied Science, Faculty of Engineering, Cairo University, Vol. 48, No. 4, pp. 727–743, August 2001.

[3] A. Y. Abdelaziz, M. M. Abu-Elnaga, M. A. Elsharkawy and K. M. Elbahrawy, 'Voltage Stability Assessment of Multi-machine Power Systems using Energy Function and Neural Networks Techniques', Electric Power Components and Systems Journal, Vol. 34, No. 12, December 2006, pp. 1313-1330.

[4] P. Kundur, Power System Stability and Control. 1994.

[5] Kai Sun, Kyeon Hur and Pei Zhang, "A new unified scheme for controlled power system separation using synchronized phasor measurements" IEEE Trans Power Systems, vol. 26 , pp. 1544-1554, 2011.

[6] Kai Sun, Qianchuan Zhao, Da-Zhong Zheng, Jin Ma and Qiang Lu, "A two-phase method based on OBDD for searching for splitting strategies of large-scale power systems" International Conference on Power System Technology, vol. 2 , pp. 834-838, 2002.

[7] A. M. El-Arabaty, H. A. Talaat, M. M. Mansour and A. Y. Abdelaziz, 'Out-of-Step Detection Based on Pattern Recognition', International Journal of Electrical Power and Energy Systems, Volume 16, No. 4, pp. 269-275, , August 1994.

[8] A. Y. Abdelaziz, M. R. Irving, A. M. El-Arabaty and M. M. Mansour, 'Out-of-Step Prediction Based on Artificial Neural Networks', Electric Power System Research, Volume 34, No. 2, pp. 135-142, , August 1995.

[9] Haibo You, Vijay Vittak, Xiaoming Wang, "Slow Coherency-based islanding" IEEE Trans Power Systems, vol. 19, pp. 483-491, 2004.

[10] Song Honglei, Wu Junyong and Wu Linfeng, "Controlled Islanding Based on Slow-coherency and KWP Theory" Innovative Smart Grid Technologies - Asia (ISGT Asia), IEEE, 2012.

[11] James Kennedy and Russel Eberhart, "Particle Swarm Optimization" IEEE International Conference Proceedings on Neural Networks, 1995.

[12] A. Y. Abdelaziz, S. F. Mekhamer, M. A. L. Badr, F. M. Mohammed and E.F. El-Saadany, 'A Modified Particle Swarm Algorithm for Distribution Systems Reconfiguration', Proceedings of the 2009 PES IEEE General Meeting, Calgary, Alberta, Canada, July 2009.

[13] Gary Pampara, Nelis Franken and A. P. Engelbrecht, "Combining Particle Swarm Optimisation with angle modulation to solve binary problem" The 2005 IEEE Congress on Evolutionary Computation, 2005.

[14] Wenxin Liu, Li Liu, David A. Cartes and Ganesh K. Venayagamoorthy "Binary Particle Swarm Optimization Based Defensive Islanding of Large Scale Power Systems" International Journal of Computer Science & Applications, vol. 4 Issue 3, pp 69-83, 2007.

[15] Li Liu, Wenxin Liu, David A. Cartes and Il-Yop Chung "Slow Coherency and Angle Modulated Particle Swarm Optimization based islanding of large-scale power systems" Advanced Engineering Informatics 23, 2009.

[16] MATLAB® Documentation.

International Journal of
Intelligent Systems and Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Speed Control of Direct Torque Controlled Induction Motor By using PI, Anti-Windup PI and Fuzzy Logic Controller

Hakan Açıkgöz[1*], Ö.Fatih Keçecioğlu[2], Ahmet Gani[2], Mustafa Şekkeli[2]

Received 9th March 2014, Accepted 29th September 2014

Abstract: In this study, comparison between PI controller, fuzzy logic controller (FLC) and an anti-windup PI (PI+AW) controller used for speed control with direct torque controlled induction motor is presented. Direct torque controlled induction motor drive system is implemented in MATLAB/Simulink environment and the FLC is developed using MATLAB/Fuzzy-Logic toolbox. The proposed control strategy is performed different operating conditions. Simulation results, obtained from PI controller, FLC and PI+AW controller showing the performance of the closed loop control systems, are illustrated in the paper. Simulation results show that FLC is more robust than PI and PI+AW controller against parameter variations and FLC gives better performance in terms of rise time, maximum peak overshoot and settling time.

Keywords: *Anti-windup PI controller, Direct torque control, Fuzzy logic controller, Induction motor, PI controller.*

1. Introduction

DC motors have high performance in terms of dynamic behaviour and their control is simple. Because its flux and torque can be controlled independently. However, DC motors have certain disadvantages due to the existence of the commutators and brushes. Nowadays, induction motors are extensively used in industrial application. Induction motors have complex mathematical models with high degree of nonlinear differential equations including speed and time dependent parameters. However, they are simple, rugged, inexpensive and available at all power ratings and they need little maintenance. Therefore, the speed control of induction motor is more important to achieve maximum torque and efficiency [1-5]. By the rapid development of microprocessor, power semiconductor technologies and various intelligent control algorithm, controlling methods of induction motors have been improved. In the recent years, researchs about induction motors which are common in industrial systems due to some important advantages are focused on vector based high performans control methods such as field orientation control (FOC) and Direct torque control (DTC) [1-7]. FOC principles were firstly presented by Blaschke [4] and Hasse [5]. FOC of induction motors are based on control principle of DC motors. Armature and excited winding currents of self-excited DC motors can be independently controlled because they are vertical to each other. There isn't such case in induction motors. Made studies on induction motors showed that these motors could be controlled such as DC motors if three-phase variables are converted to dq-axis and dq-axis currents are controlled. Vector control methods which are done transform of axis have been developed. Flux and torque of induction motors can be independently controlled. Thus induction motors can be used for variable speed drive applications [1-4].

DTC were firstly presented by Depenbrock [6] and Takahashi [7]. DTC method has simple structure and the main advantages of DTC are absence of complex coordinate transformations and current regulator systems. In the DTC method, the flux and torque of the motor are controlled directly using the flux and torque errors which are processed in two different hysteresis controllers (torque and flux). Optimum switching table depending on flux and torque hysteresis controller outputs is used to control of inverter switches in order to provide rapid flux and torque response. However, because of the hysteresis controllers, the DTC has disadvantage like high torque ripple.

In the recent years, FLC has found many applications. Fuzzy logic is a technique, improved by Zadeh [8] and it provides human-like behavior for control system. İt is widely used because FLC make possible to control nonlinear, uncertain systems even in the case where no mathematical model is available for the controlled system [8-14]. This paper deals with comparison of PI, FLC and PI+AW controller on speed control of direct torque controlled induction motor. The performance of FLC has been researched and compared with PI+AW and PI controller.

The rest of this paper is organized as follows. In Section II, direct torque control scheme is given. Section III describes proposed controller design. The simulation results are given in Section IV. Conclusions are presented in Section V.

2. DIRECT TORQUE CONTROL

The induction motor model can be developed from its fundamental electrical and mechanical equations. The d-q equations of 3-phase induction motor expressed in the stationary reference frame:

$$V_{ds} = R_s i_{ds} + p\psi_{ds} \quad (1)$$

$$V_{qs} = R_s i_{qs} + p\psi_{qs} \quad (2)$$

[1] *Kilis 7 Aralik University, Dept. of Electrical Science, Kilis/TURKEY*
[2] *K.Maras Sutcu Imam University, Dept. of Electrical and Electronics Engineering, K.Maras/TURKEY*
* *Corresponding Author: Email: hakanacikgoz@kilis.edu.tr*

$$0 = R_r i_{dr} + p\psi_{dr} - w_r \psi_{qr} \quad (3)$$

$$0 = R_r i_{qr} + p\psi_{qr} - w_r \psi_{dr} \quad (4)$$

The flux linkage equations:

$$\psi_{qs} = L_s i_{qs} + L_m i_{qr} \quad (5)$$

$$\psi_{ds} = L_s i_{ds} + L_m i_{dr} \quad (6)$$

$$\psi_{qr} = L_r i_{qr} + L_m i_{qs} \quad (7)$$

$$\psi_{dr} = L_r i_{dr} + L_m i_{ds} \quad (8)$$

Electromagnetic torque in the stationary reference frame is given as:

$$T_e = \frac{3}{2}\frac{P}{2}\left(\psi_{ds} i_{qs} - \psi_{qs} i_{ds}\right) \quad (9)$$

Where; $p = (d/dt)$, R_s, R_r are stator and rotor resistances; L_s, L_r, L_m are stator, rotor and mutual inductances; ψ_{ds}, ψ_{qs} are stator flux in d-q frame; ψ_{dr}, ψ_{qr} are rotor flux in d-q frame; i_{ds}, i_{qs}, i_{qr}, i_{qr} are stator and rotor currents in d-q frame and w_r is rotor speed.

DTC design is very simple and practicable. It consists of three parts such as DTC controller, torque-flux calculator and voltage source inverter (VSI) . In principle, the DTC method selects one of the inverter's six voltage vectors and two zero vectors in order to keep the stator flux and torque within a hysteresis band around the demand flux and torque magnitudes [1-6]. The torque produced by the induction motor can be expressed as shown below:

$$T_e = \frac{3}{2}\frac{P}{L_s}\frac{L_m}{L_r}\left|\overline{\psi_r}\right|\left|\overline{\psi_s}\right|\sin\alpha \quad (10)$$

Where, α is angle between the rotor flux and the stator flux vectors. ψ_r is the rotor flux magnitude and ψ_s is the stator flux magnitude. P is the pairs of poles, L_m is mutual inductance and L_r is rotor inductance. This equation (10) shows the torque is dependent on the stator flux magnitude, rotor flux magnitude and the phase angle between the stator and rotor flux vectors. The equation of induction motor stator is given by [6]:

$$\overline{V}_s = \frac{d\psi_s}{dt} + \overline{i}_s R_s \quad (11)$$

If the stator resistance is ignored, it can be approximated as equation (12) over a short time period [6-7]:

$$\Delta\overline{\psi_s} = \overline{V_s}\Delta t \quad (12)$$

This means that the applied voltage vector determines the change in the stator flux vector. If a voltage vector is applied to system, the stator flux changes to increase the phase angle between the stator flux and rotor flux vectors. Thus, the torque produced will increase [6-7].

Fig. 1 shows closed loop direct torque controlled induction motor system. The closed loop DTC induction motor system is implemented in MATLAB/Simulink environment. DTC induction motor model consists of four parts such as speed control, switching table, inverter and induction motor. d-q model is used for the induction motor design. DTC block has flux and torque within a hysteresis models. Two-level and three-level flux and torque within hysteresis band comparators are given in Fig. 2 and 3, respectively. Flux control is performed by two-level hysteresis band and three-level hysteresis band provides torque control. Outputs of the hysteresis bands are renewed in each sampling period and changing of the flux and torque are determined by these outputs. Voltage vectors are shown in Fig. 4. Flux control output $d\psi_s$, torque control output dT_e and voltage vector of the stator flux are determined a switching look-up table as shown in Table 1.

In DTC method, stator flux and torque are estimated to compare with references of the flux and torque values by aid of stator current, voltage and stator resistance. The obtained flux and torque errors are applied to the hysteresis layers. In these hysteresis layers, flux and torque bandwidth are defined. Afterwards, the amount of deflection is determined and the most appropriate voltage vectors are selected to apply to the inverter using switching look-up table.

If a torque increment is required then dT_e equals to +1, if a torque reduction is required then dT_e equals to -1 and if no change in the torque is required then dT_e equals to 0. If a stator flux increment is required then $d\psi_s$ is equals to +1, if a stator flux reduction is required then $d\psi_s$ equals to 0. In this way, the flux and torque control is realized.

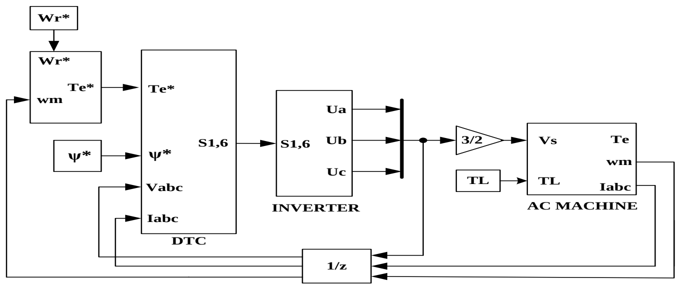


Fig. 1: DTC induction motor system in MATLAB/Simulink environment

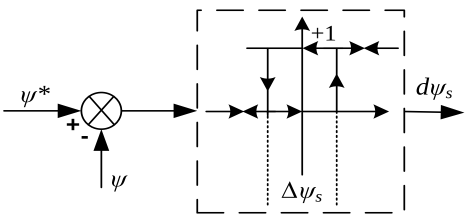


Fig. 2: Two-level flux hysteresis comparator

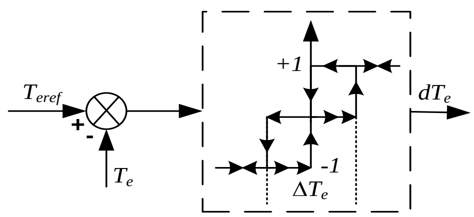


Fig. 3: Three-level torque hysteresis comparator

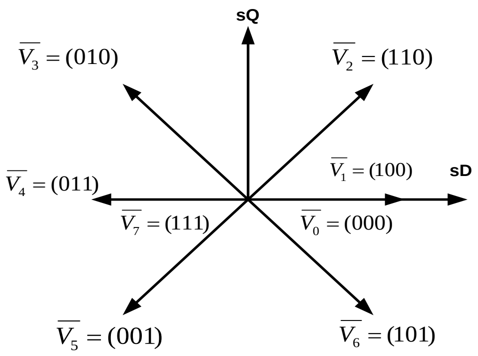


Fig. 4: Voltage vectors

Table 1: Switching Look-up Table

Flux (ψ)	Torque (T_e)	Sectors					
		SS1	SS2	SS3	S4	S5	S6
ψ=1	Te=1	V2	V3	V4	V5	V6	V1
	Te=-1	V6	V1	V2	V3	V4	V5
ψ=-1	Te=1	V3	V4	V5	V6	V1	V2
	Te=-1	V5	V6	V1	V2	V3	V4

3. DESIGN OF FLC, PI AND ANTI -WINDUP PI CONTROLLER

In this section, conventional PI controller, PI+AW controller and FLC are designed and applied to the DTC model. In the first design, the conventional PI controller and AW+PI controller are given to apply an induction motor drive in order to control its speed. In the second design, the FLC is designed for stability and robustness control. As a rule, the control algorithm for discrete PI controller can be described as:

$$u_{PI}(k) = K_P e(k) + K_I \sum_{i=1}^{k} e(k) \qquad (13)$$

Where, K_p is the proportional factor; K_I is the integral factor and e(k) is the error function. As shown in Fig. 5, the structure of PI controller is really simple and can be implemented easily. An anti-windup integrator is added to stop over-integration for the protection of the system in Fig. 6 [18-21].

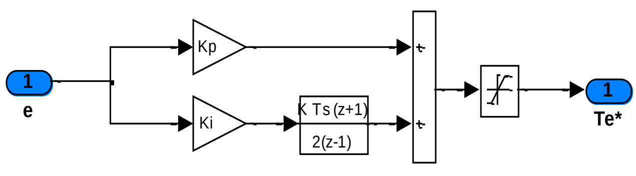


Fig 5: Simulink model of classic PI controller

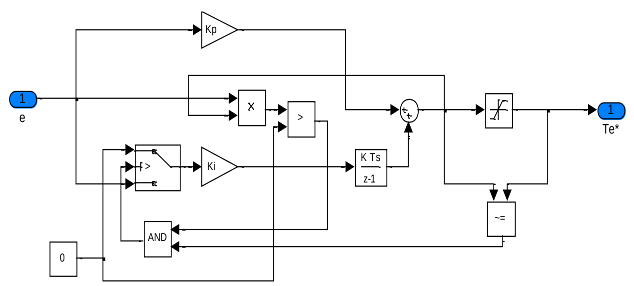


Fig. 6: Simulink model of PI controller with anti-windup

FLC is an appropriate method for designing nonlinear controllers via the use of heuristic information [9, 15]. A FLC system allows changing the control laws in order to deal with parameter variations and disturbances. Especially, the inputs of FLC are speed error and change in the speed error. These inputs are normalized to obtain error e(k) and its change Δe(k) in the range of -1 to +1. The fuzzy membership functions consist of seven fuzzy sets: NB, NM, NS, Z, PS, PM, PB as shown in Fig. 7.

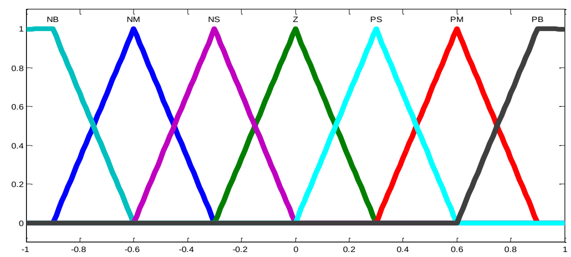


Fig. 7: Membership function of inputs and output

In the FLC, the rule has the form of: IF e is F^k_e AND de is F^k_{de} THEN du is w^k:

$$k = 1, \ldots, M \qquad (14)$$

F^k_e and F^k_{de} are the interval fuzzy sets and w^k is singleton output membership functions.

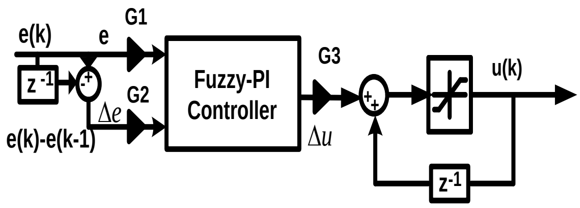


Fig. 8: Block diagram of Fuzzy-PI controller

The rule base of the FLC system is given in Table 2. The block diagram of FLC system for DTC is given in Fig. 8.

Table 2: Rule Base

e de	*NB*	*NM*	*NS*	*Z*	*PS*	*PM*	*PB*
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NM	NM	NS	Z	PS
NS	NB	NM	NS	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PS	PM	PB
PM	NS	Z	PS	PM	PM	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

4. SIMULATION RESULTS

Several simulation results for speed control of Direct Torque Controlled induction motor drive using PI, PI+AW and FLC is realized in MATLAB/Simulink environment. The simulations are performed for different reference speeds with load of 3N-m and no-load during 2 sec. The parameters of the induction motor used in the simulation are given in Table 3.

Table 3: Induction Motor Parameters

Parameters	Values
Power supply	3Φ
Stator resistance (Rs)	8.231Ω
Rotor resistance (Rr)	4.49Ω
Number of Poles (P)	2
Stator self-inductance (Ls)	0.599H
Rotor self-inductance (Lr)	0.599H
Moment of inertia (J)	0.0019kg-m2
Mutual inductance (Lm)	0.5787H

Fig. 9 shows the performance of PI, PI+AW and FLC. Conventional PI and PI+AW show overshoot during starting (%4.6 and %0.8, respectively). The PI controller response reaches to reference speed after 122 ms with overshoot and PI+AW response reaches to reference speed after 110 ms with overshoot. While the FLC response reaches to steady state after nearly 65 ms without overshoot. The simulation results show the FLC provides good speed response over the PI and PI+AW controller. The FLC performance is better than both of controllers in terms of settling time and maximum peak overshoot. The output torques controlled by PI+AW, PI and FLC controllers is illustrated in Fig. 10, 11 and 12, respectively.

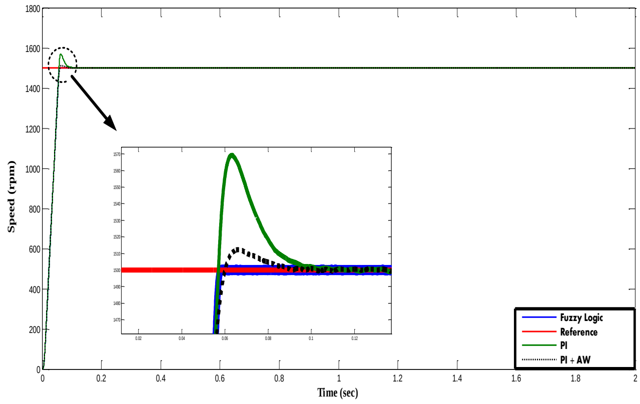


Fig. 9: Speed response comparison at no-load

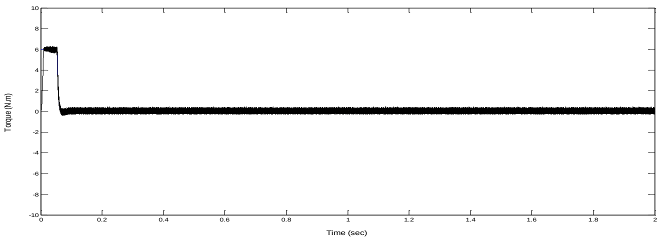


Fig. 10: The output torque response using PI+AW controller

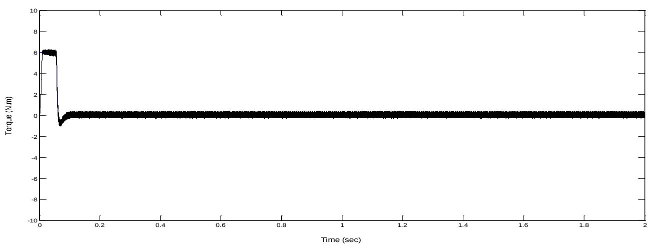


Fig. 11: The output torque response using PI controller

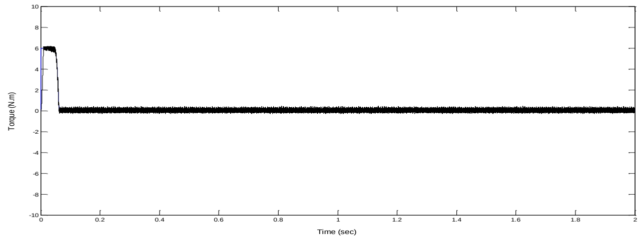


Fig. 12: The output torque response using FLC

Fig. 13 shows the speed tracking performance while sudden changing the speed from 1500 rpm to 1000 rpm at 0.8 sec. Firstly, DTC induction motor starts to operate in a steady state at 1500 rpm reference speed. Then, a sudden step speed command decreasing, from 1500 rpm to 1000 rpm is performed. The simulation results are given in the Fig. 13. The FLC follows the reference speed without any overshoot and steady state error. The performance of the FLC is much better than the PI and PI+AW controller for all speed change cases. The corresponding values are represented in Table 4. The torque responses of PI+AW, PI and FLC are given in Fig. 14, 15 and 16, respectively.

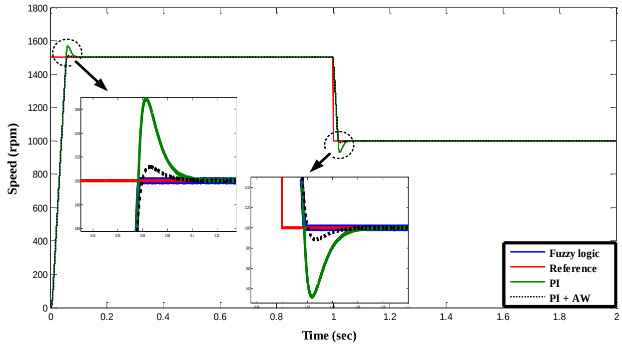


Fig. 13: Speed response comparison at no-load

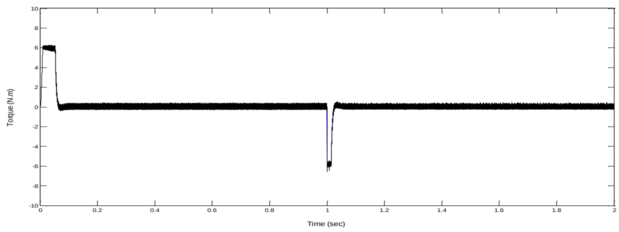


Fig. 14: The output torque response using PI+AW controller

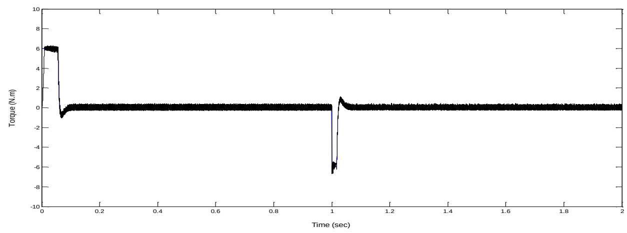


Fig. 15: The output torque response using PI controller

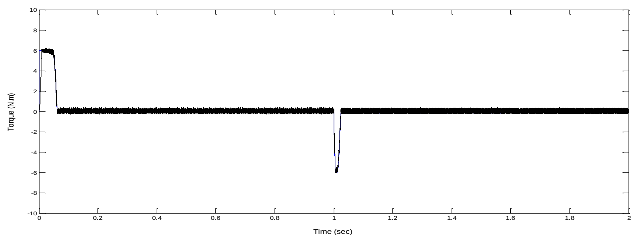


Fig. 16: The output torque response using FLC

Table 4: Performance of Controllers at No-Load

Controller Type	Settling Time ts(ms)	Overshoot Mp (%)
PI Controller	122ms 65ms (response to sudden step reduction)	%4.6(1st peak) %6.8(2nd peak)
PI + AW	110ms 52ms (response to sudden step reduction)	%0.8(1st peak) %1.12(2nd peak)
FLC	58ms 18ms(response to sudden step reduction)	%0(1st peak) %0(2nd peak)

Constant speed responses with load of 3N-m at 0.8sec are given in Fig. 17. The speed response with FLC has no overshoot and settles faster in comparison with PI and PI+AW controller and there is no steady-state error in the speed response. When the load is applied, there is sudden dip in speed. The speed falls from reference speed of 1500 rpm to 1490 rpm and it takes 3ms to reach the reference speed. The results of simulation show that the FLC gives better responses with respect to settling time and maximum peak overshoot. Moreover, the corresponding values are represented in Table 5. The torque responses of PI+AW, PI and FLC are given in figure 18, 19 and 20, respectively.

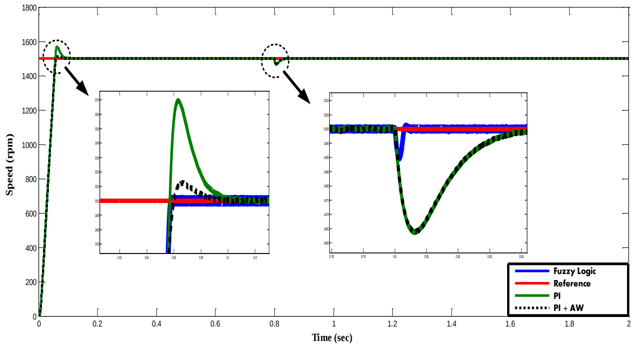


Fig. 17: Constant speed responses with load of 3N-m at 0.8 sec

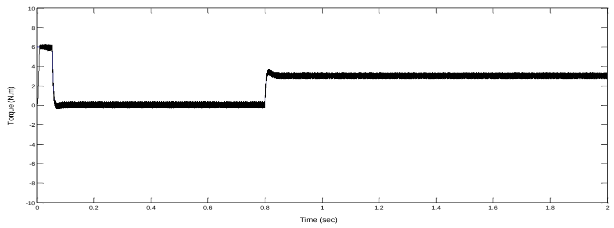


Fig. 18: The output torque response using PI+AW controller

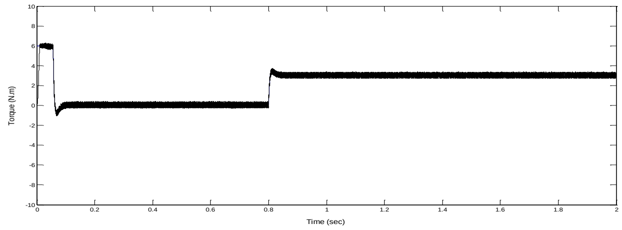


Fig. 19: The output torque response using PI controller

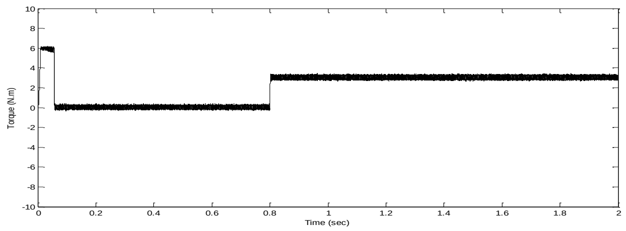


Fig. 20: The output torque response using FLC

Table 5: Performance of Controllers at Load

Controller Type	Settling Time ts(ms)	Overshoot Mp (%)
PI Controller	122ms 42.1ms(response to load torque)	%4.6(1st peak) %2.33(2nd peak)
Anti-Windup PI	110ms 42.1ms(response to load torque)	%0.8(1st peak) %2.33(2nd peak)
FLC	58ms 3ms(response to load torque)	%0(1st peak) %0.66(2nd peak)

5. CONCLUSIONS

In this study, Direct Torque Controlled induction motor drive system is presented and speed control of the induction motor is implemented. The motor drive system is carried out in MATLAB/Simulink environment using mathematical model of d-q of the induction motor. PI+AW controller, PI and FLC control systems are compared and effectiveness of the FLC against PI and PI+AW control performance is illustrated. Considering the overshoot and the response time, the FLC gives obviously better performance than PI and PI+AW controller. Moreover, it can be seen that the ripple in torque with FLC is less than PI and PI+AW controller for all speed change cases.

References

[1] K. Bose Bimal, “An Adaptive Hysteresis-Band Current Control Technique of a Voltage-Fed PWM Inverter for Machine Drive System,” IEEE Trans. Industrial Electronics, Vol 37, Oct. 1990, pp. 402-408.

[2] I. Takahashi and T. Noguchi, “A new quick-response and high efficiency control strategy of an induction motor,” in IEEE Transactions on Industry Application. Volume. IA, No. 5, 1986, pp. 820-827.

[3] T. G. Habetler, F. Profumo, M. Pastorelli, and L. M. Tolbert, “Direct torque control of induction machines using space vector modulation,” in IEEE Transactions on Industry Applications. Volume: 28, Issue: 5, 1992, pp. 1045 – 1053.

[4] M. Depenhrock. "Direct self-control of inverter-fed machine" IEEE Trans. Power Electron, X988,3:420-429

[5] D. Casadei and G.Serra, “Implementation of direct Torque control Algorithme for Induction Motors Based On Discrete Space Vector Modulation,” IEEE Trans. Power Electronics. Vol.15, No.4, July 2002.

[6] T. G. Habetler, F. Profumo, M. Pastorelli, and L. M. Tolbert, “Direct torque control of induction machines using space vector modulation,” in IEEE Transactions on Industry Applications. Volume: 28, Issue: 5, 1992, pp. 1045 – 1053.

[7] R. Toufouti and H. Benalla, “Direct torque control for induction motor using fuzzy logic,” in ACSE Journal. Volume: 6, Issue: 2, 2006. pp.19-26.

[8] R. Toufouti, and H. Benalla, “Direct torque control for induction motor using intelligent techniques,” in Journal of Theoritical and Applied Information Technology. 2007. pp. 35-44.

[9] F. Sheidaei, M. Sedighizadeh, S.H. Mohseni-Zonoozi, Y. Alinejad- Beromi, “A fuzzy logic direct torque control for induction motor sensorless drive,” in Universities Power Engineering Conference. 2007. pp. 197-202.

[10] Y. Tang and G. Lin, “ Direct Torque control of Induction Motor Based on Self-Adaptive PI Controller,” The 5th

International conference on computer Science & Education, Hefei, China. August 24-27, 2010.
[11] M. Baishan, L. Haihua, Z. Jinping, "Study of Fuzzy control in Direct Torque Control system," International Conference on Artificial Intelligence and Computational Intelligence, 2009.
[12] N.H. Ab Aziz, A. Ab Rahman, "Simulation on Simulink AC4 model (200HP DTC Induction Motor Drive) Using Fuzzy Logic Controller," International Conference Computer Applications and Industrial Electronics (ICCAIE) December 5-7, Kuala Lumpu, Malaysia, 2010.
[13] I.Ludthe, "The Direct Control of Induction Motors," Thesis, Department of Electronics and Information Technology, University of Glamorgan, May 1998.
[14] L. A. Zadeh, "fuzzy sets," Inform, Control, Vol.8, 1965, pp.338-353.
[15] Y.V.Siva Reddy, T.Brahmananda Reddy, M.Vijaya Kumara, "Direct Torque Control of Induction Motor using Robust Fuzzy Variable Structure Controller", International J. of Recent Trends in Engineering and Technology, Vol.3, No.3, May 2010.
[16] Vinod Kumar, and R.R. Joshi, "Hybrid controller based intelligent speed control of induction motor," Journal of Theoretical and Applied Information Technology (JATIT), 2005,pp-71-75,
[17] V Chitra, and R. S Prabhakar,. "Induction Motor Speed Control using Fuzzy Logic Controller", Proc. of World Academy of Science, Engineering and Technology, Vol. 17, December 2006, pp. 248-253.
[18] M. Gaddam, "Improvement in Dynamic Response of Electrical Machines with PID and Fuzzy Logic Based Controllers," Proceedings of the World Congress on Engineering and Computer Science, WCECS 2007, San Francisco, USA, October 24-26, 2207.
[19] M. Sekkeli, C. Yıldız, H. R. Ozcalik, "Fuzzy Logic Based Intelligent Speed Control of Induction Motor Using Experimental Approach" International Symposium on 1Nnovaitons in intelligent SysTems and Applications, Trabzon/Turkey, June 29, July 1, 2009 INISTA
[20] K.,Klinlaor, C.,Nontawat, "Improved Speed Control Using Anti-windup PI controller For Direct Torque Control Based on Permanent Magnet Synchronous Motor",12th International Conference on Control, Automation and Systems, Jeju Island/Korea, 17-21 Oct., 2012.
[21] D. Korkmaz, O. G. Koca, Z. H. Akpolat, "Robust Forward Speed Control of a Robotic Fish," Sixth International Advanced Technologies Symposium, Elazig/Turkey, May 16-18, 2011, pp.33-38.