ALTERNATE FUEL OPTIONS IN THE KILN

Riya Maria Kurian1, Divya. C. R2

[1] M. Tech Scholar, Energy Systems, [2] Department of EEE

Nehru College of Engineeering and Research Centre, Thrissur, (India)

ABSTRACT

The sources of fossil fuels are depleting each day and thus there is a need for the search of alternative fuels for the kiln. Nowadays, paper industry faces a major problem with the increasing cost of furnace oil. Furnace oil is used in the kilns as fuel for firing for the production of lime. The objective is to substitute an alternative fuel for furnace oil. In the present work used engine oil is blended with furnace oil in the ratio of 75 to 25 (by weight) and is used as an alternative fuel for furnace oil. The paper focuses on the study of calorific value, flash point, ash content, viscosity of the furnace oil and the blended oil samples. The pollution policies of the industries are also taken into account.

Keywords - Blend, Combustion Properties, Furnace Oil, Lime Kiln, Used Engine Oil

I. INTRODUCTION

The depletion of world petroleum sources and increased environmental concerns have stimulated recent interest in alternative sources for petroleum based fuels for firing. With rising energy costs and new environmental regulations in the past years, many paper industries have made it a priority to reduce their energy consumption and operation expenses. Lime reburning kiln is the biggest user of fossil fuels and only part of the mill, which needs significant purchasing of fuel. Fluctuating prices for fossil fuels and more stringent carbon taxes has made lime kiln energy consumption an important issue impacting the overall pulp mill profitability.

Lime kiln operating expenses can be decreased with increasing the thermal efficiency of the kiln and using new fuels for the combustion process. Most of lime kilns use heavy fuel oil as their energy source, but many mills have interest to replace them with alternative renewable fuels in the future. Operation of the lime kiln affects the whole pulp mill and it must stay stable to produce acceptable quality lime and to keep pulp products from the mill good-quality.

Many things have to be considered when replacing the traditional fuels used in the kiln. Availability, heating value, chemical composition and combustion behaviour of the alternative fuels are important matters when examining the effects of replacing on combustion, flue gas emissions and economy of the pulp mill.

Aim of this study is to examine the requirements of lime kiln fuel and to find the best fuel substitutes for the kiln. This study focuses on the blend of used engine oil and furnace oil. The waste engine oil can be obtained from the industry itself. The engine oil is used as lubricating oil in pumps and gears of various equipments and is removed and replaced periodically with fresh engine oil. This used engine oil is disposed of after use. Thus, this used oil can be used as a substitute.

The used oil may be considered as a hazardous waste and have to be disposed of according to Environmental Protection Agency (EPA) regulations. The used oil have the advantage of being inexpensive as compared to

conventional fuels and they are readily available at garages and oil change service centers, vehicle dismantlers, machine shops and industries. This paper describes a new approach to lime kiln fuel for firing that combines furnace oil and used engine oil in the ratio of 75 to 25(by weight).

II. OVERVIEW

Production of diesel fuel from used engine oil involves chemical filtrations and blending process. It could solve some of the energy problem with increasing the blending percentage of pre-treated used engine oil.

Used engine oils from vehicles, machinery have high heat value and can fit into many greenhouse operations. It is found that when even a small amount of used engine oil is co fired with the gaseous fuel it significantly enhances thermal radiation capabilities of the gaseous fuel flame.

III. DESCRIPTION OF THE KILN OPERATION

3.1 Recauticizing Process In Kraft Pulp Mill

Efficient and closed chemical recovery is great benefit of the paper mill process. It makes recirculation of cooking chemicals in the process possible while using only little amount of makeup chemicals. Recausticizing plant is important part of chemical recovery at pulp mill.

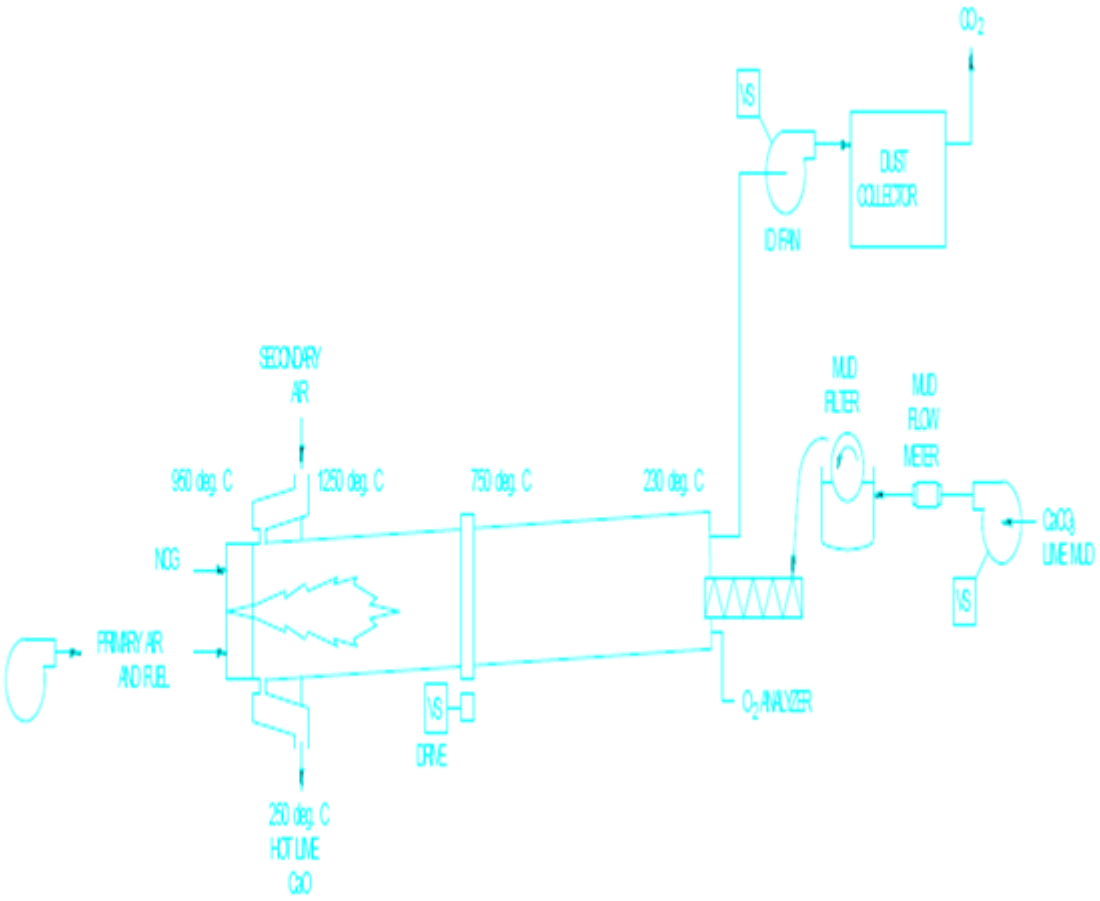


Figure 1 Lime Kiln Installation

Sodium hydroxide, which is mixed with the wood chips and cooked in the digester, is recovered in the recovery boiler and regenerated in the recausticizing plant. This is done by exchanging the sodium ion in the green liquor

(sodium carbonate stream) from the recovery boiler with a calcium ion from calcium hydroxide. After the exchange, sodium hydroxide is mixed with wood chips in the digester and the calcium carbonate is calcined to form CaO in the rotary kiln. The calcium oxide is then slaked to form calcium hydroxide. The calcium hydroxide is mixed with the sodium carbonate from the recovery boiler and the whole cycle is repeated. The lime calcining process used by most kraft pulp mills uses a rotary lime kiln to convert the calcium carbonate mud from the clarifiers to quick lime or calcium oxide.

A typical kiln installation is shown in Fig. 1. It uses green liquor from recovery boiler as raw material and consumes lime, calcium oxide (CaO) to produce white liquor, which is an important chemical used in pulping.

The recauctisizing process has two targets, to produce clean, hot white liquor containing minimum amount of unreactive chemicals for the cooking process, and prepare clean and dry lime mud to burn in the lime kiln for reuse as lime with minimum energy usage.

Two important reactions of recausticizing are slaking and causticizing. When green liquor is mixed with lime (CaO) it slakes with water and forms calcium hydroxide ($Ca(OH)_2$). Calcium hydroxide continues to react with sodium carbonate (Na_2CO_3) in green liquor forming sodium hydroxide (NaOH), main compound in white liquor and also calcium carbonate ($CaCO_3$), called lime mud as by-product. Fig. 2 shows caustisizing process as a part of the kraft pulp mill chemical recovery circuit.

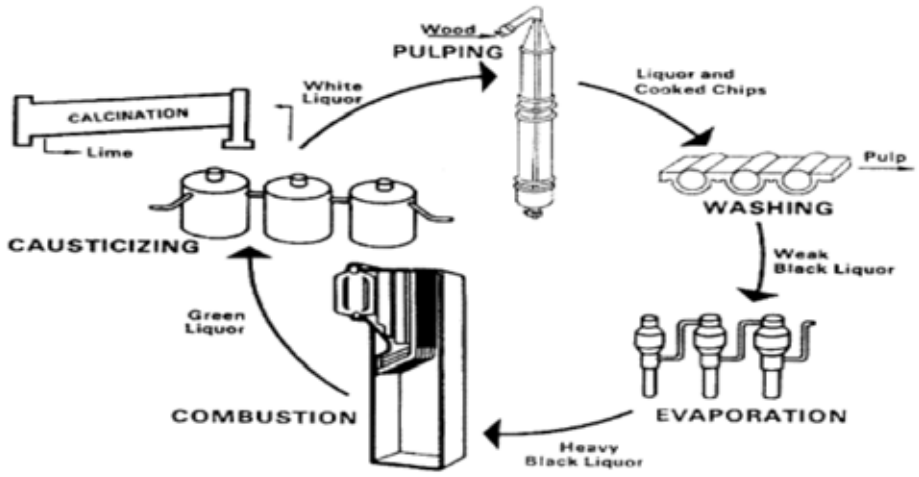


Figure 2 Causticizing process of the pulp mill recovery circuit

3.2 Function And Construction Of Lime Kiln

Lime reburning is a part of chemical circuit called lime cycle. Lime regeneration is called reburning because it involves treating lime mud in high temperatures in a lime kiln. The function of the lime kiln is to convert lime mud back to lime for reuse in the causticizing process. Equation 1 shows the conversion from lime mud to lime.

$$CaCO_3\,(s) + \text{heat} \leftrightarrow CaO\,(s) + CO_2\,(g). \qquad (1)$$

Lime kiln is a rotary combustion kiln where heat transfers from combustion gas to lime particles. Lime kilns are typically 2-4 m in diameter and 50-120 m in length with typical rotational speed of 0.5-1.5 rpm. Lime mud is fed to the kiln from cold end and the kiln slopes slightly, about 1-4 per cent toward the firing end. Lime mud moves slowly through the bottom of the kiln towards the firing end as result of inclination and rotation. Flue gases and lime dust exits the kiln from the cold end. Flue gases pass through electrostatic precipitator and wet scrubber and lime dust captured in the precipitator is fed back to the kiln.

Figure 3 Rotary kiln

Lime retention time in the kiln is approximately 2.5-4 hours depending on kiln dimensions, rotational speed and lime mud properties. Lime kiln can be divided to four process zones according to the temperature profile of solids and fuel gases:

1) Thermal drying: moisture in the lime mud evaporates.
2) Heating: lime mud gets heated to the reaction temperature.
3) Calcination: calcium carbonate dissociates into calcium oxide and carbon dioxide.
4) Sintering and cooling: formed fine powder agglomerates into nodules and then cools before leaving the kiln.

Fig. 4 shows the lime kiln heating zones. Red line in the figure is fuel oil. Calcination reaction occurs in the actual burning zone where gas temperature increases to 1100°C. The endothermic calcination occurs spontaneously when lime mud reaches 800°C and sufficient reaction rate is reached approximately at 1100°C. The flue gas temperature needs to be significantly higher because of the poor heat transfer in the kiln. Lime mud from lime mud silo is mechanically dried in filter plant before feeding it to the kiln. This is called lime mud dewatering and its purpose is to increase the dry solids in the mud. The moisture in lime mud has a significant effect on the energy consumption of the kiln. It consists of a mud filter where the moisture content is reduced to about 30 to 35%, a rotary kiln and dust collector. In LMD dryer the lime mud is fed to a flue gas stream where the heat of the gases dries the mud. Then a cyclone separates dry mud and feeds it to the kiln. Lime mud has also to be sintered in the kiln to make usable product for further processing. The kiln generally consists of three sections. The first section is a preheater or chain section where the calcium carbonate mud is dried and nodulised. In the final zone, lime powder agglomerates into lime nodules with diameter of 10-50 mm. The second section is a calcining zone where the feed is heated to about 1150°C and converted to calcium oxide.

Most kilns have a third section where the quick lime is cooled before leaving the kiln and the secondary combustion air is preheated. The cooler consists of a number of tubes arranged around the circumference of the kiln at the firing end of the kiln where lime heat is recovered to combustion air. Production lime move through the product cooler and the temperature at the outlet is $(70\text{-}90)^{o}$ C. The burned lime from the kiln has a wide particle size distribution. Oversized particles are crushed by a lump crusher or hammer mill after leaving the kiln.

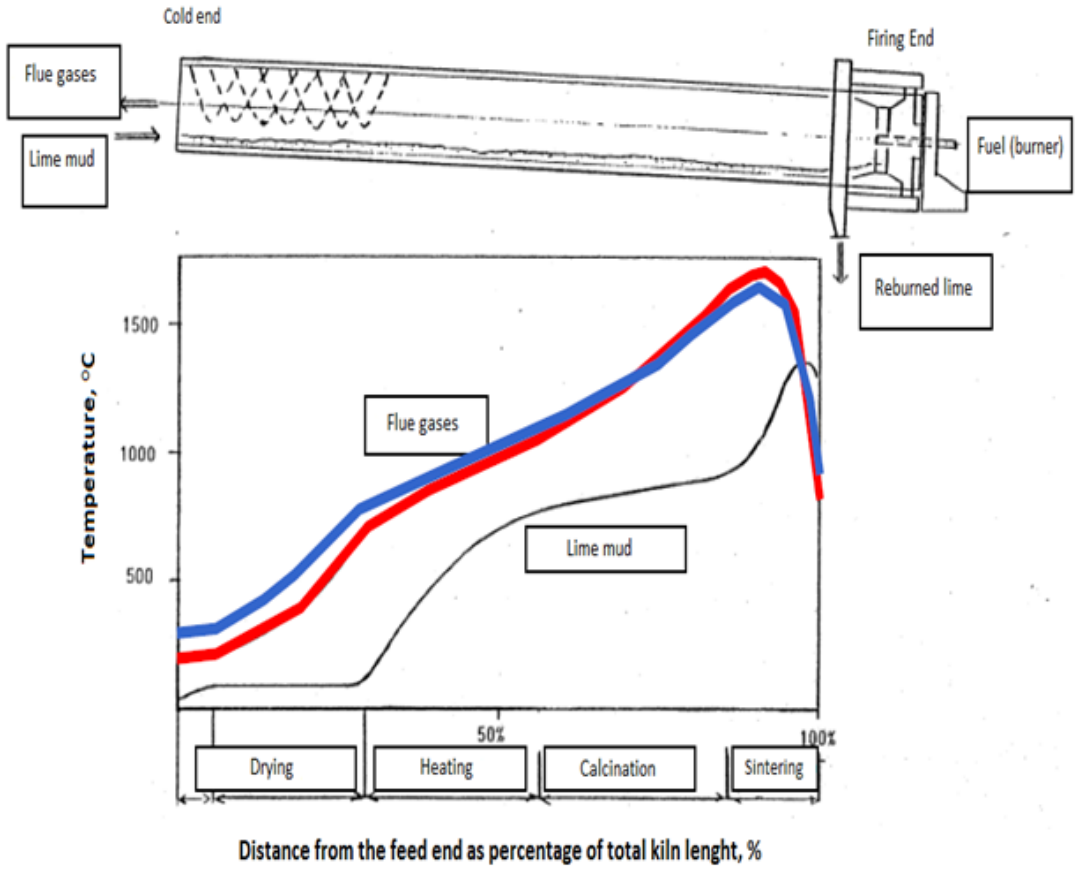


Figure 4 Lime Kiln Heating Zones

Atmospheric air is used as coolant. The lime before stabilization is not pure and contains certain lime sludge thus the drainer portion at the product end drains the impure lime and lime mud. Secondary air is used in normal and parallel to the kiln to achieve maximum purity and proper burning. All lime kilns have a refractory lining that protects kiln shell from overheating and limits heat losses. . Refractory system consists of bricks that are composed of special heat-resistant and chemical-attack resistant materials, such as alumina or silica components. Each kiln zone has a lining of a certain material and thickness.

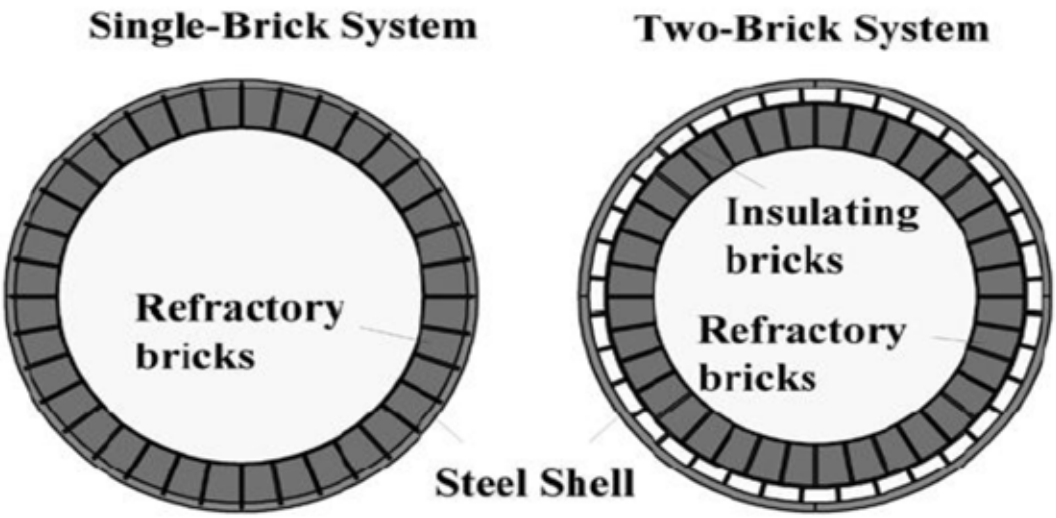


Figure 5 Brick Lining In Lime Kiln

Treatment of lime mud in the lime kiln requires external heat and this requires high fuel combustion temperature. Higher flame temperatures mean higher production capacity and efficiency, but too high temperatures cause refractory damage and over-burned, slow-reacting lime product. Therefore stability and control of the combustion temperature are also important to make good quality lime and to maintain stable operation of the kiln.

3.3 Fuels Used And Their Emissions

Main fuels used in lime kilns are heavy fuel oil or natural gas. Lime kiln is the biggest user of fossil fuels in production process and the only part of the paper mill that needs substantial purchasing of fuel. Carbon dioxide (CO_2) emission from the kiln is directly proportional to the carbon (C) in the kiln gas. This comes from two sources: lime mud conversion and combustion of fuel. Two thirds of the carbon emissions come from the lime mud conversion and one third from the fuel combustion. Carbon in lime mud originates from wood and can be considered as carbon neutral. Carbon dioxide from fuel combustion has positive carbon footprint and if fossil fuels are used for combustion, they are counted as greenhouse gas (GHG) emissions.

Lime kiln always need some amount of makeup lime to cover lime losses and lime containing impurities and thus sea shells are fed to the kiln, after washing, through the cold end.

Although losses of calcium from recovery system are usually made up using fresh lime, some amounts of make-up $CaCO_3$ are used in the kiln. Carbon contained in $CaCO_3$ is usually fossil origin and escapes as CO_2 from the kiln. This is also counted as fossil CO_2 emission.

Rising and unstable price of fuel oil has increased production costs in paper mills. Therefore, there is a need for paper mills to find more economical, carbon neutral alternative fuels that have minimal impact on lime kiln operation and chemical recovery process.

IV. EXPERIMENTAL DETAILS

Initially, the used engine oil was filtered using centrifugation at ambient temperature and humidity. The used oil was taken in four test tubes and placed inside the centrifuge. Due the rotation of the oils at high speed the impurities are settled at the bottom of the test tubes and filtered engine oil is obtained.

TABLE I. ENGINE OIL FILTERATION

Sl. No.	**Activity**			
		Time (minutes)	***Speed (rpm)***	***Observation***
1)	Centrifuge process	10	3,300	No sediments separated
		20	3,300	Sediments separated- lesser volume
		20	4,000	Sediments separated- higher volume
		20	4,000	
		20	4,000	

Sl. No.	Activity			
		Time (minutes)	*Speed (rpm)*	*Observation*
		20	4,000	
		20	4,000	
		20	4,000	
		20	4,000	
2)	Centrifuge after filteration	30	4,000	Moisture (mist) separated
		40	4,000	Moisture (droplets) separated

V. RESULTS

The calorific value of the furnace oil and filtered engine oil were tested separately using bomb calorimeter and the test results are explained in Table II.

The furnace oil and filtered engine oil was blended in two different ratios by weight for testing. The ratios are

1) 75: 25

2) 60: 40

In the ratio of 75 :25(by weight) means 0.375L of furnace oil and 0.125L of filtered engine oil is mixed together and in the 60: 40 ratio(by weight) means 0.3L of furnace oil and 0.2L of filtered engine oil is blended together. The blended oils are tested for calorific values for selecting the desired blend of the oil to be used an alternate fuel in the kiln.

TABLE II. TEST RESULTS

Sl. No.	***Results***	
	Sample	***Calorific value in KJ/ Kg***
1)	Furmace oil	46,938.215
2)	Filtered engine oil	49,676.382

TABLE III. BLENDING OF OILS

Sl. No.	Results			
		Furnace oil in L	*Filtered engine oil in L*	*Observation*
1)	Blending	0.375	0.125	There is no much difference between the ratios by visual
		0.3	0.2	

TABLE IV. TEST RESULTS

Sl. No.	***Results***	
	Sample	***Calorific value in KJ/ Kg***
1)	Furmace oil and filtered engine mix in the ratio 75:25	46,645.139
2)	Furnace oil and filtered engine oil mix in the ratio 60:40	46,929.841

The blended oil with less filtered engine oil concentration was selected as the flash point increases with the increase in concentration of filtered engine oil. Thus, the blended oil in the ratio of 75: 25(by weight) was taken for further testing of the combustion properties.

The blended oil was tested for relative density, kinematic viscosity, flash point, calorific value, ash content, acidity, sulphur and moisture content.

5.1 Density

Density is the ratio of the mass of the fuel to the volume of the fuel at a reference temperature of 15^{o}C and is measured using a hydrometer.

5.2 Kinematic Viscosity

Viscosity plays a key role in handling and storing the fuels. Viscosity is an internal property of fluid that offer resistance to flow. Viscosity depends on temperature and decreases as the temperature increases. Any numerical value for viscosity has no meaning unless the temperature is also specified. The lower the viscosity is, the better the liquid flows and is usually measured the instrument called using saybolt viscometer. It influences the degree of pre-heat required for handling, storage and satisfactory atomization. If the oil is too viscous, it may become difficult to pump, hard to light the burner, and tough to operate. Poor atomization may result in the formation of carbon deposits on the burner tips or on the walls. Therefore pre-heating is necessary for proper atomization.

5.3 Flash Point

The flash point of a fuel is the lowest temperature at which the fuel can be heated so that the vapour gives off flashes momentarily when an open flame is passed over it. The instrument used in measuring the flash point is Pensky Martens closed apparatus.

5.4 Calorific Value

The calorific value is the measurement of heat or energy produced, and is measured either as gross calorific value or net calorific value. The difference being the latent heat of condensation of the water vapour produced during the combustion process. Gross calorific value (GCV) assumes all vapour produced during the combustion process is fully condensed. Net calorific value (NCV) assumes the water leaves with the combustion products without fully being condensed. Fuels should be compared based on the net calorific value and it determines the amount of fuel needed for heat transfer in the kiln.

The calorific value of coal varies considerably depending on the ash and moisture content of the fuel and is measured using bomb calorimeter.

5.5 Ash Content

The ash value is related to the inorganic material in the fuel oil. The ash levels of distillate fuels are negligible. Residual fuels have more of the ash-forming constituents. These salts may be compounds of sodium, vanadium, calcium, magnesium, silicon, iron, aluminium, nickel, etc. Typically, the ash value is in the range 0.03–0.07%. Excessive ash in liquid fuels can cause fouling deposits in the combustion equipment. Ash has erosive effect on the burner tips, causes damage to the refractory at high temperatures and gives rise to high temperature corrosion and fouling of equipments.

5.6 Water Content

Water content in the oil may be present in free or emulsified form and can cause damage to the inside furnace surfaces during combustion especially if it contains dissolved salts. It can also cause spluttering of the flame at the burner tip, possibly extinguishing the flame and reducing the flame temperature or lengthening the flame.

5.7 Sulphur Content

The amount of sulphur in the fuel oil depends mainly on the source of the crude oil and to a lesser extent on the refining process. The main disadvantage of sulphur is the risk of corrosion by sulphuric acid formed during and after combustion, and condensing in cool parts of the chimney or stack, air preheater and economiser.

TABLE V. COMBUSTION PROPERTIES OF THE BLENDED OIL

Sl. No.	**Comparison**		
	Properties	***Requirements***	***Blended oil in the ratio of 75: 25(by weight)***
1)	Relative density at 15 ^{o}C in kg/m^{3}	850-990	960
2)	Kinematic viscosity in µm^{2}/ s at 50 ^{o}C	85-125	105.41
3)	Flash point, ^{o}C	66	132
4)	Gross calorific value in KJ/Kg	43,961.4	46,733.062
5)	Ash (% by mass)	0.1	0.01
6)	Water content (% by mass)	1	Nil
7)	Sulphur content (% by mass)	2-4.5	1.7

Sl. No.	Comparison		
	Properties	***Requirements***	***Blended oil in the ratio of 75: 25(by weight)***
8)	Acidity inorganic	Nil	Nil

The blended oil from the test results conforms to the required specifications of furnace oil and thus can be used in lime kilns for firing as an alternative fuel.

VI. CONCLUSION

Due to rising price of heavy fuel oil and natural gas and increasingly stringent environmental regulations in recent years, the interest in replacing these conventional fuels used in pulp mill lime kilns with alternative fuels has become a worldwide issue. Pulp manufacturers are looking both for money savings and nowadays increasingly important environmental reputation.

Fuels used for lime kilns have a lot of requirements compared to conventional combustion in heating boilers. As the operation of lime kiln requires lot of energy and affects to the whole chemical circuit of the pulping process, the used fuel should have high heating value, good availability, constant combustion properties and should not contain much of nitrogen, sulphur and impurities. The test results of blended mix of furnace oil and filtered engine oil in the ratio of 75: 25(by weight) suggest that the blended oil have values in the permissible limit. The advantage is that the blended oil have less ash, sulphur and water content and high calorific value to that of furnace oil and thus reduces the consumption of the fuel and increases the efficiency of the kiln. The pollution policies are also satisfied. By knowing these requirements, replacing heavy fuel oil or natural gas with the blended oil is much easier and the operation of the kiln is predictable. Thus, the blended fuel can be used as alternative fuel in the lime kiln for firing.

VII. ACKNOWLEDGMENT

The authors gratefully acknowledge Rajeevan. K, Senior Manager (Energy), Hindustan Newsprint Limited, Kerala, G. V. Ramamurthy, MSME testing centre, Chennai and the deputy director of C.T.A.L, Chennai for their valuable time and assistance.

REFERENCES

[1] Miyamoto N, Ogawa H, Nabi MN, "Approaches to extremely low emissions and efficient diesesl combustion with oxygenated fuels", International Journal of Engine Research, pp. 71-85, 2000.

[2] John W. Bartok, “Approximate heating value of common fuels”, Storrs, December 2004.

[3] Luka Zajec, “Slow pyrolysis in a rotary kiln reactor: optimization and experiments”, A Master’s Thesis done at the School for Renewable Energy Science, Akureyri, February 2009.

[4] Martin Bajus, Natalia Olahova Slovak, “Thermal conversion of scrap tyres”, Petroleum and Coal, Vol. 53(2), pp. 98-105, 2011.

[5] Moses P.M. Chinyama, “Alternative fuels in cement manufacturing”, Alternative fuel InTech, 2011.

[6] Professor Paul T. Williams, **"**Fuels, chemicals and materials from waste", Energy Research Institute University of Leeds, March 2012.

[7] Ossi Ikonen, "Alternative liquid biofuels for lime kilns", Bachelor's Thesis, Lappeenranta University of Technology, Lappeenranta, April 2012.

[8] Anders Kallenberg, "Liquid bio fuels for gas turbines", Master Thesis in Engineering Physics, Sweden, February 2013.

[9] K. Naima and A. Liazid, "Waste oils as alternative fuel for diesel engine ", Journal of Petroleum Technology and Alternative Fuels, Vol. 4(3), pp. 30-43, March 2013.

[10] K. Arumugam, S. Veeraraja and P. Esakkimuthu, "Combustion of waste/ used oil by using specialized burner", International Journal of Applied Engineering Research, Vol. 8(15), pp. 1839-1846, November 2013.

[11] K. Srinivas, N. Ramakrishna, Dr. B. Balu Naik, Dr. K. Kalyani Radha, "Performance and emission analysis of waste vegetable oil and it's blends with diesel and additive", International Journal of Engineering Research and Applications, Vol. 3, Issue 6, pp. 473-478, Nov-Dec 2013.

[12] www.andritz.com.

$f(123)$- MULTIRECURRENT Hsu-STRUCTURE MANIFOLD

Lata Bisht[1], Sandhana Shanker[2]

[1] *Applied Science Department, BTKIT, Dwarahat, Almora, Uttarakhand, India-263653*

[2]*Department of Mathematics , Reva University, Bangalore, India-560064*

ABSTRACT

In this paper we have defined -multirecurrent and -multirecurrent symmetric Hsu- Structure manifold.

Keywords*: *Curvature Tensors, Hsu-structure manifold, Multirecurrent parameter.

I. INTRODUTION

If on an even dimensional manifold V_n, n = 2m of differentiability class C^∞, there exists a vector valued real linear function f , satisfying

$$f^2 = a^r I_n, \qquad (1.1a)$$

$$\overline{\overline{X}} = a^r X \text{ , for arbitrary vector field X.} \qquad (1.1b)$$

where $\overline{X} = fX$, $0 \le r \le n$ and 'a' is a real or imaginary number.

Then {f } is said to give to V_n a Hsu-structure defined by the equations (1.1) and the manifold V_n is called a Hsu-structure manifold.

The equation (1.1) gives different structure for different values of 'a' and 'r'.

If $r = 0$, it is an almost product structure.

If $a = 0$, it is an almost tangent structure.

If $r = \pm 1$ and $a = +1$, it is an almost product structure.

If $r = \pm 1$ and $a = -1$, it is an almost complex structure.

If $r = 2$ then it is a GF-structure which includes

π-structure for $a \neq 0$,

an almost complex structure for $a = \pm i$,

an almost product structure for $a = \pm 1$,

an almost tangent structure for $a = 0$.

Let the Hsu-structure be endowed with a metric tensor g, Such that

$$g(fX, fY) + a^r g(X, Y) = 0.$$

Then $\{f, g\}$ is said to give to V_n - metric Hsu-structure and V_n is called a metric Hsu-structure manifold.

The curvature tensor R, a vector -valued tri-linear function w.r.t. the connexion D is given by

$$R(X,Y)Z = D_X D_Y Z - D_Y D_X Z - D_{[X,Y]}Z, \quad (1.2a)$$

where

$$[X,Y] = D_X Y - D_Y X. \quad (1.2b)$$

The Ricci tensor S in V_n is given by

$$S(Y,Z) = (C_1^{\ 1}R)(Y,Z). \quad (1.3)$$

Where by $(C_1^{\ 1}R)(Y,Z)$, we mean the contraction of $R(X,Y)Z$ with respect to first slot.

For Ricci tensor, we also have

$$S(Y,Z) = S(Z,Y), \quad (1.4)$$

A linear map g defined by,

$$S(Y,Z) = g(gY),Z) = g(Y,gZ)), \quad (1.5)$$

The scalar k define by $k = C_1^{\ 1}g$ (1.6)

is called the scalar curvature of V_n

Let W, C, L and V be the Projective, conformal, conharmonic and concircular curvature tensors respectively given by

$$W(X,Y)Z = K(X,Y)Z - \frac{1}{(n-1)}[S(Y,Z)X - S(X,Z)Y]. \quad (1.7)$$

$$C(X,Y)Z = K(X,Y)Z - \frac{1}{(n-2)}[S(Y,Z)X - S(X,Z)Y + g(Y,Z)g(X) - g(X,Z)g(Y)] +$$

$$+ \frac{k}{(n-1)(n-2)}[g(Y,Z)X - g(X,Z)Y]. \quad (1.8)$$

$$L(X,Y)Z = K(X,Y)Z - \frac{1}{(n-2)}[S(Y,Z)X - S(X,Z)Y - g(X,Z)g(Y) + g(Y,Z)g(X)]. \quad (1.9)$$

$$V(X,Y)Z = K(X,Y)Z - \frac{k}{n(n-1)}[g(Y,Z)X - g(X,Z)Y]. \quad (1.10)$$

A manifold is said to be recurrent, if

$$(\tilde{\mathrm{N}}_{T_1} R)(X,Y)Z = A_1(T_1)R(X,Y)Z. \quad (1.11)$$

The recurrent manifold is said to be symmetric, if

$A_1(T_1) = 0,$ in the above equation (1.12)

A manifold is said to be birecurrent, if

$$(\tilde{\mathrm{N}}_{T_1}\tilde{\mathrm{N}}_{T_2} R)(X,Y)Z = A(T_1,T_2)R(X,Y)Z. \quad (1.13)$$

It is said to be birecurrent symmetric, if

$A_2(T_1,T_2) = 0$ (1.14)

II. ϕ (123)- MULTIRECURRENCE AND SYMMETRY

Let P , a vector- valued tri-linear function, be any one of the curvature tensor K,W,C,L or V then we will define multirecurrent and symmetry as follows:**

Definition (2.1): A Hsu-structure manifold is said to be ϕ (123)-multirecurrent in P if

$$\alpha^r(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_2}\nabla_{T_1}P)(X,\phi Y)\phi Z$$

$$+(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_2}\nabla_{T_1}P)\left((\nabla_{T_n}\phi)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}P)(X,(\nabla_{T_n}\phi)Y)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}P)(X,\phi Y)(\nabla_{T_n}\phi)Z$$

+……

$$+(\nabla_{T_n}\dots\dots\dots\nabla_{T_2}P)\left((\nabla_{T_1}\phi)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots\nabla_{T_2}P)(X,(\nabla_{T_1}\phi)Y)\phi Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots\nabla_{T_2}P)(X,\phi Y)(\nabla_{T_1}\phi)Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_n}\phi)\phi X,(\nabla_{T_{n-1}}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_{n-1}}\phi)\phi X,(\nabla_{T_n}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_n}\phi)\phi X,\phi Y\right)(\nabla_{T_{n-1}}\phi)Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_{n-1}}\phi)\phi X,\phi Y\right)(\nabla_{T_n}\phi)Z$$

$$+\alpha^r\left(\nabla_{T_{n-2}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)(X,(\nabla_{T_n}\phi)Y)(\nabla_{T_{n-1}}\phi)Z$$

$$+\alpha^r\left(\nabla_{T_{n-2}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)(X,(\nabla_{T_{n-1}}\phi)Y)(\nabla_{T_n}\phi)Z$$

+…….....

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)\left((\nabla_{T_1}\phi)\phi X,(\nabla_{T_2}\phi)Y\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)\left((\nabla_{T_2}\phi)\phi X,(\nabla_{T_1}\phi)Y\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)\left((\nabla_{T_1}\phi)\phi X,\phi Y\right)(\nabla_{T_2}\phi)Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)\left((\nabla_{T_2}\phi)\phi X,\phi Y\right)(\nabla_{T_1}\phi)Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)(X,(\nabla_{T_1}\phi)Y)(\nabla_{T_2}\phi)Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)(X,(\nabla_{T_1}\phi)Z)(\nabla_{T_2}\phi)Y$$

$$+\left(\nabla_{T_{n-2}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-2}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)(X,(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Y)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-2}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)(X,\phi Y)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Z$$

+…….....

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)\left((\nabla_{T_2}\nabla_{T_1}\phi)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)(X,(\nabla_{T_2}\nabla_{T_1}\phi)Y)\phi Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots\ldots\nabla_{T_3}P\right)(X,\phi Y)(\nabla_{T_2}\nabla_{T_1}\phi)Z$$

$$+\left(\nabla_{T_{n-3}}\ldots\ldots\ldots\nabla_{T_1}P\right)\left((\nabla_{T_n}\phi)\phi X,(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Y\right)\phi Z$$

$$+\left(\nabla_{T_{n-3}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)\left((\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\phi X,(\nabla_{T_n}\phi)Y\right)\phi Z$$

$$+\left(\nabla_{T_{n-3}}\ldots\ldots\ldots\nabla_{T_1}P\right)\left((\nabla_{T_n}\phi)\phi X,\phi Y\right)(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Z$$

$$+\left(\nabla_{T_{n-3}}\ldots\ldots\ldots\ldots\nabla_{T_1}P\right)\left((\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\phi X,\phi Y\right)(\nabla_{T_n}\phi)Z$$

$$+\alpha^r\left(\nabla_{T_{n-3}}\ldots\ldots\ldots\nabla_{T_1}P\right)(X,(\nabla_{T_n}\phi)Y)(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Z$$

$+a^r(\nabla_{T_{n-3}} \dots \dots \dots \nabla_{T_1}P)(X,(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Y)(\nabla_{T_n}\phi)Z$

+……………………………………..........................

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)\left((\nabla_{T_1}\phi)\,\phi X,\ (\nabla_{T_3}\nabla_{T_2}\phi)Y\right)\phi Z$

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)\left((\nabla_{T_3}\nabla_{T_2}\phi)\,\phi X,(\nabla_{T_1}\phi)Y\right)\phi Z$

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)\left((\nabla_{T_1}\phi)\phi X,\phi Y\right)(\nabla_{T_3}\nabla_{T_2}\phi)Z$

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)\left((\nabla_{T_3}\nabla_{T_2}\phi)\,\phi X,\phi Y\right)(\nabla_{T_1}\phi)Z$

$+a^r(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)(X,(\nabla_{T_3}\nabla_{T_2}\phi)Y)(\nabla_{T_1}\phi)Z$

$+a^r(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)(X,(\nabla_{T_1}\phi)Y)(\nabla_{T_3}\nabla_{T_2}\phi)Z$

$+(\nabla_{T_{n-3}} \dots .. \nabla_{T_1}P)\left((\nabla_{T_n}\phi)\,\phi X,(\nabla_{T_{n-1}}\phi)Y\right)(\nabla_{T_{n-2}}\phi)Z$

+ …………………………………………......................

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)\left((\nabla_{T_1}\phi)\,\phi X,(\nabla_{T_2}\phi)Y\right)(\nabla_{T_3}\phi)Z$

$+(\nabla_{T_{n-3}} \dots .. \nabla_{T_1}P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\,\phi X,\phi Y\right)\phi Z$

$+a^r(\nabla_{T_{n-3}} \dots .. \nabla_{T_1}P)(X,(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Y)\phi Z$

$+a^r(\nabla_{T_{n-3}} \dots .. \nabla_{T_1}P)(X,\phi Y)(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Z$

+…………………………………………....................

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)\left((\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)\,\phi X,\phi Y\right)\phi Z$

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)(\phi X,(\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)Y)\phi Z$

$+(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_4}P)(\phi X,\phi Y)(\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)Z$

$+(\nabla_{T_{n-4}} \dots .. \nabla_{T_1}P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)\,\phi X,(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Y\right)\phi Z$

$+(\nabla_{T_{n-4}} \dots .. \nabla_{T_1}P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\,\phi X,(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Y\right)\phi Z$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)\phi X,\phi Y\right)(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\phi X,\phi Y\right)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Z$$

$$+\alpha^r(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)(X,(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Y)(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Z$$

$$+\alpha^r(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)(X,(\nabla_{T_{n-2}}\nabla_{n-3}\phi)Y)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Z$$

+…….....

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_2}\nabla_{T_1}\phi)\phi X,(\nabla_{T_4}\nabla_{T_3}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_4}\nabla_{T_3}\phi)\phi X,(\nabla_{T_2}\nabla_{T_1}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_2}\nabla_{T_1}\phi)\phi X,\phi Y\right)(\nabla_{T_4}\nabla_{T_3}\phi)Z$$

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_4}\nabla_{T_3}\phi)\phi X,\phi Y\right)(\nabla_{T_2}\nabla_{T_1}\phi)Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)(X,(\nabla_{T_2}\nabla_{T_1}\phi)Y)(\nabla_{T_4}\nabla_{T_3}\phi)Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)(X,(\nabla_{T_4}\nabla_{T_3}\phi)Y)(\nabla_{T_2}\nabla_{T_1}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots.\nabla_{T_1}P)\left((\nabla_{T_n}\phi)\phi X,(\nabla_{n-1}\phi)Y\right)(\nabla_{T_{n-2}}\nabla_{n-3}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots.\nabla_{T_1}P)\left((\nabla_{T_n}\phi)\phi X,(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Y\right)(\nabla_{T_{n-1}}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots.\nabla_{T_1}P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\phi X,(\nabla_{T_n}\phi)Y\right)(\nabla_{T_{n-1}}\phi)Z$$

+…….....

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_1}\phi)\phi X,(\nabla_{T_2}\phi)Y\right)(\nabla_{T_4}\nabla_{T_3}\phi)Z$$

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_1}\phi)\phi X,(\nabla_{T_4}\nabla_{T_3}\phi)Y\right)(\nabla_{T_2}\phi)Z$$

$$+(\nabla_{T_n}\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_4}\nabla_{T_3}\phi)\phi X,(\nabla_{T_1}\phi)Y\right)(\nabla_{T_2}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_n}\phi)\phi X,(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Y\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)\phi X,\left(\nabla_{T_n}\phi\right)Y\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_n}\phi\right)\phi X,\phi Y\right)\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)\phi X,\phi Y\right)\left(\nabla_{T_n}\phi\right)Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(X,\left(\nabla_{T_n}\phi\right)Y\right)\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(X,\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Y\right)\left(\nabla_{T_n}\phi\right)Z$$

+…….....

$$+\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(\left(\nabla_{T_1}\phi\right)\phi X,\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)Y\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)\phi X,\left(\nabla_{T_1}\phi\right)Y\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(\left(\nabla_{T_1}\phi\right)\phi X,\phi Y\right)\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)\phi X,\phi Y\right)\left(\nabla_{T_1}\phi\right)Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(X,\left(\nabla_{T_1}\phi\right)Y\right)\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(X,\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)Y\right)\left(\nabla_{T_1}\phi\right)Z$$

$$+\left(\nabla_{T_{n-4}}\ldots.\nabla_{T_1}P\right)\left(\left(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}}\ldots.\nabla_{T_1}P\right)\left(X,\left(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}}\ldots.\nabla_{T_1}P\right)\left(X,\phi Y\right)\left(\nabla_{T_n}\nabla_{T_{n-2}}\nabla_{T_{n-1}}\nabla_{T_{n-3}}\phi\right)Z$$

+……

$$+\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi\right)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(X,\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi\right)Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots..\nabla_{T_5}P\right)\left(X,\phi Y\right)\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi\right)Z$$

+ .....

+ .....

$$+(\nabla_{T_n}P)\left((\nabla_{T_{n-1}}\phi)\phi X,(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1})Y\right)\phi Z$$

$$+(\nabla_{T_n}P)\left((\nabla_{n-2}\dots\dots\dots\nabla_{T_1})\phi X,(\nabla_{T_{n-1}}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_n}P)\left((\nabla_{T_{n-1}}\phi)\phi X,\phi Y\right)(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1})Z$$

$$+(\nabla_{T_n}P)\left((\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)\phi X,\phi Y\right)(\nabla_{T_{n-1}}\phi)Z$$

$$+\alpha^r(\nabla_{T_n}P)(X,(\nabla_{T_{n-1}}\phi)Y)(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1})Z$$

$$+\alpha^r(\nabla_{T_n}P)(X,(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)Y)(\nabla_{n-1}\phi)Z$$

+…….....

$$+(\nabla_{T_1}P)\left((\nabla_{T_2}\phi)\phi X,(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_1}P)\left((\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)\phi X,(\nabla_{T_2}\phi)Y\right)\phi Z$$

$$+(\nabla_{T_1}P)\left((\nabla_{T_2}\phi)\phi X,\phi Y\right)(\nabla_{T_n}\dots\dots\dots\nabla_3\phi)z$$

$$+(\nabla_{T_1}P)\left((\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)\phi X,\phi Y\right)(\nabla_{T_2}\phi)Z$$

$$+\alpha^r(\nabla_{T_1}P)(X,(\nabla_{T_2}\phi)Y)(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)Z$$

$$+\alpha^r(\nabla_{T_1}P)(X,(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)Y)(\nabla_{T_2}\phi)Z$$

$$+P\left((\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)\phi X,\phi Y\right)\phi Z$$

$$+\alpha^r P(X,(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)Y)\phi Z$$

$$+\alpha^r P(X,\phi Y)(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)Z$$

$$=\alpha^r A_n(T_1,T_2,\dots\dots\dots\dots\dots T_n)P(X,\phi Y)\phi Z. \qquad (2.1)$$

Or

$$\alpha^r(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\dots\nabla_{T_2}\nabla_{T_1}P)(\phi X,Y)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\dots\dots\dots\dots\nabla_{T_2}\nabla_{T_1}P)\left((\nabla_{T_n}\phi)X,Y\right)\phi Z$$

$$+(\nabla_{T_{n-1}}\dots\dots\dots\dots\nabla_{T_1}P)(X,(\nabla_{T_n}\phi)\phi Y)\phi Z$$

$+a^r\left(\nabla_{T_{n-1}} \dots \dots \dots .. \nabla_{T_1} P\right)(\phi X, Y)\left(\nabla_{T_n} \phi\right) Z$

+…………………………………………........

$+a^r\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_2} P\right)\left(\left(\nabla_{T_1} \phi\right) X, Y\right) \phi Z$

$+\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_2} P\right)\left(X, \left(\nabla_{T_1} \phi\right) \phi Y\right) \phi Z$

$+a^r\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_2} P\right)(\phi X, Y)\left(\nabla_{T_1} \phi\right) Z$

$+\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\left(\nabla_{T_n} \phi\right) X, \left(\nabla_{T_{n-1}} \phi\right) \phi Y\right) \phi Z$

$+\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\left(\nabla_{T_{n-1}} \phi\right) X, \left(\nabla_{T_n} \phi\right) \phi Y\right) \phi Z$

$+a^r\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\left(\nabla_{T_n} \phi\right) X, Y\right)\left(\nabla_{T_{n-1}} \phi\right) Z$

$+a^r\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\left(\nabla_{T_{n-1}} \phi\right) X, Y\right)\left(\nabla_{T_n} \phi\right) Z$

$+\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\phi X, \left(\nabla_{T_n} \phi\right) \phi Y\right)\left(\nabla_{T_{n-1}} \phi\right) Z$

$+\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\phi X, \left(\nabla_{T_{n-1}} \phi\right) \phi Y\right)\left(\nabla_{T_n} \phi\right) Z$

+……………………………………….....................

$+\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_3} P\right)\left(\left(\nabla_{T_1} \phi\right) X, \left(\nabla_{T_2} \phi\right) \phi Y\right) \phi Z$

$+\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_3} P\right)\left(\left(\nabla_{T_2} \phi\right) X, \left(\nabla_{T_1} \phi\right) \phi Y\right) \phi Z$

$+a^r\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_3} P\right)\left(\left(\nabla_{T_1} \phi\right) X, Y\right)\left(\nabla_{T_2} \phi\right) Z$

$+a^r\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_3} P\right)\left(\left(\nabla_{T_2} \phi\right) X, Y\right)\left(\nabla_{T_1} \phi\right) Z$

$+\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_3} P\right)\left(\phi X, \left(\nabla_{T_1} \phi\right) \phi Y\right)\left(\nabla_{T_2} \phi\right) Z$

$+\left(\nabla_{T_n} \dots \dots \dots .. \nabla_{T_3} P\right)\left(\phi X, \left(\nabla_{T_2} \phi\right) \phi Y\right)\left(\nabla_{T_1} \phi\right) Z$

$+a^r\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\left(\nabla_{T_n} \nabla_{T_{n-1}} \phi\right) X, Y\right) \phi Z$

$+\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)\left(\phi X, \left(\nabla_{T_n} \nabla_{T_{n-1}} \phi\right) \phi Y\right) \phi Z$

$+a^r\left(\nabla_{T_{n-2}} \dots \dots \dots .. \nabla_{T_1} P\right)(\phi X, Y)\left(\nabla_{T_n} \nabla_{T_{n-1}} \phi\right) Z$

+…………………………………………..............

$+a^r\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P\right)\left((\nabla_{T_2}\nabla_{T_1}\phi)X,Y\right)\phi Z$

$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P\right)\left(\phi X,(\nabla_{T_2}\nabla_{T_1}\phi)\phi Y\right)\phi Z$

$+a^r\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P\right)(\phi X,Y)(\nabla_{T_2}\nabla_{T_1}\phi)Z$

$+\left(\nabla_{T_{n-3}}\dots\dots\dots\nabla_{T_1}P\right)\left((\nabla_{T_n}\phi)X,(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\phi Y\right)\phi Z$

$+\left(\nabla_{T_{n-3}}\dots\dots\dots\nabla_{T_1}P\right)\left((\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)X,(\nabla_{T_n}\phi)\phi Y\right)\phi Z$

$+a^r\left(\nabla_{T_{n-3}}\dots\dots\nabla_{T_1}P\right)\left((\nabla_{T_n}\phi)X,Y\right)(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Z$

$+a^r\left(\nabla_{T_{n-3}}\dots\dots\dots\nabla_{T_1}P\right)\left((\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)X,Y\right)(\nabla_{T_n}\phi)Z$

$+\left(\nabla_{T_{n-3}}\dots\dots\dots\nabla_{T_1}P\right)\left(\phi X,(\nabla_{T_n}\phi)\phi Y\right)(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Z$

$+\left(\nabla_{T_{n-3}}\dots\dots\dots\nabla_{T_1}P\right)\left(\phi X,(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\phi Y\right)(\nabla_{T_n}\phi)Z$

+…………………………………………................................

$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left((\nabla_{T_1}\phi)X,\ (\nabla_{T_3}\nabla_{T_2}\phi)\phi Y\right)\phi Z$

$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left((\nabla_{T_3}\nabla_{T_2}\phi)X,(\nabla_{T_1}\phi)\phi Y\right)\phi Z$

$+a^r\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left((\nabla_{T_1}\phi)X,Y\right)(\nabla_{T_3}\nabla_{T_2}\phi)Z$

$+a^r\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left((\nabla_{T_3}\nabla_{T_2}\phi)X,Y\right)(\nabla_{T_1}\phi)Z$

$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left(\phi X,(\nabla_{T_3}\nabla_{T_2}\phi)\phi Y\right)(\nabla_{T_1}\phi)Z$

$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left(\phi X,(\nabla_{T_1}\phi)\phi Y\right)(\nabla_{T_3}\nabla_{T_2}\phi)Z$

$+\left(\nabla_{T_{n-3}}\dots\nabla_{T_1}P\right)\left((\nabla_{T_n}\phi)X,(\nabla_{T_{n-1}}\phi)\phi Y\right)(\nabla_{T_{n-2}}\phi)Z$

+…………………………………………..........................

$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_4}P\right)\left((\nabla_{T_1}\phi)X,(\nabla_{T_2}\phi)\phi Y\right)(\nabla_{T_3}\phi)Z$

$+a^r\left(\nabla_{T_{n-3}}\dots\nabla_{T_1}P\right)\left((\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)X,Y\right)\phi Z$

$$+(\nabla_{T_{n-3}}\dots..\nabla_{T_1}P)(\phi X,(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\phi Y)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-3}}\dots..\nabla_{T_1}P)(\phi X,Y)(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Z$$

+…….....

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots..\nabla_{T_4}P)\left((\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)X,Y\right)\phi Z$$

$$+(\nabla_{T_n}\dots\dots\dots..\nabla_{T_4}P)(\phi X,(\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)\phi Y)\phi Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots..\nabla_{T_4}P)(\phi X,Y)(\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)X,(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\phi Y\right)\phi Z$$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)X,(\nabla_{T_n}\nabla_{T_{n-1}}\phi)\phi Y\right)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)X,Y\right)(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Z$$

$$+\alpha^r(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)X,Y\right)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)(\phi X,(\nabla_{T_n}\nabla_{T_{n-1}}\phi)\phi Y)(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots..\nabla_{T_1}P)(\phi X,(\nabla_{T_{n-2}}\nabla_{n-3}\phi)\phi Y)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)Z$$

+…….....

$$+(\nabla_{T_n}\dots\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_2}\nabla_{T_1}\phi)X,(\nabla_{T_4}\nabla_{T_3}\phi)\phi Y\right)\phi Z$$

$$+(\nabla_{T_n}\dots\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_4}\nabla_{T_3}\phi)X,(\nabla_{T_2}\nabla_{T_1}\phi)\phi Y\right)\phi Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_2}\nabla_{T_1}\phi)X,Y\right)(\nabla_{T_4}\nabla_{T_3}\phi)Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots..\nabla_{T_5}P)\left((\nabla_{T_4}\nabla_{T_3}\phi)X,Y\right)(\nabla_{T_2}\nabla_{T_1}\phi)Z$$

$$+(\nabla_{T_n}\dots\dots\dots..\nabla_{T_5}P)(\phi X,(\nabla_{T_2}\nabla_{T_1}\phi)Y)(\nabla_{T_4}\nabla_{T_3}\phi)Z$$

$$+(\nabla_{T_n}\dots\dots\dots..\nabla_{T_5}P)(\phi X,(\nabla_{T_4}\nabla_{T_3}\phi)\phi Y)(\nabla_{T_2}\nabla_{T_1}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots.\nabla_{T_1}P)\left((\nabla_{T_n}\phi)X,(\nabla_{n-1}\phi)\phi Y\right)(\nabla_{T_{n-2}}\nabla_{n-3}\phi)Z$$

$$+(\nabla_{T_{n-4}}\dots.\nabla_{T_1}P)\left((\nabla_{T_n}\phi)X,(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\phi Y\right)(\nabla_{T_{n-1}}\phi)Z$$

$+(\nabla_{T_{n-4}} \dots . \nabla_{T_1} P)\left((\nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)X, (\nabla_{T_n} \phi)\phi Y\right)(\nabla_{T_{n-1}} \phi)Z$

+……………………………………...............................

$+(\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)\left((\nabla_{T_1} \phi)X, (\nabla_{T_2} \phi)\phi Y\right)(\nabla_{T_4} \nabla_{T_3} \phi)Z$

$+(\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)\left((\nabla_{T_1} \phi)X, (\nabla_{T_4} \nabla_{T_3} \phi)\phi Y\right)(\nabla_{T_2} \phi)Z$

$+(\nabla_{T_{n-4}} \dots . . \nabla_{T_1} P)\left((\nabla_{T_n} \phi)X, (\nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)\phi Y\right)\phi Z$

$+(\nabla_{T_{n-4}} \dots . . \nabla_{T_1} P)\left((\nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)X, (\nabla_{T_n} \phi)\phi Y\right)\phi Z$

$+a^r (\nabla_{T_{n-4}} \dots . . \nabla_{T_1} P)\left((\nabla_{T_n} \phi)X, Y\right)(\nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)Z$

$+a^r (\nabla_{T_{n-4}} \dots . . \nabla_{T_1} P)\left((\nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)X, Y\right)(\nabla_{T_n} \phi)Z$

$+(\nabla_{T_{n-4}} \dots . . \nabla_{T_1} P)(\phi X, (\nabla_{T_n} \phi)\phi Y)(\nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)Z$

$+(\nabla_{T_{n-4}} \dots . . \nabla_{T_1} P)(\phi X, (\nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)\phi Y)(\nabla_{T_n} \phi)Z$

+……………………………………...........................

$+(\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)\left((\nabla_{T_1} \phi)X, (\nabla_{T_4} \nabla_{T_3} \nabla_{T_2} \phi)\phi Y\right)\phi Z$

$+(\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)\left((\nabla_{T_4} \nabla_{T_3} \nabla_{T_2} \phi)X, (\nabla_{T_1} \phi)\phi Y\right)\phi Z$

$+a^r (\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)\left((\nabla_{T_1} \phi)X, Y\right)(\nabla_{T_4} \nabla_{T_3} \nabla_{T_2} \phi)Z$

$+a^r (\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)\left((\nabla_{T_4} \nabla_{T_3} \nabla_{T_2} \phi)X, Y\right)(\nabla_{T_1} \phi)Z$

$+(\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)(\phi X, (\nabla_{T_1} \phi)\phi Y)(\nabla_{T_4} \nabla_{T_3} \nabla_{T_2} \phi)Z$

$+(\nabla_{T_n} \dots \dots \dots . . \nabla_{T_5} P)(\phi X, (\nabla_{T_4} \nabla_{T_3} \nabla_{T_2} \phi)\phi Y)(\nabla_{T_1} \phi)Z$

$+a^r (\nabla_{T_{n-4}} \dots . \nabla_{T_1} P)\left((\nabla_{T_n} \nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)X, Y\right)\phi Z$

$+(\nabla_{T_{n-4}} \dots . \nabla_{T_1} P)(\phi X, (\nabla_{T_n} \nabla_{T_{n-1}} \nabla_{T_{n-2}} \nabla_{T_{n-3}} \phi)\phi Y)\phi Z$

$+a^r (\nabla_{T_{n-4}} \dots . \nabla_{T_1} P)(\phi X, Y)(\nabla_{T_n} \nabla_{T_{n-2}} \nabla_{T_{n-1}} \nabla_{T_{n-3}} \phi)Z$

+……………………………………….

$$+\alpha^r\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_5}P\right)\left((\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)X,Y\right)\phi Z$$

$$+\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_5}P\right)(\phi X,(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)\phi Y)\phi Z$$

$$+\alpha^r\left(\nabla_{T_n}\dots\dots\dots\nabla_{T_5}P\right)(\phi X,Y)(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)Z$$

+ .....

+ .....

$$+\left(\nabla_{T_n}P\right)\left((\nabla_{T_{n-1}}\phi)X,(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)\phi Y\right)\phi Z$$

$$+\left(\nabla_{T_n}P\right)\left((\nabla_{n-2}\dots\dots\dots\nabla_{T_1}\phi)X,(\nabla_{T_{n-1}}\phi)\phi Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_n}P\right)\left((\nabla_{T_{n-1}}\phi)X,Y\right)(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)Z$$

$$+\alpha^r\left(\nabla_{T_n}P\right)\left((\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)X,Y\right)(\nabla_{T_{n-1}}\phi)Z$$

$$+\left(\nabla_{T_n}P\right)(\phi X,(\nabla_{T_{n-1}}\phi)\phi Y)(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)Z$$

$$+\left(\nabla_{T_n}P\right)(\phi X,(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}\phi)\phi Y)(\nabla_{n-1}\phi)Z$$

+.....

$$+\left(\nabla_{T_1}P\right)\left((\nabla_{T_2}\phi)X,(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)\phi Y\right)\phi Z$$

$$+\left(\nabla_{T_1}P\right)\left((\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)X,(\nabla_{T_2}\phi)\phi Y\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_1}P\right)\left((\nabla_{T_2}\phi)X,Y\right)(\nabla_{T_n}\dots\dots\dots\nabla_3\phi)z$$

$$+\alpha^r\left(\nabla_{T_1}P\right)\left((\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)X,Y\right)(\nabla_{T_2}\phi)Z$$

$$+\left(\nabla_{T_1}P\right)(\phi X,(\nabla_{T_2}\phi)\phi Y)(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)Z$$

$$+\left(\nabla_{T_1}P\right)(\phi X,(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)\phi Y)(\nabla_{T_2}\phi)Z$$

$$+\alpha^r P\left((\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)X,Y\right)\phi Z$$

$$+P(\phi X,(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)\phi Y)\phi Z$$

$$+\alpha^r P(\phi X,Y)(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)Z$$

$$= \alpha^r A_n(T_1, T_2, \dots\dots\dots\dots T_n)P(\phi X, Y)\phi Z. \qquad (2.2)$$

Or

$$\alpha^r(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_2}\nabla_{T_1}P)(\phi X, \phi Y)Z$$

$$+\alpha^r(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_2}\nabla_{T_1}P)\left((\nabla_{T_n}\phi)X, \phi Y\right)Z$$

$$+\alpha^r(\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}P)(\phi X, (\nabla_{T_n}\phi)Y)Z$$

$$+(\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}P)(\phi X, \phi Y)(\nabla_{T_n}\phi)\phi Z$$

+……..

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots\nabla_{T_2}P)\left((\nabla_{T_1}\phi)X, \phi Y\right)Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots\nabla_{T_2}P)(X, (\nabla_{T_1}\phi)Y)Z$$

$$+(\nabla_{T_n}\dots\dots\dots\nabla_{T_2}P)(\phi X, \phi Y)(\nabla_{T_1}\phi)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_n}\phi)X, (\nabla_{T_{n-1}}\phi)Y\right)Z$$

$$+\alpha^r(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_{n-1}}\phi)X, (\nabla_{T_n}\phi)Y\right)Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_n}\phi)X, \phi Y\right)(\nabla_{T_{n-1}}\phi)\phi Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)\left((\nabla_{T_{n-1}}\phi)X, \phi Y\right)(\nabla_{T_n}\phi)\phi Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)(\phi X, (\nabla_{T_n}\phi)Y)(\nabla_{T_{n-1}}\phi)\phi Z$$

$$+(\nabla_{T_{n-2}}\dots\dots\dots\nabla_{T_1}P)(\phi X, (\nabla_{T_{n-1}}\phi)Y)(\nabla_{T_n}\phi)\phi Z$$

+……..

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P)\left((\nabla_{T_1}\phi)X, (\nabla_{T_2}\phi)Y\right)Z$$

$$+\alpha^r(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P)\left((\nabla_{T_2}\phi)X, (\nabla_{T_1}\phi)Y\right)Z$$

$$+(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P)\left((\nabla_{T_1}\phi)X, \phi Y\right)(\nabla_{T_2}\phi)\phi Z$$

$$+(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}P)\left((\nabla_{T_2}\phi)X, \phi Y\right)(\nabla_{T_1}\phi)\phi Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3} P\right)\left(\phi X,\left(\nabla_{T_1}\phi\right)Y\right)\left(\nabla_{T_2}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3} P\right)\left(\phi X,\left(\nabla_{T_2}\phi\right)Y\right)\left(\nabla_{T_1}\phi\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1} P\right)\left(\left(\nabla_{T_n}\nabla_{T_{n-1}}\phi\right)X,\phi Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1} P\right)\left(\phi X,\left(\nabla_{T_n}\nabla_{T_{n-1}}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1} P\right)\left(\phi X,\phi Y\right)\left(\nabla_{T_n}\nabla_{T_{n-1}}\phi\right)\phi Z$$

+…….....

$$+\alpha^r\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3} P\right)\left(\left(\nabla_{T_2}\nabla_{T_1}\phi\right)X,\phi Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3} P\right)\left(\phi X,\left(\nabla_{T_2}\nabla_{T_1}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3} P\right)\left(\phi X,\phi Y\right)\left(\nabla_{T_2}\nabla_{T_1}\phi\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-3}} \dots\dots\dots \nabla_{T_1} P\right)\left(\left(\nabla_{T_n}\phi\right)X,\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi\right)Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_{n-3}} \dots\dots\dots \nabla_{T_1} P\right)\left(\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi\right)X,\left(\nabla_{T_n}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_{n-3}} \dots\dots\dots \nabla_{T_1} P\right)\left(\left(\nabla_{T_n}\phi\right)X,\phi Y\right)\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-3}} \dots\dots\dots \nabla_{T_1} P\right)\left(\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi\right)X,\phi Y\right)\left(\nabla_{T_n}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-3}} \dots\dots\dots \nabla_{T_1} P\right)\left(\phi X,\left(\nabla_{T_n}\phi\right)Y\right)\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-3}} \dots\dots\dots \nabla_{T_1} P\right)\left(\phi X,\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi\right)Y\right)\left(\nabla_{T_n}\phi\right)\phi Z$$

+…….....

$$+\alpha^r\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P\right)\left(\left(\nabla_{T_1}\phi\right)X,\ \left(\nabla_{T_3}\nabla_{T_2}\phi\right)Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P\right)\left(\left(\nabla_{T_3}\nabla_{T_2}\phi\right)X,\left(\nabla_{T_1}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P\right)\left(\left(\nabla_{T_1}\phi\right)X,\phi Y\right)\left(\nabla_{T_3}\nabla_{T_2}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P\right)\left(\left(\nabla_{T_3}\nabla_{T_2}\phi\right)X,\phi Y\right)\left(\nabla_{T_1}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P\right)\left(\phi X,\left(\nabla_{T_3}\nabla_{T_2}\phi\right)Y\right)\left(\nabla_{T_1}\phi\right)\phi Z$$

$$+(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P)(\phi X, (\nabla_{T_1}\phi)Y)(\nabla_{T_3}\nabla_{T_2}\phi)\phi Z$$

$$+(\nabla_{T_{n-3}} \dots \nabla_{T_1} P)\left((\nabla_{T_n}\phi)X, (\nabla_{T_{n-1}}\phi)Y\right)(\nabla_{T_{n-2}}\phi)\phi Z$$

+…….....

$$+(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P)\left((\nabla_{T_1}\phi)X, (\nabla_{T_2}\phi)Y\right)(\nabla_{T_3}\phi)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-3}} \dots \nabla_{T_1} P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)X, \phi Y\right)Z$$

$$+\alpha^r(\nabla_{T_{n-3}} \dots \nabla_{T_1} P)(\phi X, (\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)Y)Z$$

$$+(\nabla_{T_{n-3}} \dots \nabla_{T_1} P)(\phi X, \phi Y)(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\phi)\phi Z$$

+…….....

$$+\alpha^r(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P)\left((\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)X, \phi Y\right)Z$$

$$+\alpha^r(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P)(X, (\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)Y)Z$$

$$+(\nabla_{T_n} \dots\dots\dots \nabla_{T_4} P)(X, \phi Y)(\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi)\phi Z$$

$$+\alpha^r(\nabla_{T_{n-4}} \dots \nabla_{T_1} P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)X, (\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)Y\right)Z$$

$$+\alpha^r(\nabla_{T_{n-4}} \dots \nabla_{T_1} P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)X, (\nabla_{T_n}\nabla_{T_{n-1}}\phi)Y\right)Z$$

$$+(\nabla_{T_{n-4}} \dots \nabla_{T_1} P)\left((\nabla_{T_n}\nabla_{T_{n-1}}\phi)X, \phi Y\right)(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\phi Z$$

$$+(\nabla_{T_{n-4}} \dots \nabla_{T_1} P)\left((\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)X, \phi Y\right)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)\phi Z$$

$$+(\nabla_{T_{n-4}} \dots \nabla_{T_1} P)(\phi X, (\nabla_{T_n}\nabla_{T_{n-1}}\phi)Y)(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi)\phi Z$$

$$+(\nabla_{T_{n-4}} \dots \nabla_{T_1} P)(\phi X, (\nabla_{T_{n-2}}\nabla_{n-3}\phi)Y)(\nabla_{T_n}\nabla_{T_{n-1}}\phi)\phi Z$$

+…….....

$$+\alpha^r(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P)\left((\nabla_{T_2}\nabla_{T_1}\phi)X, (\nabla_{T_4}\nabla_{T_3}\phi)Y\right)Z$$

$$+\alpha^r(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P)\left((\nabla_{T_4}\nabla_{T_3}\phi)X, (\nabla_{T_2}\nabla_{T_1}\phi)Y\right)Z$$

$$+(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P)\left((\nabla_{T_2}\nabla_{T_1}\phi)X, \phi Y\right)(\nabla_{T_4}\nabla_{T_3}\phi)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\phi\right)X,\phi Y\right)\left(\nabla_{T_2}\nabla_{T_1}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\phi X,\left(\nabla_{T_2}\nabla_{T_1}\phi\right)Y\right)\left(\nabla_{T_4}\nabla_{T_3}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\phi X,\left(\nabla_{T_4}\nabla_{T_3}\phi\right)Y\right)\left(\nabla_{T_2}\nabla_{T_1}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots.\nabla_{T_1}P\right)\left(\left(\nabla_{T_n}\phi\right)X,\left(\nabla_{n-1}\phi\right)Y\right)\left(\nabla_{T_{n-2}}\nabla_{n-3}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots.\nabla_{T_1}P\right)\left(\left(\nabla_{T_n}\phi\right)X,\left(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Y\right)\left(\nabla_{T_{n-1}}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots.\nabla_{T_1}P\right)\left(\left(\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)X,\left(\nabla_{T_n}\phi\right)Y\right)\left(\nabla_{T_{n-1}}\phi\right)\phi Z$$

+.....

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_1}\phi\right)X,\left(\nabla_{T_2}\phi\right)Y\right)\left(\nabla_{T_4}\nabla_{T_3}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_1}\phi\right)X,\left(\nabla_{T_4}\nabla_{T_3}\phi\right)Y\right)\left(\nabla_{T_2}\phi\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_n}\phi\right)X,\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)X,\left(\nabla_{T_n}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_n}\phi\right)X,\phi Y\right)\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)X,\phi Y\right)\left(\nabla_{T_n}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\phi X,\left(\nabla_{T_n}\phi\right)Y\right)\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)\phi Z$$

$$+\left(\nabla_{T_{n-4}}\ldots..\nabla_{T_1}P\right)\left(\phi X,\left(\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Y\right)\left(\nabla_{T_n}\phi\right)\phi Z$$

+.....

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_1}\phi\right)X,\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)X,\left(\nabla_{T_1}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_1}\phi\right)X,\phi Y\right)\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n}\ldots\ldots\ldots\nabla_{T_5}P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)X,\phi Y\right)\left(\nabla_{T_1}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P\right)\left(\phi X, \left(\nabla_{T_1}\phi\right)\phi Y\right)\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P\right)\left(\phi X, \left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\phi\right)Y\right)\left(\nabla_{T_1}\phi\right)\phi Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}} \dots \nabla_{T_1} P\right)\left(\left(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)X, \phi Y\right) Z$$

$$+\alpha^r\left(\nabla_{T_{n-4}} \dots \nabla_{T_1} P\right)\left(\phi X, \left(\nabla_{T_n}\nabla_{T_{n-1}}\nabla_{T_{n-2}}\nabla_{T_{n-3}}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_{n-4}} \dots \nabla_{T_1} P\right)\left(\phi X, \phi Y\right)\left(\nabla_{T_n}\nabla_{T_{n-2}}\nabla_{T_{n-1}}\nabla_{T_{n-3}}\phi\right)\phi Z$$

+……

$$+\alpha^r\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P\right)\left(\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi\right)X, \phi Y\right) Z$$

$$+\alpha^r\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P\right)\left(\phi X, \left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi\right)\phi Y\right)Z$$

$$+\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_5} P\right)\left(\phi X, \phi Y\right)\left(\nabla_{T_4}\nabla_{T_3}\nabla_{T_2}\nabla_{T_1}\phi\right)\phi Z$$

+…….....

+ .....

$$+\alpha^r\left(\nabla_{T_n} P\right)\left(\left(\nabla_{T_{n-1}}\phi\right)X, \left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1}\right)Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_n} P\right)\left(\left(\nabla_{n-2} \dots\dots\dots \nabla_{T_1}\right)X, \left(\nabla_{T_{n-1}}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_n} P\right)\left(\left(\nabla_{T_{n-1}}\phi\right)X, \phi Y\right)\left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1}\right)\phi Z$$

$$+\left(\nabla_{T_n} P\right)\left(\left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1}\phi\right)X, \phi Y\right)\left(\nabla_{T_{n-1}}\phi\right)\phi Z$$

$$+\left(\nabla_{T_n} P\right)\left(\phi X, \left(\nabla_{T_{n-1}}\phi\right)Y\right)\left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1}\right)\phi Z$$

$$+\left(\nabla_{T_n} P\right)\left(\phi X, \left(\nabla_{T_{n-2}} \dots\dots\dots \nabla_{T_1}\phi\right)Y\right)\left(\nabla_{n-1}\phi\right)\phi Z$$

+…….....

$$+\alpha^r\left(\nabla_{T_1} P\right)\left(\left(\nabla_{T_2}\phi\right)X, \left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3}\phi\right)Y\right)Z$$

$$+\alpha^r\left(\nabla_{T_1} P\right)\left(\left(\nabla_{T_n} \dots\dots\dots \nabla_{T_3}\phi\right)X, \left(\nabla_{T_2}\phi\right)Y\right)Z$$

$$+\left(\nabla_{T_1} P\right)\left(\left(\nabla_{T_2}\phi\right)X, \phi Y\right)\left(\nabla_{T_n} \dots\dots\dots \nabla_3\phi\right)\phi Z$$

$$+(\nabla_{T_1}P)\left((\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)X,\phi Y\right)(\nabla_{T_2}\phi)\phi Z$$

$$+(\nabla_{T_1}P)(\phi X,(\nabla_{T_2}\phi)Y)(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)\phi Z$$

$$+(\nabla_{T_1}P)(\phi X,(\nabla_{T_n}\dots\dots\dots\nabla_{T_3}\phi)Y)(\nabla_{T_2}\phi)\phi Z$$

$$+\alpha^r P\left((\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)X,\phi Y\right)Z$$

$$+\alpha^r P(\phi X,(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)Y)Z$$

$$+P(\phi X,\phi Y)(\nabla_{T_n}\nabla_{T_{n-1}}\dots\dots\dots\nabla_{T_1}\phi)\phi Z$$

$$=\alpha^r A_n(T_1,T_2,\dots\dots\dots\dots\dots T_n)P(\phi X,\phi Y)Z. \qquad (2.3)$$

Note: The total number of terms on the left hand side of the equation is

$${}^nC_0 3^0+{}^nC_1 3^1+{}^nC_2 3^2+\dots\dots\dots\dots\dots+{}^nC_r 3^r+\dots\dots\dots\dots+{}^nC_n 3^n=4^n$$

Where ${}^nC_r$, n is the total number of $\tilde{N}$ operators operated on the tensor P and structure f and r=0,1,2,…….,{n-1},n is the number of $\tilde{N}$'s is associated with the structure f only.

Definition (2.2): A ϕ (123)-multirecurrent Hsu-structure manifold is said to be P-symmetric or Ricci-symmetric if

$$A_n(T_1,T_2,\dots\dots\dots\dots\dots\dots,T_n)=0 \qquad (2.4)$$

REFERENCES

[1] C.J. Hsu, "On some structure which are similar to quaternion structure", *Tohoku Math. J.*, (12), 1960, pp. 403-428.

[2] Q. Khan, "On recurrent Riemannian manifolds", *Kuungpook Math. J.*, (44), 2004, pp. 269-276.

[3] R.S. Mishra, S.B. Pandey and U. Sharma "Pseudo n-Recurrent Manifold with KH-Structure", *Indian J. pure applied Mathematics*, 7(10), 1974, pp. 1277-1287.

[4] R.S. Mishra, "A course in tensors with applications to Riemannian Geometry", II edition Pothishala Pvt., Ltd. Allahabad (1973).

[5] R.S. Mishra, "Structure on a differentiable manifold and their applications", Chandrama Prakashan, 50-A, Balrampur House, Allahabad (1984).

[6] Lata Bisht,On GF- Structure Manifold , Multirecurrent and symmetry . Ultra Scientist Vol. 23 (2) B, 423-440, (2011).

[7] H.singh and Q.Khan, On generalized recurrent Riemanian manifolds, Publ. Math. De-brecen 56, 87-95(2000).

[8] K.Arslan, U.C. De, C. Murathan and A. Yildiz, On generalized recurrent Riemannian manifolds, Acta Math, Hungar., 123, 27-39 (2009).

[9] Jay Prakash Singh, On m-Projective Recurrent Riemannian Manifold, Int. Journal of Math. Analysis,Vol.6,no.24,1173-1178,(2012).

[10] K. Takano; On a space with birecurrent curvature tensor. Tensor. N.S. 22, 329-332, (1971).

DETECTION AND MONITORING OF BRIDGE HEALTH STATUS USING WSN AND ARDUINO

Manoranjani.S [1], Lavanya.S [2], Lakshmi Priya.A [3]

[1, 2, 3] *U.G Students, Department of Electronics and Communication Engineering,*

Raja College of Engineering and Technology, Madurai, Tamilnadu, (India)

ABSTRACT

This paper focuses on preemption system for infrastructure like bridges. The stability of the structure is directly related to the structural strength. The structural analysis of such a bridge can be analyzed using vibration pattern and comprehensive study. MEMS based accelerometer is used to analyze the vibration patterns. The vibration signal measured by the node is communicated over WSN. Three levels of vibrations are detected. If the vibrations are in Safe level no action is taken. If the vibrations are in warning level a signal is sent to the monitoring person through RF. If the vibrations are in critical level then the bridge is closed for traffic. The inspection of building structures especially bridge structures is currently made by visual inspection. Structural health monitoring is a field that relies on different methodologies to develop procedures that characterize the dynamic properties of physical structures to identify possible deteriorations of their behaviors. The few non visual methodologies make use of wired sensor networks, which are relatively expensive, vulnerable to damage, and time consuming to install. Systems based on wireless sensor networks should be both cost efficient and easy to install, scalable and adaptive to different type of structures. Acoustic emission techniques are an additional monitoring method to investigate the status of a bridge of its components. Micro-Electro-Mechanical-Systems (MEMS) and hybrid sensors form the heart of network nodes.

Keywords: Transmitter Section, Receiver Section, MEMS, ARDUINO

I. INTRODUCTION

Existing monitoring systems use traditional wired sensors technologies and several other devices that are time consuming to install and relatively expensive as compare to value of the structure. Typically they are using a large number of sensors (i.e. more than ten) which are connected through long cables and will therefore be installed only on few structures. A wireless monitoring system with MEMS sensors could reduce cost significantly. MEMS are small integrated devices or systems combining electrical and mechanical components that could be produced for 50 euro each.

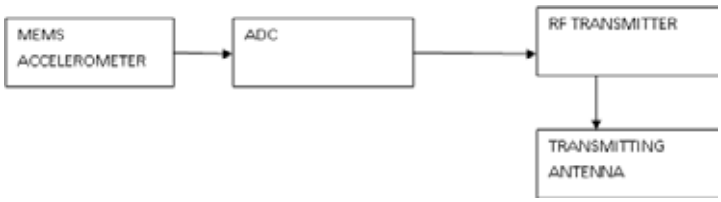
II. EXISTING SYSTEM

Now days the structural stability of the bridge is monitored by manually and also the traffic control of heavy duty vehicles over the bridge are also done manually. Rather the traditional structure monitoring system can be done using wired technology. Manual control of bridge leads to wastage of manpower and will not be effective during the time of calamities. Also wired system is too expensive, power hungry and difficult to implement and maintain. In this method, the same pattern of the bridge has been designed with the same composition of

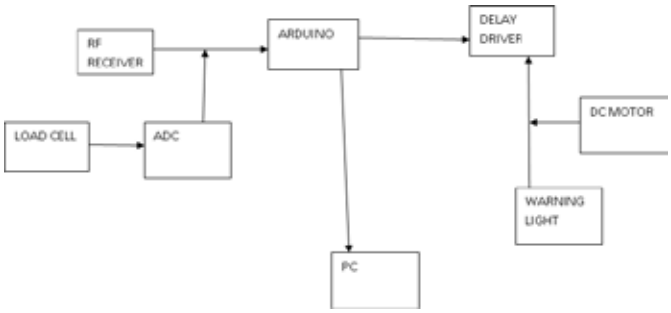
material as that the one used in the bridge. And by performing several tests like destructive testing on the bridge model, the strength and the life time of the bridge has been calculated. Ultrasonic C-scan imaging is done by sending out ultrasonic waves into a material. The reflections of these waves are read by a transducer and sent to a computer program. This program processes the data and creates a two-dimensional map of the bridge components.

III. PROPOSED WORK

3.1 Transmitter Section



3.2 Receiver Section



A practical solution to overcome this problem is by using wireless sensor network with MEMS accelerometer which is the integration of mechanical elements, sensors and etc., It is used to measure the vibration from dynamic load and also parameters such as strain, acceleration and angular displacement with strain. The wireless technology here we are using will be RF through which constant monitoring through pc is made easy and effective.

MEMS technologies are well suited to improve the performance, size, and cost of sensing systems. MEMS can be used in both monitoring and testing of transportation infrastructure systems. Several applications of MEMS in bridge engineering field are reported. Differential settlement between bridges and pavements causes bumps or uneven joints at the bridge ends. When vehicles, especially heavy trucks, approach and leave bridges, the bumps cause large impact loads to the bridge and pavements.

The pre stress forces can also be adjusted to deal with cracking issues in both positive and negative moment zones. With the combined application of the smart bearings and smart stands, the bridge can adjust its internal force distribution and mobilized each element to adopt itself to different environmental loads. Wireless monitoring system with MEMS sensors could reduce installation and maintenance cost dramatically.

The devices would performs sensing and signal interpretation, and would report their findings remotely. The concept is to build an ultrasonic flaw detection system on a chip using a MEMS device as a receiver array with, a mm scale piezoelectric element as a ultrasonic source. The system is intended to scavenge power from structural strains and to report results with fly-by polling using radio frequency communications. The concept

requires the development of phased array signal processing, and signature analysis signal processing, to perform flaw detection (flaw imaging) from the fixed location of a resident transducer.

The power of the Arduino is not its ability to crunch code, but rather its ability to interact with the outside world through its input-output (I/O) pins. The Arduino has 14 digital I/O pins labeled 0 to 13 that can be used to turn motors and lights on and off and read the state of switches. Micro-electromechanical systems (MEMS) are Freescale's enabling technology for acceleration and pressure sensors. MEMS-based sensor products provide an interface that can sense, process and/or control the surrounding environment.

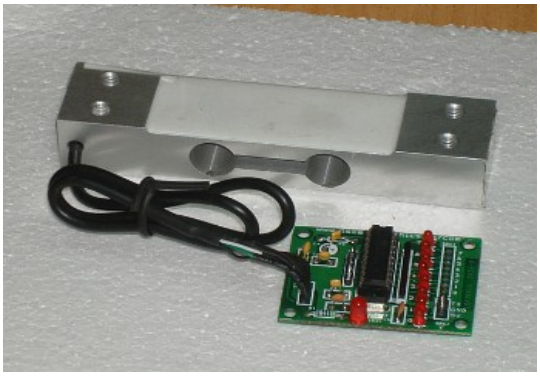


Figure 1: Load Cell

In addition, the environment in which the MEMS devices has to operate and the possible effect of the environment on the performance of the MEMS device has to be assessed. MEMS device against damage from installation or construction procedures as well as from contact with materials is paramount. Furthermore, there is the need to carry out extensive experimentation to ascertain the reliability and consistency over time of the information obtained from the embedded devices. The impacts of the infrastructure system dynamics on the embedded device have to be evaluated and vice versa.

IV. CONCLUSION

In this paper, an attempt is made to provide a general overview of application of MEMS and nano technologies for civil engineering and transportation. The synthesis provides information on current and potential applications, especially in bridge structures. Several case studies in the literatures demonstrate that MEMS technology has the potential to offer significant benefits to the civil engineering and transportation field. Finally the challenges in the application of MEMS technology into transportation infrastructure systems are summarized.

REFERENCES

[1] Carvalho, F., and Labuz, J.F. 2002. Moment Tensors of Acoustic Emission in Shear Faulting Under Plane-Strain Compression. Tectonophysics, 356, pp. 199-211.

[2] Glaser, S.D., Shoureshi, R., and Pescovitz, D.2005. Future Sensing Systems, Smart Structures & Systems, 1(1), 103 - 120.

[3] Kato, M. and Shimada, S. (1986) “Vibration of PC Bridge During Failure Process.” Journal of Structural Engineering ASCE 112(7) p. 1692-1703.

[4] Aluru, N.R. (1999). A reproducing kernel particle method for mesh less analysis of micro-electro-mechanical-systems. Computational Mechanics, 23: 324–338.

[5] Attoh-Okine, N.O. (2001). Potential applications of micro-electro-mechanical-systems (MEMS) in the management of infrastructure assets. Fifth International Conference on Managing Pavements, Seattle, Wash.

[6] Huff, M. (2002). MEMS fabrication. Sensor Review, 22(1): 18–33.

[7] Jain, A., Greve, D. and Oppenheim, J. (2002). A MEMS transducer for ultrasonic flow detection. ISARC, pp. 375–386.

COMPARATIVE STUDY FOR CURRENT, DELAY & POWER DISSIPATION OF CMOS INVERTERS

Kumod Kumar Gupta[1], Payal Garg[2]

[1]*Assistant Professor, Raj Kumar Goel Engineering College, Ghaziabad U.P, (India)*

[2]*M.Tech scholar, Al-Falah School of Engineering & Technology, Faridabad (India)*

ABSTRACT

We present a new model for predicting the delay, current & power in a CMOS Inverter. These delays, current & power are three major issues in design & synthesis of VLSI circuits, which depends on many other design parameter. In this paper we have used Tanner technology which deals with channel length in the order of 25 nm or even less. And simulation results are taken for different technology (32nm, 45nm) with the help of Tanner (T- spice) simulation tool. The propagation delay time determine the input to output signal delay during the high to low and low to high transitions of the output, respectively.

Keywords: VLSI, CMOS Inverter, Tanner (T- spice) simulation tool.

I. INTRODUCTION

Scaling of MOS transistor is concerned with systematic reduction of overall dimensions of the devices as allowed by the available technology. The operational characteristics of MOS transistors are changed with the reduction of its dimensions. The interconnection layers play a very important role in determining the performance of deep sub-micron technologies makes it essential to develop analysis tool that take these RC effects into account. The effect of RC interconnection trees can be efficiently incorporated into fast and accurate analysis tools. many authors choose to use a π- model along with a nonlinear macro-model of the inverter to find the delay & the output voltage waveform of the driving inverter. There are lots of researches on delay model for CMOS inverter and the most popular delay model is described in nth power law [9] where velocity saturation is main consideration. The effect of the resistance of interconnection wires can be thought of as a "shielding" of the overall capacitive load seen by the inverter. As a result, the delay through the inverter is reduced, and the output voltage and supply current waveform no longer resemble those of an inverter with a lumped capacitive load. These two effects, combined create the need to model CMOS gates with RC loads.

II. BACKGROUND

Based on nth Power law, delay model of CMOS inverter includes for both very fast as well as very slow input signal. The critical input transition time t_{T0}.

$$t_{TO} = \frac{C_O V_{DD}}{2I_{DO}} \frac{(n+1)(1-v_T)^n}{(1-v_T)^{n-1} - (v_V - v_T)^{n-1}} \quad (1)$$

Where $v_T = V_{TO}/V_{DD}$ and $v_v = V_{INV}/V_{DD}$. Then the delay t_d the delay from 0.5 V_{DD} of input to 0.5 V_{DD} of output, and the effective output, and the effective output transition time t_{TOUT} can be expressed as follows. t_{TOUT} can be used as t_T for the next logic gate;

($t_T \leq tT_O$:for the faster input)

$$t_d = t_T \left\{ \frac{1}{2} - \frac{1-v_T}{n+1} + \frac{(v_V - v_T)^{n-1}}{(n+1)(1-v_T)^n} \right\} + \frac{1}{2} \frac{C_O V_{DD}}{I_{DD}} \quad (2)$$

$$t_{OUT} = \left\{ \frac{C_O V_{DD}}{.7 I_{DD}} \frac{4 v_{DD}^2}{(4 v_{DD} - 1)} \right\} \quad (3)$$

$t_T > t_{TO}$ for the slower input

$$t_d = t_T \left[v_T - \frac{1}{2} \left\{ (v_V - v_T)^{n+1} \frac{(n+1)(1-v_T)^n}{2 t_T I_{DD}/C_O V_{DD}} \right\}^{\frac{1}{n+1}} \right] \quad (4)$$

$$t_{OUT} = \left\{ \frac{C_O V_{DD}}{.7 I_{DD}} \left(\frac{1-v_T}{t_d/t_T + 1/2 - v_T} \right)^n \right\} \quad (5)$$

Where C_O is an output capacitance, $v_{DO} = V_{DO}/V_{DD}$ and n=velocity saturation index.

2.1 Power Modelling of CMOS Inverter

Three types of power dissipation occur in CMOS inverter circuits, which are given below:

A. Leakage power Dissipation: In off- state, the main components of leakage currents are sub- threshold leakage (I_{sub}), gate induced drain leakage (I_{GIDI}), gate tunnelling leakage (I_{GATE}) and band-to-band tunnelling ($_{BTBT}$).

B. Short-circuit power: From the- power law the short circuit power dissipation model is

$$P_{sht\_da\_pw} = V_{DD}\, t_T I_{DO} \frac{1}{\alpha+1} \frac{1}{2^{\alpha-1}} \frac{(1-2v_T)^{\alpha+1}}{(1-v_T)^{\alpha}} \quad (6)$$

Where $v_T = \frac{V_{TH}}{V_{DD}}$

C. Dynamic power or switching power:

This type of power dissipation occurs due to the charging and discharging of load and parasitic capacitors.

$$P_{dynamic} = \alpha.\, C_L\, V_{DD}^2 .\, f + \sum_{i=1}^{n} \alpha_i . C_i . V_{DD} . (V_{DD} - V_T) \quad (7)$$

Dynamic power expression indicate that the average dynamic power of a complex gate due to the output load capacitance.

Where, CL=load capacitance, VDD = Supply Voltage, f = operating clock frequency and $\alpha \rightarrow$ switching activity of gate (the probability of a 0-1 switch in a cycle.

III. THE PROPOSED MODEL

The proposed model is divided into two main parts. The first deals with estimating an effective capacitance that captures the output voltage and the supply current up to tm, the time of the occurrence of the current peak. The second deals with obtaining the rest of the voltage and current waveforms. In the interest of brevity, only a falling input transition will be considered.

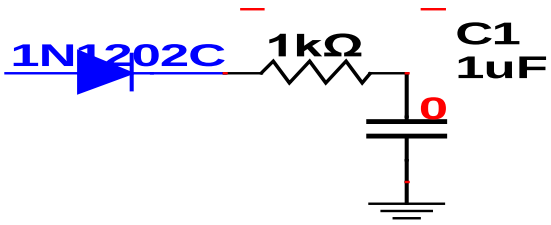


Fig-1 Inverter Circuit

3.1 The Effective Capacitance

An effective capacitance for a given RC load can be obtained by equating, over a certain time interval, the average of the current that flows into a lumped capacitive load to the average current that flows into the actual RC load. Since, as can be seen from fig-(2). the use of an effective lumped capacitive load , C_{eff}, in place of actual one yields good agreement in the inverter output voltage waveform only up to t_m, the time when the supply current peak occurs, we define the time interval for the current averaging as[0, t_m]. The PMOS device leaves the saturation region, can be used as a very accurate and efficient method for determining t_{st} was presented.

So the value of C_{eff} is given by the solution of

$$\frac{1}{t_m}\int_0^{t_m} C_{eff}\frac{dv_c}{dt}dt = \frac{1}{t_m}\int_0^{t_m} C_L\frac{dv_3}{dt}dt \qquad (8)$$

Where v_c is the inverter output across C_{eff} and v_3 is as defined in figure-(1). Usually [2], an analytical expression for v_c (which has to be equal to v_2 of fig-1) is assumed. Then v_3, then voltage at the output of RC lump is expressed in term of this v_c and v_3 are substituted into (8) to get an expression for C_{eff}. However, in this paper, we note that while v_2, in fig-(1), assumes a shape that is different from the normal shape of CMOS transitions, the voltage at the output of RC load v_3, does have the more normal shape of CMOS transitions, the voltage at the output of RC load,v_3, does have the more normal shape. If this waveform, or a good approximation to it, was known, then the inverter output voltage v_2, could be determined, at least up to $t=t_m$ by replacing C_L in fig-(1) with a voltage source producing this approximation to v_3.This is demonstrated in fig-(2) where the actual waveform of v_2 & v_3 are compared with the result of replacing a voltage

across C_L by a piecewise linear approximation to the true v_3, and determining the resulting v_2. Such an approach leads to a very simple approximation for C_{eff}.

The piecewise linear approximation to v_3 is obtained by passing a straight line through the 20% and 80% points of the actual waveform respectively, at times t_{20} and t_{80}. It is therefore given by

$$v_3(t)= \begin{cases} 0 & t < t_{start} \\ K(t - t_{start}) & t_{start} \leq t < t_{stop} \\ V_{DD} & t_{stop} \end{cases} \quad (9)$$

Where

$$K = \frac{.6V_{DD}}{t_{80} - t_{20}} \quad (10)$$

$$t_{start} = t_{20} - \frac{.2V_{DD}}{K} \quad (11)$$

$$t_{stop} = t_{80} + \frac{.2V_{DD}}{K} \quad (12)$$

t_{20} and t_{80} are obtained from empirical equations with coefficients that are determined by pre-simulating the circuit in fig-(1) with load size and input voltage slopes.

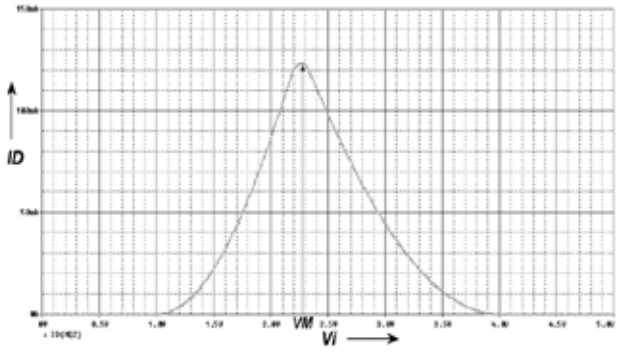


Fig-2 Input voltage v/s current waveform for an inverter

Details on these empirical models are presented in the next section. Here, sufficient it to say that it is easy to obtain the values of t_{20} & t_{80} using a very small set of coefficients that are determined only by the technology. So, having obtained an analytical description of v_3 as a function of time, the right hand side of (8) is now easy to evaluate.

In order to evaluate the left hand side of (8), we note that the shape of v_c (for v_2) has the normal CMOS form up to tm. Hence, we approximate v_c by a straight line passing through t_{start} and t_m. That is,

$$v_c(t)= \begin{cases} 0 & t \leq t_{satrt} \\ \frac{v_c(t_m)}{(t_m - t_{start})}(t - t_{start}) & t_{satrt} < t \leq t_m \end{cases} \quad (13)$$

Finally, substituting (13) into (8), we get the following expression for C_{eff}

$$C_{eff} = \frac{v_3(t_m)}{v_c(t_m)} C_L \quad (14)$$

As can be seen, then piece-wise linear approximation for C_{eff}. Also, notice that, for an inverter with a capacitive load, t_m and $v_c(t_m)$ can be computed using the model in [1]. Thus, the following iterative algorithm is used to solve for C_{eff}.

1. Set $C_{eff}=C_L$.
2. Use model in [1] and the current value of C_{eff} to compute t_m. Combine this t_m with any delay model to get $v_c(t_m)$. Compute $v_3(t_m)$ using (9): finally, compute a new value of C_{eff} using (14).
3. If the new C_{eff} is different from the old one by more than some threshold (say 5%) , then , set C_{eff} to the new value and go back to 2. Otherwise, terminate the iteration.

   At the end of this procedure, which usually converges in three or four steps, the model in [1] or any other suitable inverter model can be used with C_{eff} to get quantities like delay and supply current peak.

3.2. Empirical Parameter Extraction

In this section, we present the method for determining t_{20} and t_{80}. It should be noted that the same set of equations, with different coefficients, applies to both t_{20} and t_{80} and also for both charging and discharging, Now, for a given reference inverter and RC load, it can be shown that t_{20} (t_{80}) depends on T_i, the input transition time, in a linear manner. Thus, we can write

$$t_{20} = a(R_L,C_L) + b(R_L,C_L) \cdot T_i \quad (15)$$

In a similar fashion, It can be shown experimentally that, for a given R_L,$a(R_L,C_L)$ depends linearly on C_L. In other words,

$$a(R_L,C_L) = c(R_L) + d(R_L).C_L \quad (16)$$

Where $c(R_L)$ can be sufficiently approximated by a constant that depends only on the technology and the reference inverter of choice, and $d(R_L)$ can be very well approximated by another linear function of the form

$$d(R_L) = e + f.R_L \quad (17)$$

As can be seen, the model, so far, requires three empirical coefficients: c, e and f. As for the slope term in (15), a similar analysis leads to the following expression for b in terms of R_L and C_L.

$$b(R_L,C_L) = \min[g+h(R_L).C_L, k + l.R_L] \quad (18)$$

Where

$$h(R_L) = i+j.R_L \quad (19)$$

As can be seen, for a given inverter, equation (15)-(19) completely specify $t_{20}(t_{80})$ for any load and any input slope. So, in principle, for a general model, a lookup table of the empirical coefficients should be produced in order to account for the dependence of $t_{20}(t_{80})$ on the inverter size. However, since the effective capacitance that we are trying to compute is really a model of the admittance seen looking into RC load, its value should not be affected by the characteristics of the driver. Consequently, for the purposes of computing C_{eff}, we only need to look at a single reference inverter. Once this C_{eff} is determined, and as will be described in the next section, obtaining the final output voltage waveform and supply current pulse of actual circuit under test is a trivial matter.

IV. SIMULATIONRESULTS & DISCUSSIONS

All simulations are done in Cadence tool using the PTM technology of level 54 [6] 32 nm, 45 nm, 65 nm and 90 nm. Model parameters are extracted from BSIM4.6.1 user manual [7 & 8].

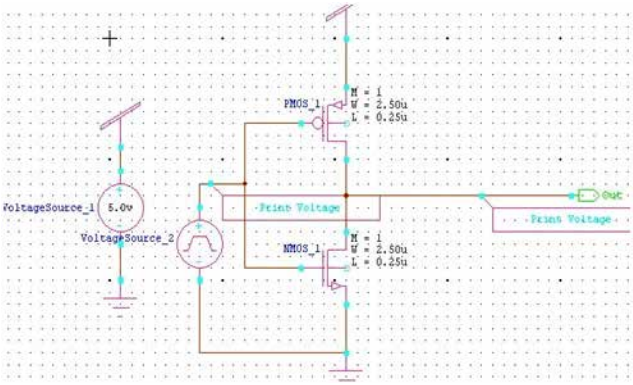


Fig-3 Schematic Diagram of CMOS Inverter

The fig-(3) shows the snapshot of CMOS inverter which is taken from Cadence Simulation tool. The results clearly show how power dissipation & delay relate to the different design parameters like supply voltage, load capacitance, width of PMOS & NMOS transistor. In all the cases the pulse type {PULSE (0V 1V 0NS 10NS 100NS 250NS)} input signal is used with rise time (tr=10ns), fall time (tf=10 ns) and time period (tper=250ns). We compare our results in 32 nm, 45nm, 65nm and 90nm for USDM CMOS inverter.

Since the output waveform expression for each of the regions of operation is known, propagation delay can be calculated as the time from 50 % of the rising/falling input to 50 % of the falling / rising of the output waveform.

Transient response of CMOS inverter is showing in fig-(4) i.e. input and output waveform w.r.t times in ns.

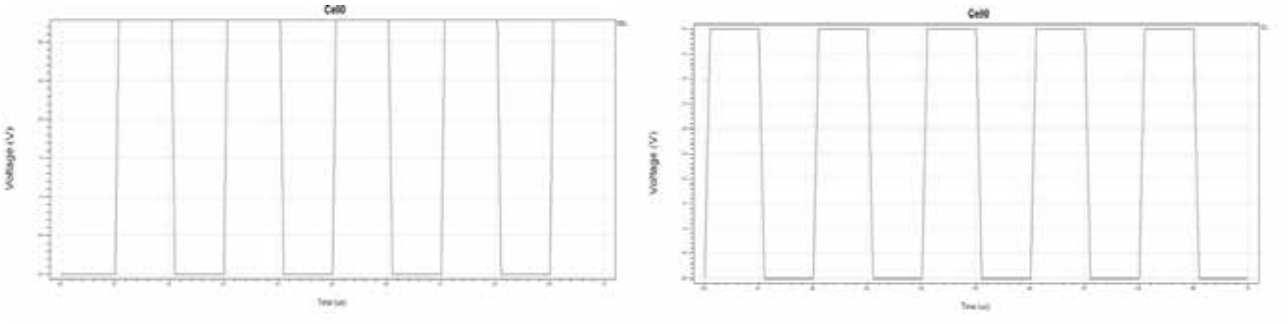


Fig- 4 Transient Characteristics of CMOS Inverter

The table shows that variation of supply voltage V_{dd} how delay, current and power dissipation will change for different technology. Power dissipation (Pd) will decrease by decreasing supply voltage. In case of propagation delay (tp) is inversely proportional to the supply voltage. From this table we also calculate the power delay product.

Table-1: **Delay, power dissipation and PDP related with supply voltage (CL=10 fF, Wp=0.5µm, Wn=0.25µm.)**

V_{dd}	32 nm			45 µm		
	PD(µW)	t_P(µs)	PDP(fJ)	PD(µW)	t_P(µs)	PDP(fj)
0.5	0.02	116.07	2.68	0.02	4.60	0.10
1	1.56	0.13	0.21	0.98	0.19	0.19
1.5	119.85	0.11	13.63	8.66	0.13	1.14
2	183.19	0.08	14.19	44.90	0.03	2.97
2.5	930.46	0.03	25.82	436.40	0.01	6.02

Table- 2: **Inverter Output and supply current**

$R_L(\Omega)$	C_L(pF)	Current Peak (mA)	80% Delay (nsec)
90	1.5	12.90	1.087
230	3.8	12.49	1.393
74	2.2	16.78	1.145
70	2	12.52	1.407

V. SUMMARY & CONCLUSION

In this paper, basically compare inverter in UDSM range and also how these are related with different design parameters. From the simulation it has been seen that when channel length less than or equal to 10 nm then output waveform are not appeared due to in this rang size of the channel length below atomic width. This is the limitation of

nano range. Supply voltage, current and power delay product are shown in the table form. This will help to proper characterize and analyze of the CMOS inverter in the nano range.

REFERENCES

[1]. A. Nabavi-Lishi, N. Rumin, "Delay and bus current Evaluation in CMOS Logic Circuits." Proc. IEEE Int. Conf Computer- Aided Design , pp. 189-203,1992

[2]. J. Qian, S. Pullela, & L. Pillage, "Modeling the 'effective Capacitance' for the RC Interconnect of CMOS gates,, IEEE Trans. Computer Aided Design, Vol. 13, pp. 15261535, Dec. 1994.

[3]. j. kong, d. Overthauser, "Combining RC – interconnect Effects with NonlinearMOS Macromodels," Proc. IEEE Int. Symp. Circuits & Systems, pp. 570=573, 1995.

[4]. F. Dartu, n. Menezes, & L. pilleggi, "Performance Computation for Precharacterized CMOS Gates with RC Loads." IEEE Trans. Computer-Aided Design, Vol. 15, pp. 544-553, May 1996.

[5]. F. Rouatbi, B. Haroun, & A. Al-Khalili, "Power estimation Tool for Sub-Micron CMOS VLSI Circuits," Proc. IEEE lnt. Conf. Computer-Aided Design,pp. 204-209, 1992.

[6]. ASU, Berkeley Predictive Technology Model (BPTM) Dept. of EE, Arizona State Univ., Tempe, AZ, 2006[7] W. Liu et al., BSIM3v3.2.2 MOSFET Model – User Manual, Department of Electrical and Computer Engineering, university of California, Berkeley, 1999.

[7]. X. Xi et al., BSIM4.6.1 MOSFET Model – User Manual, Department of Electrical and Computer Engineering, university of California, Berkeley, 2007.

[8]. T. Sakurai. And R. Newton. "A simple short- channel MOSFET model and its application to delay analysis of Inverters and series connected MOSFETs," Proceedings of IEEE International Conference on Circuits and Systems, pp. 105-108. 1990.

CONGESTION AVOIDANCE DETECTION AND ALLEVIATION PROTOCOL FOR DELAY TOLERANT NETWORKS

S.Kannadhasan [1], Suresh.R [2]

[1] *Research Scholar, Department of Electronics and Communication Engineering, Raja College of Engineering and Technology, Madurai, Tamilnadu, (India)*

[2] *Research Scholar, Department of Electronics and Communication Engineering, Government Polytechnic College, Madurai, Tamilnadu, (India)*

ABSTRACT

Congestion in wireless WSN not only causes severe information loss but also leads to excessive energy consumption. To address this problem, a novel scheme for congestion avoidance detection and alleviation (CADA) in WSNs is proposed. By exploiting data characteristics, a small number of representative nodes are chosen from those in the event area as data sources, so that the source traffic can be suppressed proactively to avoid potential congestion. Once congestion occurs inevitably due to traffic mergence, it will be detected in a timely way by the hotspot node based on a combination of buffer occupancy and channel utilization. Congestion is then alleviated reactively by either dynamic traffic multiplexing or source rate regulation in accordance with the specific hotspot scenarios.

Keywords: CCS, CODA, LACAGS, BGR

I.INTRODUCTION

Many congestion control schemes have been proposed for WSNs. These works involve complicated computation for resource allocation and utilization, which renders their application impractical for resource-constrained WSNs. Meanwhile, they did not take the data fidelity requirements into consideration when performing the control operations.In recent years, several new congestion control solutions have been studied for WSNs. Among them, some prior work focuses on traffic reduction for congestion control in the WSN.

WSN composed of a large number of sensor nodes and a set of sinks in an interested area is considered in this work. Sensor nodes are randomly distributed in the area and remain stationary after deployment. All of them have similar capabilities and equal significance. They are constrained in memory space, processing capability, communication bandwidth and energy storage. Sensor nodes and the sink communicate via bidirectional multi-hop wireless. Sinks are uniformly scattered across the sensing field. They have more powerful resources to perform data gathering and network management tasks. For the same reason as described in all nodes are assumed to implement a carrier sense multiple access (CSMA)-like medium access control (MAC) protocol for data transmission. According to some preset rules (such as event signal type and event spot location), the source node can assign a corresponding priority to its packets based on the importance of the event to be reported. Traffic flows initiated will remain active long enough to contribute to changes in the network load.

II. RELATED WORK

Cooperative communication, which exploits the broadcast nature of wireless communication to enhance network performance, especially to improve energy efficiency. Without additional transmissions, nodes in the neighborhood of a sender can obtain a copy of the forwarded packet through overhearing. Cooperative communication enables these nodes with a copy to cooperate with the sender for the relaying task. Although many previous cooperative schemes have exhibited satisfactory effectiveness in improving network performance, most of them cannot be directly applied to the wireless sensor network due to their requirements on more powerful or special designed radio hardware.

Energy-Efficient Cooperative Communication (EECC), scheme for the sensor network to provide reliable and efficient transmission against unreliable wireless links. In EECC, cooperative relay is performed at each intermediate hop between source and sink. When a node fails to receive a data packet from its upstream sender node, nearby nodes which have successfully overheard the packet will start the cooperation proactively and select the “best” relay out of them to participate in the transmission. The node cooperation is implemented by a cross-layer design between the network and Medium Access Control (MAC) layers: the cooperative relay node is elected through the MAC layer acknowledgement contention from relevant candidate node sets, which are formed through partial routing information broadcasts. The theoretical analysis shows that EECC outperforms the non-cooperative mechanisms in terms of energy consumption in presence of transmission failures. Extensive simulation results confirm that EECC significantly improves data transmission performance.

III. PROPOSED METHOD

3.1 Congestion Control Schemes

Many congestion control schemes have been proposed for WSNs. These works involve complicated computation for resource allocation and utilization, which renders their application impractical for resource-constrained WSNs. Meanwhile, they did not take the data fidelity requirements into consideration when performing the control operations. In recent years, several new congestion control solutions have been studied for WSNs. Among them, some prior work focuses on traffic reduction for congestion control in the WSN.

3.1.1 Congestion detection and avoidance (CODA)

This is the first detailed investigation on congestion control in WSNs, which combines local backpressure techniques and centralized sink-to sensors notifications but is not specifically concerned with different classes of traffic flows.

3.1.2 Congestion Control and Fairness (CCF)

This controls congestion in a hop-by-hop manner and each node adjusts rate based on its available service rate and child node number. The rate adjustment in CCF relies only on packet service time, which could lead to low utilization when some nodes do not have enough traffic or the packet error rate is high.

3.1.3. Event-To-Sink Reliable Transport (ESRT)

The sink is required to regulate the source reporting rate in an undifferentiated manner by broadcasting control messages to all source nodes. The underlying assumption is that a sink can reach all nodes via a high-energy one-hop broadcast, which is not practical for a large-scale sensor network.

3.1.4 Congestion Control for Sensor Networks (CONCERT)

It uses the adaptive data aggregation technique to reduce the amount of information travelling throughout the network. This leverages a unique characteristic of WSNs to handle the congestion problem.

3.1.5 Interference-Minimized Multipath Routing (I2MR)

It integrates source rate adaptation and multipath routing for congestion control in WSNs.

3.1.6 Learning Automate Based Congestion Avoidance Scheme (LACAS)

It avoids congestion by using a learning automata based approach. It adaptively makes the data packet arrival rate in the nodes equal to the rate, so that the occurrence of congestion in the node can be seamlessly avoided. It must be noted that pure traffic reduction could impose a negative impact on data fidelity.

Apart from the schemes based on traffic control, there have been attempts to explore other mechanisms for congestion alleviation in WSNs. Siphon proposes to add multi-radio virtual sinks to the network as a means of dealing with congestion. When congestion occurs, Siphon redirects traffic off the primary low-power radio network and onto the overlay network with long-range radio. The cost for adding and using the long-range radio is not negligible.

3.1.7 Biased Geographical Routing (BGR)

This is a geographic routing that reactively split traffic during congestion. The bias, which determines how far the trajectory of splitting traffic will deviate from the original path, is randomly chosen and could make congestion worse under some situations.

3.1.8 Topology-Aware Resource Adaption (TARA)

It adopts different traffic multiplexing strategies depending on specific topologies. It requires knowledge about the local and the end-to-end topology for capacity estimation, which causes too high an overhead for a large-scale network.

3.1.9 Congestion Aware Routing (CAR)

It adopts to use a priority aware routing protocol with data prioritization to alleviate congestion. Congestion zones are dedicated for high priority data while other data can only be routed out of the congestion zones. Multiple sinks are required to be deployed on the area border for gathering data with different priorities.

IV. SIMULATION AND RESULTS

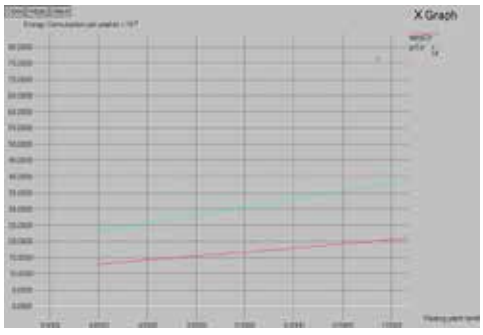


Figure 1. Comparison on Energy Consumption Per Packet

The decrease in retransmissions also results in the energy saving for packet transmission, as shown in Fig 1. These results justify the application of EECC in large-scale sensor networks with energy constrained sensor nodes and unreliable wireless links.

EECC significantly reduces end-to-end packet delay as compared to the pure retransmission method. An examination of ns-2 simulation traces reveals that this is attributed to a considerable decrease in the number of packet retransmissions: without EECC, the intended sender will time-out and resend the lost packet. The delay caused by such time-outs is far longer than that introduced by the new NAV setting in EECC.

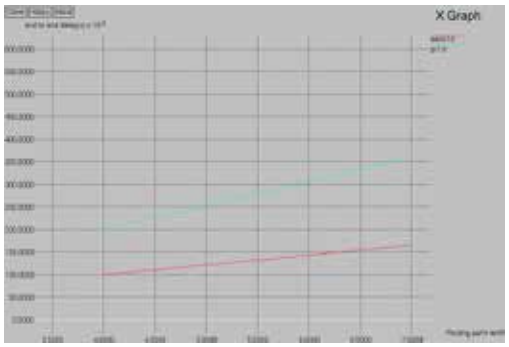


Figure 2. Comparison on end-to-end packet delay

V. CONCLUSION AND FUTURE WORK

In this project, we proposed an energy-efficient cooperative communication scheme (EECC) for unreliable sensor networks. This new scheme takes advantage of cooperative transmission to enhance the routing robustness against link unreliability. A best co-operator is elected from qualified neighbors of the relay node on the routing path to participate in the data transmission. In this way, EECC can reduce the total number of transmission times in the network. Through analysis and experiments, we validate that EECC is capable to improve data transmission efficiency in the sensor network with unreliable wireless links.

As future work, we would study how to extend EECC to operate in the presence of node mobility or network congestion. The scheme may need to be modified so that cooperative node sets could be updated in time to compensate for frequent changes of the link quality.

REFERENCES

[1]J. Zhao, R. Govindan. "Understanding packet delivery performance in dense wireless sensor networks," Proceedings of 1st International Conferenceon Embedded Networked Sensor Systems, pp. 1-13, 2003.

[2]. S. Kim, R. Fonseca, D. Culler. "Reliable transfer on wireless sensor networks," Proceedings of 1st International Conference on Sensor and Ad HocCommunications and Networks, pp. 449-459, 2004.

[3]. Q. Cao, T. Abdelzaher, T. He, R. Kravets."Cluster-based forwarding for reliable end-to-end delivery in wireless sensor networks," Proceedings of 25th Annual IEEE Conference on Computer Communications, pp. 1928-1936, 2007.

[4]. W. Fang, J. Chen, L. Shu, T. Chu, D. Qian. "Congestion avoidance, detection and alleviation in wireless sensor networks," Springer Journalof Zhejiang University Science C, Vol. 11, No. 1, pp. 63-73, 2010.

[5]. G. J. Pottie, W. J. Kaiser. "Wireless integrated network sensors," Communications of the ACM, vol. 43, no. 5, pp. 51-58, 2000.

[6]. G. Anastasi, M. Conti, A. Falchi, E. Gregori, A. Passarella. "Performance measurements of mote sensor networks," Proceedings of 7th ACM International Symposium on Modeling, Analysis and Simulation ofWireless and Mobile Systems, pp. 174-181, 2004.

[7]. S. Cui, A. J. Goldsmith, A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO in sensor network," IEEE Journal on Selected Areasin Communications, Vol. 22, No. 6, pp. 1089-1098, 2004.

[8]. S. Hussain, A. Azim, J. H. Park. "Energy efficient virtual MIMO communication for wireless sensor networks," SpringerTelecommunication Systems, Vol. 42, No. 1-2, pp. 139-149, 2009.

[9]. M. Z. Zamalloa, K. Seada, B. Krishnamachari, "Efficient geographic routing over lossy links in wireless sensor networks," ACM Transactionson Sensor Networks, Vol. 4, No. 3, Article 12, 33 pages, 2008.

[10] J. Wang, H. Zhai, W. Liu, Y. Fang, "Reliable and efficient packet forwarding by utilizing path diversity in wireless ad hoc networks," Proceedingsof IEEE Military Communications Conference 2004, pp. 258-264, 2004.

[11]. D. Puccinelli, M. Haenggi. "Reliable data delivery in large-scale lowpower sensor networks," ACM Transactions on Sensor Networks, Vol. 6, No. 4, Article 28, 41 pages, 2010.

[12]. A. Miu, H. Balakrishnan, C. E. Koksal, "Improving loss resilience with multi-radio diversity in wireless networks," Proceedings of 11th AnnualInternational Conference on Mobile Computing and Networking, pp. 16- 30, 2005.

SCALAR MULTIPLICATION ALGORITHMS AND APPLICATIONS OF ELLIPTIC CURVE

Indu Bala Thingom

Department of Information Technology, N.E.H.U., Shillong (India)

ABSTRACT

Cryptosystems based on elliptic curves become more and more popular. The security of these cryptosystems is based on the intractability of the discrete logarithm problem on elliptic curves, since no sub-exponential attack is known for a general elliptic curve over a finite field. With a much shorter key length, they offer the same level of security as other public key cryptosystems such as RSA. Efficient implementation of scalar multiplication is crucial for elliptic curve cryptographic systems. We revisit some recently proposed scalar multiplication algorithms and study some of the attacks that can be done on them. Next we study some of the applications of elliptic curve.

Keywords*: *Elliptic Curve Discrete Logarithmic Problem, Double-And-Add, Montgomery's Ladder, Naf, Scalar Multiplication, Side Channel Attacks, Sliding Window.

I INTRODUCTION

An elliptic curve over a field K is a non-singular cubic curve in two variables, f(x,y) = 0 with a rational point (which may be a point at infinity).

Elliptic curve arithmetic is defined on the basis of the operation of the field it depends on, and the operational efficiency is the key, thus efficient realization of elliptic curve arithmetic is a crucial problem. Fast realization of the elliptic curve relates to a variety of factors, research shows that scalar multiplication operation speed has a great influence on the realization of the elliptic curve speed (K. Zhang, T. Yan, 2013).

Scalar multiplication multiplies some point on an elliptic curve by some (usually secret) scalar. If P is a point on an EC and d is an integer, the operation computing the d-fold of P, namely the point dP, is called scalar multiplication. When such an operation is implemented on an embedded system such as a smart card, it is subjected to side channel attacks.

The strategies used for enhancement of efficiency are: (1) efficient group arithmetic in the elliptic curve group, (2) using a sparser representation for the scalar, (3) use of precomputation to precompute some points required later (4) using efficient algorithms like sliding window method, NAF or use of efficient addition chains, like Montgomery's ladder etc (P. K. Mishra, V. Dimitrov, 2007).

The advantage of using elliptic curve groups is: there is no known subexponential algorithm to solve the Elliptic Curve Discrete Logarithmic Problem (ECDLP). This means that a desired security level can be achieved with a much smaller key size in comparison to other public key schemes.

II ELLIPTIC CURVES OVER Z_p

Let p > 3 be prime. The elliptic curve $E_p(a, b)$ over Z_p is the set of solutions $(x, y) \in Z_p \times Z_p$ to the congruence

$$y^2 \bmod p = (x^3 + ax + b) \bmod p \quad (1)$$

where $a, b \in Z_p$ are constants such that $(4a^3 + 27b^2) \bmod p \neq 0 \bmod p$ together with a special point *O* called the point at infinity.

The addition operation on $E_p(a, b)$ is defined as follows:

Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are points on $E_p(a, b)$. If $x_2 = x_1$ and $y_2 = -y_1$, then $P + Q = O$; otherwise $P + Q = (x_3, y_3)$, where

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p \quad (2)$$

$$y_3 = (\lambda (x_1 - x_3) - y_1) \bmod p, \quad (3)$$

and

$$\lambda = \begin{cases} \frac{(y_2 - y_1)}{(x_2 - x_1)} \bmod p, & if\ P \neq Q \\ \frac{(3x_1^2 + a)}{2y_1} \bmod p, & if\ P = Q \end{cases} \quad (4)$$

The multiplication operation on $E_p(a, b)$ is defined as repeated addition. So 2P = P + P

III ELLIPTIC CURVES OVER GF(2^m)

The form of cubic equation for elliptic curves over GF(2^m) is

$$y^2 + xy = x^3 + ax^2 + b \quad (5)$$

where the variables x and y and the coefficients a and b are elements of GF(2^m).

The addition operation on $E_2^m (a, b)$ is defined as follows:

Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are points on $E_2^m (a, b)$. If $x_2 = x_1$ and $y_2 = -y_1$, then $P + Q = O$; otherwise $P + Q = (x_3, y_3)$, where

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \quad (6)$$

$$y_3 = \lambda (x_1 + x_3) + x_3 + y_1, \quad (7)$$

and

$$\lambda = \frac{y_2 + y_1}{x_2 + x_1} \quad (8)$$

Multiplication is defined as $R = 2P = (x_3, y_3)$, where

$$x_3 = \lambda^2 + \lambda + a \quad (9)$$

$$y_3 = x_1^2 + (\lambda + 1x)_3, \quad (10)$$

and

$$\lambda = x_1 + \frac{y_1}{x_1} \quad (11)$$

IV COMPUTING POINT MULTIPLES ON ELLIPTIC CURVES

Consider the equation Q = kP where Q, P ϵ $E_p(a, b)$ and k < p. It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P. This is called the discrete logarithmic problem for elliptic curves.

In an elliptic curve setting, where the group operation is written additively, we would compute a multiple aP of an elliptic curve point P using DOUBLE-AND-ADD algorithm (A. Byrne et al., 2007) (M. Hedabou, P. Pinel, L. Bénéteau, 2004) (M. Joye, 2003) (Y. Yarom and N. Benger, 2014), in which the squaring operation P^2 would be replaced by the doubling operation 2P, and the multiplication of two group elements would be replaced by the addition of two elliptic curve points (D. R. Stinson, 2006).

Algorithm: DOUBLE-AND-ADD(P,(d_m, ...,d_0))

The algorithm works as follows:

To compute dP, we start with the binary representation for d:

$d = d_0 + 2d_1 + 2^2d_2 + ... + 2^md_m$, where $[d_0, ..., d_m] \epsilon \{0,1\}$

Q = 0

for i = m down to 0 do

Q = 2Q

if d_i = 1 then

Q = Q + P

return Q

This algorithm requires $\log_2(n)$ iterations of point doubling and addition to compute the full-point multiplication. Such an algorithm would suffer from power analysis attacks when doubling and addition operations are distinguishable (D. R. Stinson, 2006).

The addition operation on an elliptic curve has the property that additive inverses are very easy to compute. This fact can be exploited in a generalization of the DOUBLE-AND-ADD algorithm, which is called the DOUBLE-AND-(ADD OR SUBTRACT) algorithm (D. R. Stinson, 2006). This technique is described as follows:

Let c be an integer. A signed binary representation of c is an equation of the form

$$c = \sum_{i=0}^{l-1} c_i 2^i \qquad (12)$$

where $c_i \epsilon$ -1, 0, 1 for all i.

Let P be a point of order n on an elliptic curve. Given any signed binary representation (c_{l-1}, ..., c_0) of an integer c, where $0 \leq c \leq n-1$, it is possible to compute the multiple cP of the elliptic curve point P by a series of doublings, additions and subtractions.

Algorithm: DOUBLE-AND-(ADD OR SUBTRACT) (P, (c_{l-1}, ..., c_0))

Q ß 0

for i ß *l*-1 down to 0

do{

Q ß 2Q

if c_i = 1

then Q ß Q + P

else if $c_i = -1$

then Q ß Q – P}

return (Q)

The subtraction operation Q - P would be performed by first computing the additive inverse -P of P, and then adding the result to Q.

A signed binary representation $(c_{l-1}, ..., c_0)$ of an integer c is said to be in non-adjacent form provided that no two consecutive c_i's are non-zero. Such a representation is denoted as a NAF representation (D. R. Stinson, 2006). The basis of this transformation is to replace substrings of the form (0, 1, ..., 1, 1) in the binary representation by (1, 0, ... ,0, -1). Substitutions of this type do not change the value of c, due to the identity $2^i + 2^{i-1} + ... + 2^j = 2^{i+1} - 2^j$, where i > j. This process is repeated as often as needed, starting with the rightmost bits and proceeding to the left.

Every non-negative integer has a NAF representation and that the NAF representation of an integer is unique. On average, a NAF representation contains more zeroes than the traditional binary representation of a positive integer. An *l*-bit integer contains *l*/2 zeroes in its binary representation and 2*l*/3 zeroes in its NAF representation (Md. R. Islam, Md. S. Hasan, I. S. M. Asaduzzaman, 2008).

These results make it easy to compare the average efficiency of the DOUBLE-AND-ADD algorithm using a binary representation to the DOUBLE-AND-(ADD OR SUBTRACT) algorithm using the NAF representation. Each algorithm requires *l* doublings, but the number of additions (or subtractions) is *l*/2 in the first case and *l*/3 in the second case (M. Rivain, 2011) (Md. R. Islam, Md. S. Hasan, I. S. M. Asaduzzaman, 2008) (V. Dimitrov, L. Imbert, P. K. Mishra, 2005). If it is assumed that a doubling takes roughly the same amount of time as an addition (or subtraction), then the ratio of the average times required by the two algorithms is approximately

$$\frac{l + \frac{l}{2}}{l + \frac{l}{3}} = \frac{9}{8} \qquad (13)$$

Therefore a roughly 11% speedup is obtained, on average, by this simple technique (D. R. Stinson, 2006).

Algorithm: Windowed version of DOUBLE-AND-ADD algorithm

In this algorithm (Y. Yarom, N. Benger, 2014), one selects a window size w and computes all 2^w values of dP for d = 0, 1, 2, ..., 2^w-1. The algorithm now uses the representation $d = d_0 + 2^w d_1 + 2^{2w}d_2 + ... + 2^{mw}d_m$ and becomes

Q = 0

for i = 0 to m do

Q = 2^wQ

if $d_i > 0$ then

Q = Q + d_iP

return Q

This algorithm has the same complexity as the DOUBLE-AND-ADD approach with the benefit of using fewer point additions. Typically, the value of w is chosen to be fairly small making the pre-computation stage a trivial

component of the algorithm. The entire complexity for a n-bit number is measured as n+1 point doubles and $2^w - 2 + \frac{n}{w}$ point additions.

Algorithm: Sliding - window method

In the sliding window (Y. Yarom, N. Benger, 2014) version of the DOUBLE-AND-ADD algorithm, point additions are trade off for point doubles. Here the points dP are computed for d = 2^{w-1}, 2^{w-1} + 1, ..., 2^w – 1. Effectively, only the values are computed for which the most significant bit of the window is set. The algorithm then uses the original DOUBLE-AND-ADD representation of d = $d_0 + 2d_1 + 2^2d_2 + ... + 2^m d_m$.

Q = 0
for i = 0 to m do
    Q = 2^wQ
    if d_i = 0 then
        Q = 2Q
    else
        Grab up to w-1 additional bits from d to store into (including d_i) t and decrement i suitably.
    if fewer than w bits were grabbed
        Perform DOUBLE-AND-ADD using t
        return Q
    else
        Q = 2^wQ
        Q = Q + tP
return Q

This algorithm has the benefit that the precomputation stage is roughly half as complex as the normal windowed method while also trading slower point additions for point doublings. It requires w – 1 + n point doubles and at most $2^{w-1} - 1 + \frac{n}{w}$ point additions.

Algorithm: Montgomery ladder

The Montgomery ladder (R. Afreen and S.C. Mehrotra, 2011) (Y. Yarom, N. Benger, 2014) approach computes the point multiplication in a fixed amount of time. This can be beneficial when timing or power consumption measurements are exposed to an attacker performing a side channel attack. The algorithm uses the same representation as from DOUBLE-AND-ADD.

R_0 = 0, R_1 = P
for i = 0 to m do
    if d_i = 0 then
        $R_1 = R_0 + R_1$
        $R_0 = 2R_0$
    else
        $R_0 = R_0 + R_1$
        $R_1 = 2R_1$
return R_0

This algorithm is in effect the same speed as the DOUBLE-AND-ADD approach except that it computes the same number of point additions and doubles regardless of the value of the multiplicand d. This means that at this level the algorithm does not leak any information through timing or power.

However, it was shown by (Y. Yarom, N. Benger, 2014) that through application of FLUSH+RELOAD side channel attack the full private key can be revealed in only one multiplication operation.

In general, finding the optimal addition chain for a given exponent is a hard problem, for which no efficient algorithms are known, so optimal chains are typically only used for small exponents. However, there are a number of heuristic algorithms that, while not being optimal, have fewer multiplications than exponentiation by squaring at the cost of additional bookkeeping work and memory usage. Regardless, the number of multiplications never grows more slowly than $\Theta(\log n)$, so these algorithms only improve asymptotically upon exponentiation by squaring by a constant factor at best.

V SIDE CHANNEL ATTACK

The basic algorithm for computing Q = dP on an elliptic curve is based on doubling and adding operations.

A side channel attack (SCA) called Simple Power Analysis (SPA) (K. Zhang and T. Yan, 2013) (M. Hedabou, P. Pinel and L. Bénéteau, 2004) involves monitoring the power consumption of a single execution of a cryptographic algorithm (M. Hedabou, P. Pinel and L. Bénéteau, 2004). Every instruction has different power consumption; therefore it is possible to retrieve the sequence of instructions during the algorithm execution. For example, the DOUBLE-AND-ADD algorithm has two primary operations, point addition and point doubling. Each of these operations produce a different power trace when executed because of the different number of multiplications and additions in each algorithm. Since, the execution of a point addition in the DOUBLE-AND-ADD is directly related to the secret key, it is possible to retrieve the secret key by monitoring the power consumption of a single execution of a scalar multiplication (A. Byrne et al., 2007) (M. Joye, 2003).

SPA attacks work well on algorithms where the power consumption can be directly related to the instruction being executed. In order to resist SPA attacks, the instructions executed in a cryptographic algorithm must not be directly related to the secret data. In the DOUBLE-AND-ADD method, the branch instruction based on d_i leaks information about the secret key. A simple solution would be to execute a point doubling for every bit of d but this vastly increases the execution time of the algorithm. For this we can make use of special Addition Chains to perform a point multiplication using only point additions. In this way, SPA cannot be used to determine the secret key (A. Byrne et al., 2007).

Improved point multiplication algorithms such as the sliding window method will obscure d to some degree, but plenty of information may still be revealed. To prevent side channel attacks, we can incorporate in the implementation some countermeasures intended to make the processing time of the algorithm independent from the data. These standard countermeasures usually consist of performing some dummy operations and using data randomization to obtain a resistant algorithm against SCA attacks. The inconvenient of this method is that it penalizes the running time (M. Hedabou, P. Pinel and L. Bénéteau, 2004). (P. Y. Liardet, N. P. Smart, 2001) have proposed to reduce information leakage by using a special point representation in some elliptic curves pertaining to a particular category, such that a single formula can be used for adding and doubling operations.

When a FLUSH+RELOAD attack was done, around 95% of the ephemeral private key was recovered when the Montgomery ladder was used for the scalar multiplication step. Until then, Montgomery ladder was not considered to be vulnerable to side-channel attacks. The full ephemeral private key was then recovered at very

small cost using a Baby-Step-Giant-Step (BSGS) algorithm. Knowledge of the ephemeral private key leads to recovery of the signer's private key, thus fully breaking the ECDSA implementation using only one signature (N. Benger et al., 2014) (Y. Yarom, K. Falkner, 2013).

One issue hindering the extension of the attack to implementations using the sliding window method for scalar multiplications instead of the Montgomery ladder is that only a lower proportion of the bits of the ephemeral private key can be recovered so the BSGS reconstruction becomes infeasible (N. Benger et al., 2014).

VI EUCLID'S ADDITION CHAIN

An addition chain is a finite sequence of integers $(v_0, ..., v_s)$ satisfying $\forall k \leq s$, $v_k = v_i + v_j$ for some $i, j < k$. A Euclid's addition chain is an addition chain which satisfies $v_1 = 1$, $v_2 = 2$, $v = = v_2 + v_1$ and $\forall 3 \leq i \leq s\text{-}1$, if $v_i = v_{i-1} + v_j$ for some $j < i\text{-}1$, then $v_{i+1} = v_i + v_{i-1}$ (case 1) or $v_{i+1} = v_i + v_j$ (case 2). Case 1 is called the Fibonacci step (it corresponds to one step of the Fibonacci sequence) and case 2 is called a small step (The smaller of the two possible numbers are added to v_i). Finding such chains is quite simple, it suffices to choose an integer k' co prime with k and apply the subtractive form of Euclid's algorithm.

The main advantage of these chains is that they're only made of additions (no doublings) which make them resistant against simple channel analysis. Up to now Euclid's chain was mainly used with curves in Montgomery form (on which the addition of two points P and Q can be performed in only 4 multiplications and 2 squaring if the difference P - Q is known), however not every curve can be transformed into Montgomery form. The drawback of Euclid's chains is the fact that it is not easy to find small ones (A. Byrne et al., 2007).

VII APPLICATIONS OF ELLIPTIC CURVE

Encryption/decryption:

Several approaches to encryption/decryption using elliptic curves have been analyzed. One of them is given here (G.V.S. Raju, R. Akbani, 2003):

The first task in this system is to encode the plaintext message m to be sent as an x-y point P_m. It is the point P_m that will be encrypted as a cipher text and subsequently decrypted. As with the key exchange system, an encryption/decryption system requires a point G and an elliptic group $E_p(a, b)$ as parameters. Each user A selects a private key n_A and generates a public key

$$P_A = n_A * G \tag{14}$$

To encrypt and send a message P_m to B, A chooses a random positive integer x and produces the cipher text C_m consisting of the pair of points

$$C_m = \{xG, P_m + xP_B\} \tag{15}$$

A has used B's public key P_B. To decrypt the cipher text, B multiplies the first point in the pair by B's secret key and subtracts the result from the second point:

$$P_m + xP_B - n_B(xG) = P_m + x_BG) - n_B(xG) = P_m \tag{16}$$

A has masked the message P_m by adding xP_B to it. Nobody but A knows the value of x, so even though P_B is a public key, nobody can remove the mask xP_B. However, A also includes a clue, which is enough to remove the mask if one knows the private key n_B. For an attacker to recover the message, the attacker would have to

compute x given G and xG, which is hard, and known as Elliptic Curve Discrete Logarithmic Problem (ECDLP).

ECDH – Elliptic Curve Diffie Hellman:

ECDH, a variant of Diffie Hellman (DH), is a key agreement algorithm. For generating a shared secret between A and B using ECDH, both have to agree upon on Elliptic Curve domain parameters. An overview of ECDH is given below (R. Afreen, S.C. Mehrotra, 2011) (S. K. Verma, Dr. D.B. Ojha, 2012).

Key Agreement Algorithm:

For establishing shared secret between two devices A and B:

- Let d_A and d_B be the private key of device A and B respectively. Private keys are random number less than n, where n is a domain parameter.
- Let $Q_A = d_A * G$ and $Q_B = d_B * G$ be the public key of device A and B respectively, G is a domain parameter.
- A and B exchanged their public keys.
- The end A computes $K = (x_K, y_K) = d_A * Q_B$.
- The end B computes $L = (x_L, y_L) = d_B * Q_A$.
- Since K = L, shared secret is chosen as x_K

Since it is practically impossible to find the private key d_A or d_B from the public key Q_A or Q_B, it's not possible to obtain the shared secret for a third party.

ECDSA - Elliptic Curve Digital Signature Algorithm:

Digital Signature Algorithm (DSA) is a public key algorithm that is used for Digital Signature. ECDSA is a variant of the Digital Signature Algorithm (DSA). For sending a signed message from A to B, both have to agree upon on Elliptic Curve domain parameters. Sender A has a key pair consisting of a private key d_A (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key $Q_A = d_A * G$ (G is the generator point, an elliptic curve domain parameter). An overview of ECDSA process is defined below (R. Afreen, S.C. Mehrotra, 2011) (S. K. Verma, Dr. D.B. Ojha, 2012).

Signing:

- Consider the device A that signs the data M that it sends to B.
- Let d_A be A's private key.
- Calculate m = HASH (M), where HASH is a hash function.
- Select a random integer k such that $0 < k < n$.
- Calculate $r = x_1 \bmod n$, where $(x_1, y_1) = k * G$.
- Calculate $s = k^{-1}(m + d_A * r) \bmod n$.
- The signature is the pair (r, s).
-

Verification:

- Let M be the message and (r, s) be the signature received from A.
- Let Q_A be A's public key. Since Q_A is public, B has access to it.

- Calculate m = HASH (M).
- Calculate $w = s^{-1} \bmod n$.
- Calculate $u_1 = (m * w) \bmod n$ and $u_2 = (r * w) \bmod n$
- Calculate $(x_1, y_1) = u_1 * G + u_2 * Q_A$
- The signature is valid if $x_1 = r \bmod n$, invalid otherwise.

VIII CONCLUSION

We studied various algorithms for point multiplication of elliptic curves and we can see that these algorithms are prone to side channel attacks, mainly simple power analysis attacks. Montgomery ladder was not considered to be vulnerable to side-channel attacks but it was found that it was also vulnerable to FLUSH+RELOAD attack. Then we gave a brief account on Euclid's addition chain which has been used to produce an algorithm resilient of SCAs but it also has a drawback that it is not easy to find small addition chains. Finally some applications of elliptic curve are discussed.

REFERENCES

[1] A. Byrne, N. Meloni, A. Tisserand, E.M.Popovici and W.P.Marnane, "Comparison of Simple Power Analysis Attack Resistant Algorithms for an Elliptic Curve Cryptosystem", Journal of Computers, vol. 2 (10), 2007, 52-62.

[2] A. Byrne, N. Meloni, F. Crowe, W. P. Marnane, A. Tisserand and E. M. Popovici, "SPA resistant Elliptic Curve Cryptosystem using Addition Chains", 4th International Conference on Information Technology (ITNG'07), IEEE 2007, 995-1000.

[3] D. R. Stinson, "Cryptography: Theory and Practice", Third Edition, Chapman & Hall/CRC Taylor & Francis Group, 2006, 258-267.

[4] G.V.S. Raju and R. Akbani, "Elliptic Curve Cryptosystem and its Applications", Systems, Man and Cybernetics, IEEE International Conference, vol. 2, 2003, 1540-1543.

[5] K. Zhang and T. Yan, "An Improved Algorithm of Elliptic Curve Cryptograph", International Journal of Security and Its Applications Vol.7 (5), 2013, 237-248.

[6] M. Hedabou, P. Pinel and L. Bénéteau, "A comb method to render ECC resistant against Side Channel Attacks", http://eprint.iacr.org/2004/342.pdf, 2004.

[7] M. Joye, "Elliptic Curves and Side-Channel Analysis", ST Journal of System Research 4(1), 2003, 283-306.

[8] M. Rivain, "Fast and Regular Algorithms for Scalar Multiplication over Elliptic Curves", https://eprint.iacr.org/2011/338.pdf, 2011.

[9] Md. R. Islam, Md. S. Hasan, I. S. M. Asaduzzaman, "A New Point Multiplication Method for Elliptic Curve Cryptography Using Modified Base Representation", International Journal of The Computer, the Internet and Management Vol.16 (2), 2008, 9-16.

[10] N. Benger, J. van de Pol, N. P. Smart, and Y. Yarom, ""Ooh Aah... Just a Little Bit": A small amount of side channel can go a long way", https://eprint.iacr.org/2014/161.pdf, 2014.

[11] P. K. Mishra, V. Dimitrov, "Efficient Quintuple Formulas for Elliptic Curves and Efficient Scalar Multiplication Using Multibase Number Representation", Information Security, vol. 4779, 2007, 390-406.

[12] P.-Y. Liardet and N. P. Smart, "Preventing SPA/DPA in ECC systems using the Jacobi form", In Cryptography Hardware and Embedded Systems-CHES'01", vol 2162, LNCS, 2001, 391-401 .

[13] R. Afreen and S.C. Mehrotra, "A review on elliptic curve cryptography for embedded systems", International Journal of Computer Science & Information Technology (IJCSIT), vol. 3 (3), 2011, 84-103.

[14] S. K. Verma and Dr. D.B. Ojha, "A Discussion on Elliptic Curve Cryptography and Its Applications", IJCSI International Journal of Computer Science Issues, vol. 9 (1), 2012, 74-77.

[15] V. Dimitrov, L. Imbert, and P. K. Mishra, "Efficient and Secure Elliptic Curve Point Multiplication using Double-Base Chains", Advances in Cryptology- ASIACRYPT, vol. 3788, 2005, 59-78.

[16] Y. Yarom and K. Falkner, "Flush+Reload: a High Resolution, Low Noise, L3 Cache Side-Channel Attack", https://eprint.iacr.org/2013/448.pdf, 2013.

[17] Y. Yarom and N. Benger, "Recovering OpenSSL ECDSA Nonces Using the FLUSH+RELOAD Cache Side-channel Attack", IACR Cryptology ePrint Archive, vol. 2014, 2014.

Biographical Notes

Indu Bala Thingom is presently pursuing M.Tech. final year in Information Technology Department (Specialization in Cryptography) from N.E.H.U. Shillong, Meghalaya, India.

PREDICTION OF IMPROVEMENT OF EFFICIENCY IN A THERMAL POWER PLANT

Anjali T H[1], Dr. G Kalivarathan[2]

[1]*Department of Electrical Engineering, NCERC, Thrissur (India)*

[2]*Professor, Department of Mechanical Engineering, NCERC, Thrissur (India)*

ABSTRACT

In recent era, most of the electricity produced throughout the world is from steam power plants. So it is very important to ensure that the plants are working at their maximum efficiency. Thermodynamic analysis of the Rankine cycle has been undertaken to enhance the efficiency and reliability of steam power plants. Power plants contain many equipments and the plant efficiency is an important factor. Even a small improvement in any part of the plant will result in a significant improvement in the plant efficiency. Factors affecting efficiency of the Rankine cycle have been identified and analyzed for improved working of thermal power plants.

This paper deals with the importance of the efficiency of a thermal power plant and Rankine cycles used in thermal power plants. In order to improve the efficiency and performance of a plant, it is necessary to regularly check all the equipments and estimate their efficiencies separately and periodically.

***Keywords*: Overall Efficiency, Pump Efficiency, Thermal Efficiency, Turbine Power, Turbine Efficiency**

I INTRODUCTION

In a power plant, burning of coal produces the heat energy. However, the coal is normally coming in big chunks. Burn this chunks is not possible. So pulverize the coal, and then blow it with the air. That goes along with the air to the chamber, where there is a flame. Therefore, the whole things burns and the complete burning occur.

In an actual physical thermal power plant, the water coming out through outside the boiler. There are pipes that bring all the way down, it is inserted from here, and there are pipes along the wall. Water by natural circulation will come down. The water is get heat and then return to the drum which concept called water walls. The drum actually stores the water and from which it is circulated. This circulation ensures highest amount of heat transfer from the burning gas to the fluid. Through this water walls the elementary boiling process taken place. In this way, the steam is generated and steam needs to be taken out and the state of the steam is just before the super heated steam state.

The steam from the steam drum then goes to the superheater, the super heater heat transfer elements essentially made of copper, because of high heat transfer constant of this copper material. The output is superheated steam.

The turbine that is something that allows the steam to expand. The steam expand in nozzle essentially converts pressure energy to kinetic energy. The kinetic energy makes it impinge on to the blades and makes the blades rotate. This turbine is coupled to a generator, which generates electrical energy. The steam from the turbine is led to the steam condenser of turbine, which makes suction at low pressure and permits the expansion of the steam in the turbine to a low pressure.

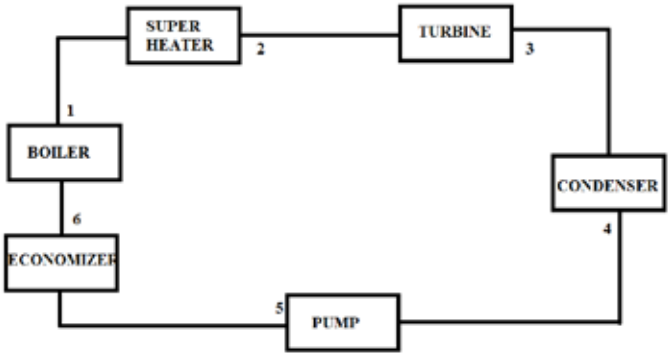


Fig. 1.Thermal Power Plant

The condensate along with some fresh make up feed water is again fed into the boiler by pump (called the boiler feed pump). The steam is condensed by cooling water. Cooling water recycles through cooling tower. These constitute cooling water circuit. The ambient air is allowed to enter in the boiler. In addition, the flue gas comes out of the boiler and exhausted into atmosphere through stacks. These constitute flue gas and air circuit. The flow of air and the static pressure inside the steam boiler (called draught) is maintained by two fans called Forced Draught (FD) fan and Induced Draught (ID) fan.

II RANKINE CYCLE

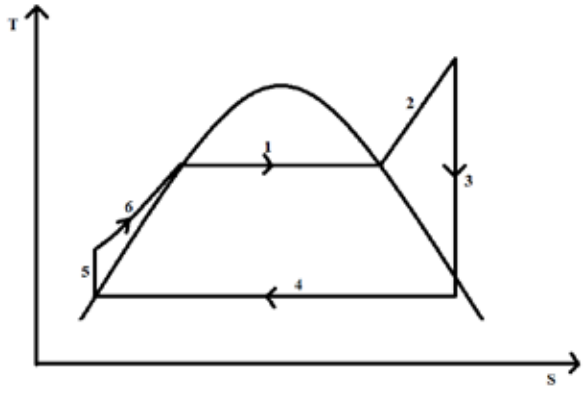


Fig. 2. Ideal Rankine Cycle

The ideal thermodynamic cycle to which the operation of a thermal power station closely resembles to the Rankine cycle.

1. Water is heated up at constant temperature. Water is being heated up means it has reached the 100^0C temperature or depending on the pressure inside the boiler the temperature increases.

2. At super heater, there is a change in temperature and change in entropy. Whenever there is a heat transfer, there is a change in entropy.
3. Turbine: it extracts energy out of it. Nevertheless, there is no heat transfer so no entropy change. But in practice the temperature falls.
4. Condenses steam to water. Here taking the heat back then there is a change in entropy. But the temperature is same. Because the latent heat is removing from the steam.
5. The pump will require far less amount of power because it is handling water. Pump is just like the turbine no heat transfer is happen. So, ideally the pump also should have the entropy constant. But the pushing of water makes some work into it. This makes some rise in temperature. All pumps have the temperature rising a bit, so it should be a bit raised.

III ANALYSIS OF RANKINE CYCLE

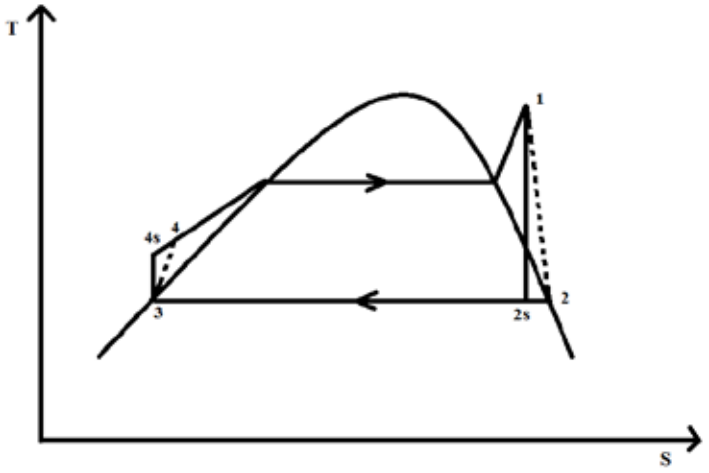


Fig. 3. Rnkine Cycle

All the power plants are run under steady state conditions (assume). The starting and shutting down working conditions are excluded where deviations from a steady state cannot be avoided. With reference to the Rankine cycle shown in Figure 3, a valve, that is exercise the control of the steam flow. The adjustments in the valve regulates the flow of steam in the turbine and power output. In the ideal Rankine cycle, working fluid follows reversible an adiabatic path in the turbine and is subjected to lower pressure and temperature in the condenser.

Applying the First Law of Thermodynamics for an isentropic turbine is:

$$q=0=h_2-h_1+w_t[KJ/Kg] \quad (1)$$

Where potential and kinetic energy differences between the inlet and outlet are negligibly small. So that the turbine work per unit mass passing through the turbine is the difference between the entrance and exit enthalpies,

$$w_t= h_2-h_1 [KJ/Kg] \quad (2)$$

The power delivered by the turbine to an external load, like an electrical generator, is given as,

Turbine power=$m_s w_t = m_s(h_2-h_1)$[KW] **(3)**

Applying the steady-flow First Law of Thermodynamics to the boiler, shaft work is zero and the boiler heat transfer is given by the equation;

$q_b = h_1-h_4$[KJ/Kg] **(4)**

The condenser is positioned next to the turbine to receive a large flow rate of low-pressure steam. This steam in the condenser goes under a phase change from vapour to liquid water (latent heat). External cooling water is inducted through thousands of tubes in the condenser to transport the heat of condensation of the steam away from the plant. The condensate leaving from the condenser is at a low temperature and pressure. The rejection of heat to the surroundings by the cooling water is essential to maintain the low pressure in the condenser. By applying the steady-flow, First Law of Thermodynamics to the condensing steam enables:

$q_c = h_3-h_2$[KJ/Kg] **(5)**

The value of q_c is negative because h_2 is greater than h_3. According to sign convention, q_c represents an outflow of heat from the condensing steam. This heat is absorbed by the cooling water and passing through the condenser tubes.

A pump is a device, which moves liquid from a low pressure to high pressure. In the Rankine cycle, the condensate is raised to the pressure of the boiler by boiler feed pumps (BFP). The high-pressure water entering the boiler is called feed water. From the steady-flow First Law of Thermodynamics, the work required to drive the pump are given by the equations,

Work, $w_p = h_3-h_4$[KJ/Kg] **(6)**

The pump work has a negative value as h_4 greater than h_3. According to the thermodynamic sign convention, this indicates that work and power must be given to operate the pump.

The difference between the turbine power and the magnitude of the pump power is called the net power delivered by the Rankine cycle.

Net Power= $(h_1-h_2+h_3-h_4)$ **(7)**

Thermal efficiency is a measure of the effectiveness of an energy conversion device. It is the ratio of the cycle network to the heat supplied from external sources. Thus, by using the above equations (2,4,6) the ideal Rankine cycle thermal efficiency in terms of cycle enthalpies is given as:

$\eta = (h_1-h_2+h_3-h_4)/(h_1-h_4)$ **(8)**

Rankine-cycle efficiency improves when the average heat-addition temperature increases and the heat rejection temperature decreases. The cycle efficiency will be improved by increasing turbine inlet temperature and decreasing the condenser pressure (and thus the condenser temperature).

3.1 Efficiency of Power Plants

Turbine, boiler and a pump are the basic components of a steam thermal power plant. An analytical discussion about the steam turbine and pump functions has been taken up as these affect the Rankine cycle efficiency.

The efficiency of a steam turbine is defined as the actual work produced divided by the work produced by an isentropic expansion. An isentropic expansion is the amount of work that would be produced if there is no change in entropy occurred.

3.1.1 Steam turbine

An isentropic process is an idealized process that represents the amount of available energy. The second law of thermodynamics, however, states that the conversion of this thermal energy to useful work cannot be 100% efficient.

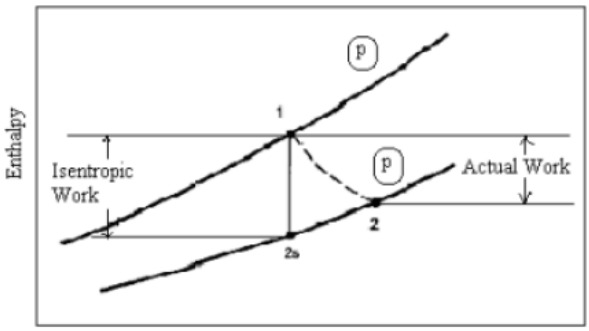


Fig.4. Turbine Efficiency

The adiabatic steam turbine with irreversible flow exhibits the same thermodynamic results as in the case of an isentropic turbine shown in the equation below;

$$w_t = h_2 - h_1 \text{ [KJ/Kg]} \quad (9)$$

Here h_2 represents the actual exit enthalpy and w_t is the actual work of an adiabatic turbine where all the above discussed non-ideal irreversible components working under non-ideal conditions that exist. The efficiency of a real turbine, known as isentropic efficiency, is defined as the ratio of the actual shaft work to the shaft work for an isentropic expansion between the same inlet and exit pressure.

The turbine efficiency is:

$$\eta turb = w_t = (h_1 - h_2)/(h_1 - h_{2s}) \quad (10)$$

where h_{2s} is enthalpy evaluated at the turbine inlet entropy and at the exit pressure.

3.1.2 Pump

The turbine condensate is recycled by using a pressure centrifugal pump. External work must be supplied to a pump to move liquid from a low pressure to a high pressure. The sensible internal energy of the liquid water is enhanced by means of doing work through pumping, however, considerable work energy is lost due to irreversibility. Therefore, the remaining effective work to raise the pressure is less than supplied. Pump efficiency is the ratio of the

isentropic work to the actual work input when operating between two given pressures. The isentropic pump work, $w_{ps} = h_3 – h_{4s}$, and the pump isentropic efficiency is:

$$\eta_{pump} = (h_{4s}–h_3) / (h_4–h_3) \quad (11)$$

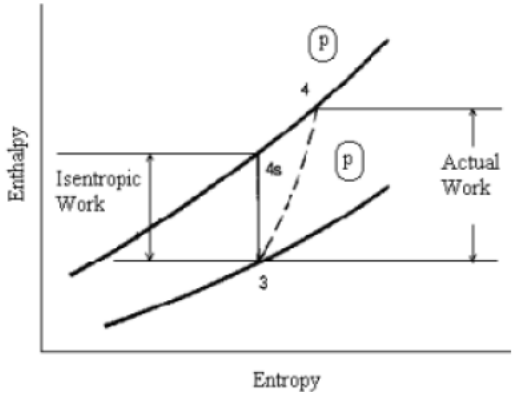


Fig.5. Pump Efficiency

IV OVERALL EFFICIENCY OF A PLANT

Ideal work output per unit mass of steam,

$$(W_t)_{max}=h_1-h_{2s} \quad (12)$$

This work is the reversible and adiabatic enthalpy drop in turbine. But it is not obtainable, since no real process is reversible. The expansion process is accompanied by irreversibilities. The actual final state 2 can be defined (Fig. 3), since the temperature, pressure, and quality can be found by actual measurement. The actual path 1-2 is not known and its nature is immaterial, since the work output is here being expressed in terms of the change of a property, enthalpy. Accordingly, the work done by the turbine in irreversible adiabatic expansion from 1-2 is

$$(W_t)_{actual}=h_1-h_2 \quad (13)$$

This work is known as internal work, since only the irreversibilities within the flow passages of turbine are affecting the state of steam at the turbine exhaust.

Internal output =Ideal output-Friction and other losses (14)

If W_s is the steam flow rate in Kg h^{-1}

Internal output=$W_s(h_1-h_2)$KJ h^{-1}

Ideal output= $W_s(h_1-h_{2S})$ KJ h^{-1}

For calculating the overall efficiency of the plant we have the equation is,

$$\eta_{overall}= \eta_{boiler*}\eta_{cycle*} \eta_{tur-mech*} \eta_{generator} \quad (15)$$

Where,

$$\eta_{boiler}= \text{energy utilized/energy supplied} \quad (16)$$

$$\eta_{boiler}=W_s(h_1-h_4)/W_fCV \quad (17)$$

Where

W_f is the fuel burning rate in the boiler (Kgh^{-1})

CV is the calorific value of the fuel (KJ Kg^{-1})

η_{cycle}=internal output/Q **(18)**

$\eta_{cycle}=W_s(h_1-h_2)/ W_s(h_1-h_4)$ **(19)**

$\eta_{tur-mech}$=Brake output/internal output **(20)**

$\eta_{tur-mech}=$Brake output/ $W_s(h_1-h_2)$ **(21)**

Where,

Brake Output=Internal output-External Losses

η_{gen}=output at the generator terminals/Brake output of turbine **(22)**

η_{gen} = KW*3600/Brake output **(23)**

V AREAS OF PULVERIZED COAL PLANT WHERE EFFICIENCY LOSS CAN OCCURE

The overall efficiency of a power plant encompasses the efficiency of the various components of a particular generating unit. Sometimes these systems are unique to a generating unit, while in other instances these systems may be shared between generating units at a power plant site. As coal-fired power plants age, they lose efficiency. Much of this loss in efficiency is due to mechanical wear on a variety of components resulting in heat losses, as can be seen in Figure 6. Lower power plant efficiency results in more CO_2 being emitted per unit of electricity generated.

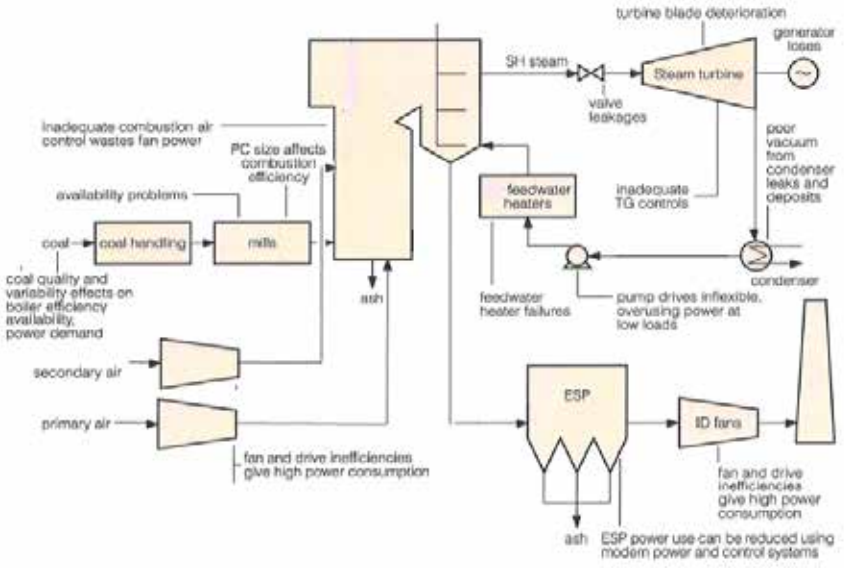


Fig.6. Efficiency Losses Can Occure In A Thermal Power Plant

VI CONCLUSION

The objective of the study was to analyze the overall efficiency and the Rankine cycle efficiency of a thermal power plant. There are many factors, which are influencing the efficiency of the thermal power plant. The fuel used for

combustion, type of boiler, varying load, power plant age, they lose efficiency. Much of this loss in efficiency due to mechanical wear on variety of components resulting heat losses. Therefore, it is necessary to regularly check all the equipments periodically. Moreover, it is noticed that the overall efficiency of any thermal power plant depends upon the technical difficulties under unpredictable conditions. Hence, a viable study is carried out to assess the performance of thermal power plant in this context.

REFERENCES

[1] Manoj Kumar Gupta," Power Plant Engineering" (PHI Learning Private Limited, New Delhi, 2012.)

[2] K.Krishnaswamy and M. PonniBala,"Power Plant Engineering"(PHI Learning Private Limited, New Delhi, 2012)

[3] B.K Sarkar," Thermal Engineering" (19th print Tata Mc graw Hill Education Private Limited, New Delhi, 2011)

[4] RaviprakashKurkiya and Sharad Chaudhary," Energy Analysis of Thermal Power Plant", International Journal of Scientific & Engineering Research, Vol.3, Issue 7, 2012, pp.1-7,.

[5] R K Kapooria, S Kumar and K S Kasana, "An Analysis of a Thermal Power Plant Working on a Rankine Cycle: A Theoretical Investigation", Journal of Energy in Southern Africa, Vol.19 No 1, 2008, pp.77-83.

SOLUTION OF OPTIMAL POWER FLOW USING EVOLUTIONARY TECHNIQUES

Suman Ranjan[1] , Kumari Sarwagya[2]

[1,2]*Dept. of Electrical Engineering,Indian School of Mines, Dhanbad, Jharkhand (India)*

ABSTRACT

The optimal power flow is an important problem of power systems in which certain control variables are adjusted to minimize an objective function such as to minimize the active power generation loss and improves the voltage profile while satisfying physical and operating limits on control variables. In this paper, two evolutionary search techniques i.e. Modified Particle Swarm Optimization (MPSO) and Gravitational Search Algorithm (GSA), have been proposed to solve the optimal power flow problem as an optimization problem. Comparison of both the techniques has been made between each other and among the other predefined heuristic search algorithms. GSA has a better performance than others in converging to global optimum. It also avoids falling into local optimum and premature convergence and hence the search behavior is better. The results of GSA are promising and it shows the effectiveness and robustness than other methods.

Keywords: Gravitational Search Algorithm, Modified Particle Swarm Optimization, Optimal Power Flow.

I INTRODUCTION

Power-flow studies are of great importance in planning and designing the future expansion of power systems as well as in deter mining the best operation of existing systems.[1-3] The principal information obtained from a power-flow study is the magnitude and phase angle of the voltage at each bus and the real and reactive power flowing in each line. Optimal load flow deals with the steady state analysis of an interconnected power system during normal operation. The system is assumed to be operating under balanced condition and is represented by a single-phase network. The network contains hundreds of nodes and branches with impedances specified in per unit on a common MVA base. When node currents are specified, the set of linear equations can be solved for the node voltages.[4-6] However, in a power system, powers are known rather than currents. Thus, the resulting equations in terms of power, known as the power flow equation, become non-linear and must be solved by iterative techniques (such as Gauss-Siedal Method, Newton-Raphson Method etc). Power flow studies, commonly referred to as load flow, are the backbone of the power system analysis and design. They are necessary for planning, operation, economic scheduling and exchange of power between utilities. In addition, power flow analysis is required for many other analyses such as transient stability and contingency studies.

The conventional methods used in the literature for solving OPF problem, which was initially introduced by Carpentier in 1962 , are i) lambda iteration method, ii) gradient method, iii) Newton method, iv) linear programming and v) interior point algorithm. Although these techniques have been successfully applied for solving the OPF problem, there are some difficulties related to them in handling inequality constraints and

convergence to global minima. Therefore, the heuristic approaches have been widely used to overcome these difficulties recently. Artificial Bee Colony (ABC), Genetic Algorithm (GA), Evolutionary Programming (EP) and Particle Swarm Optimization (PSO) can be listed among the examples of the heuristic methods used for solution of OPF problem [7-14].

OPF minimizes a given objective function satisfying the equality and inequality constraints. The objective function of the OPF problem is to minimize the active power losses and improves the voltage profile.

This paper determines the possible optimal solution for OPF problems while considering all equality and inequality constraints. The proposed approach have been applied and tested on IEEE 14 and IEEE 30-bus standard system. MATLAB software has been used for simulation. Simulation results, confirms the proficiency and capabilities of proposed approach, compared to those reported in the literature.

II OPTIMAL POWER FLOW

A. Introduction

The objective of OPF problem is to identify minimum generation cost of generator units meeting equality and inequality constraints. The equality constraints represent conventional power flow equations and the inequality constraints represent the system operating and control limits. Mathematically, the OPF problem is formulated as a nonlinear optimization problem with equality and inequality constraints, as shown below

Minimize $f(u,x)$ ..…….. (1)

Subject to:

$h(u,x)=0$ …..…… (2)

$g(u,x) \pounds 0$ ……. .(3)

where “u” is the set of controllable quantities in the system and “x” is the set dependent variables. The objective function “f” is to minimize the total fuel cost of all generating units.

B. Constraints

There are few constraints which should be considered before the solution of Optimal Power flow. Some of these constraints are described below:

1. Generation Constraints

P_G, V_G, Q_G in generator are restricted by their upper and lower limits as follows :

$V_{G_i}^{\min} \pounds V_G \pounds V_{G_i}^{\max}$ i=1,…N …… (4)

$P_{G_i}^{\min} \pounds P_G \pounds P_{G_i}^{\max}$ i=1,…N ……. (5)

$Q_{G_i}^{\min} \pounds Q_G \pounds Q_{G_i}^{\max}$ i=1,…N …….. (6)

2. Transformer Constraints

The constraints include Transformer Tap setting that is given by

$$T_i^{\min} \le T_i \le T_i^{\max} \qquad i=1,\ldots N \qquad \ldots\ldots\ldots (7)$$

3. Shunt VAR Constraints

The constraints is given by :

$$Q_{ci}^{\min} \le Q_{ci} \le Q_{c_i}^{\max} \qquad i=1,\ldots N \qquad \ldots\ldots\ldots (8)$$

4. Security Constraints

It includes load bus voltage constraints and transmission line loadings that are given by

$$V_{Li}^{\min} \le V_{Li} \le V_{L_i}^{\max} \qquad i=1,\ldots N \qquad \ldots\ldots\ldots (9)$$

$$S_{Li} \le S_{L_i}^{\max} \qquad i=1,\ldots N \qquad \ldots\ldots\ldots (10)$$

III PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart invented Particle Swarm Optimization (PSO) in 1995 . The PSO can be best understood through an analogy of a swarm of birds in a field. Without any prior knowledge of the field, the birds move in random locations with random velocities looking for foods. In PSO, particles change their positions (states) with time. Let 'x' and 'v' denote a particle coordinates (position) and its corresponding flight speed (velocity) in a search space respectively. The best previous position of the i^{th} particle is recorded and represented as pbest. The index of the best particle among all the particles in the group is represented by the gbest. The modified velocity and position of each particle can be calculated as per following formulas:

$$v_{id}^{k+1} = w.v_{id}^{k} + c_1.rand_1(pbest_{id} - x_{id}^{k}) + c_2.rand_2(gbest_{id} - x_{id}^{k}) \qquad \ldots\ldots\ldots (11)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad \ldots\ldots\ldots (12)$$

The inertia weighting function (w^k) is usually calculated using following equation

$$w^k = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \qquad \ldots\ldots\ldots (13)$$

where, i=1, 2, 3, …….n; d=1, 2, 3, ……N; 'n' is the number of particles in a group; 'N' is the number of members in a particle.

M. Clerc has derived a constriction coefficient K

$$K = \frac{2}{\left| 2\text{Æ} - \sqrt{\text{Æ}^2 - 4\text{Æ}} \right|} \qquad \ldots\ldots\ldots (14)$$

Where $\text{F} = c_1 + c_2 \;, \text{F} \ge 4$ ……. (15)

The new velocity and position updating equation are

$$v_{id}^{k+1} = K' \left(w.v_{id}^{k} + c_1.rand_1(pbest_{id} - x_{id}^{k}) + c_2.rand_2(gbest_{id} - x_{id}^{k})\right) \quad \ldots\ldots\ldots\ldots (16)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$$

3.1 Exploration and Exploitation of Inertia weight factor

To accurately locate the global minimum with the PSO algorithm the step size has to be reduced when particles approaches the global minimum. This is usually done by incorporating an inertia weight

$0<w<1$

Use of inertia weight essentially narrows the region of search as search progresses. Values of "w" is initially 1 and decreases to 0.

w =1; means exploration of new regions of search space

w ≅0; means exploitations of detailed search of the current region.

Main steps of PSO techniques are given below:

Step I This particles are randomly generated between the operating limits.

Step II The values of the fitness function of the particles are evaluated using objective function and dimensions (variables) of the particles are initialized as *pbest(s).*

Step III The best value of *pbest(s)* is represented as *gbest.*

Step IV The particle's velocities and positions are updated using velocity and position updating equations.

Step V The new fitness function values are evaluated using the updated positions of the particles. If the current position of the particle is better than its previous *pbest*, the *pbest* is updated by the current particle, otherwise not updated. The updated *gbest* is the best among all the *pbest(s).*

Step VI If the stopping criterion is satisfied, go to step 7, otherwise, go to step 2.

Step VII The particle that generates the largest *gbest* yields the optimal variables.

IV GRAVITATIONAL SEARCH ALGORITHM

In this section, we introduce our optimization algorithm based on the law of gravity [28]. In the proposed algorithm, agents are considered as objects and their performance is ensured by their masses. All these objects attract each other by the gravity force, and this force causes a global movement of all objects towards the objects with heavier masses. Hence, masses co-operate using a direct form of communication, through gravitational force. The heavy masses which correspond to good solutions move more slowly than lighter ones, this guarantees the exploitation step of the algorithm, in GSA, and each mass (agent) has four specifications [15-16]:

1. Position
2. Inertial mass
3. Active gravitational mass
4. Passive gravitational mass.

The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses and determined using a fitness function. In other words, each mass presents a solution, and the algorithm is navigated by properly adjusting the gravitational and inertial masses. By lapse of time, we expect that masses be attracted by the heaviest mass. This mass will present an optimum solution in the search space. The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. More precisely, masses obey the following laws:

Law of gravity: Each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them, R. We use here R instead of R^2, because according of our experiment results, R provides better results than R^2 in all experiment cases.

Law of motion: The current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by the mass of inertia.

Now, consider a system with N agents (masses). We define the position of the i^{th} agent by

$X_i = (x_i^1, \ldots\ldots\ldots, x_i^d, \ldots\ldots x_i^n)$

for $i = 1,2,3\ldots..N$.

where x_i^d presents the position of i^{th} agent in the d^{th} dimension.

At a specific time 't', we define the force acting on the mass 'i' from mass 'j' as following:

$$F_{ij}^{d}(t) = G(t)\frac{M_{pi}(t) * M_{aj}(t)}{R_{ij}(t) + \hat{I}}(x_j^d(t) - x_i^d(t)) \qquad \ldots\ldots\ldots (16)$$

where M_{aj} is the active gravitational mass related to agent j, M_{pi} is the passive gravitational mass related to agent i, $G(t)$ is the gravitational constant at time t, ε is a small constant, $R_{ij}(t)$ is the Euclidian distance between two agents i and j:

$$R_{ij}(t) = \square X_i(t), X_j(t) \square . \qquad \ldots\ldots\ldots (17)$$

To give a stochastic characteristic to our algorithm, we suppose that the total force that acts on agent i in a dimension d be the randomly weighted sum of d^{th} components of the forces exerted from other agents:

$$F_i^d(t) = \sum_{j=1, j \neq i}^{N} rand_j F_{ij}^d(t) \qquad \ldots\ldots\ldots (18)$$

where $rand_j$ is a random number in the interval [0,1].

Hence, by the law of motion, the acceleration of the agent i at time t, and in dimension d^{th}, $a_i^d(t)$, is given below;

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \qquad \ldots\ldots (19)$$

where $M_{ii}(t)$ is the inertial mass of i^{th} agent.

Furthermore, the next velocity of an agent is considered as a fraction of its current velocity added to its acceleration. Therefore its position and its velocity could be calculated as follows:

$$v_i^d(t+1) = rand_i * v_i^d + a_i^d, , \quad \text{…….. (20)}$$

$$x_i^d(t+1) = x_i^d + v_i^d(t+1), \quad \text{…….. (21)}$$

where, $rand_i$ is a uniform random variable in the interval [0,1]. We use this random number to give a randomized characteristic to the search.

r t The gravitational constant, G, is initialized at the beginning and will be reduced with time to control the search accuracy. In other words, G is a function of the initial value 'G_0' and time 't'.

G (t) = G (G_0, t) …..… (22)

Gravitational and inertia masses are simply calculated by the fitness evaluation. A heavier mass means a more efficient agent. This means that better agents have higher attractions and walk more slowly. Assuming the equality of the gravitational and inertia mass, the values of masses is calculated using the map of fitness. We update the gravitational and inetial masses by the following equation:

$M_{ai} = M_{ii} = M_{pi} = M_i$, $i=1,2,\ldots N$.

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad \text{…… (23)}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad \text{…… (24)}$$

where, $fit_i(t)$ represent the fitness value of the agent *i* at time *t*, $\min_{j\hat{I}\,\acute{a}1\ldots N\tilde{n}} fit_j(t)$ and *worst(t)* and the *best(t)* are defined as follows (for a minimization problem):

best(t) = $\min_{j\hat{I}\,\acute{a}1\ldots N\tilde{n}} fit_j(t)$ …... (25)

worst(t) = $\max_{j\hat{I}\,\acute{a}1\ldots N\tilde{n}} fit_j(t)$ …… (26)

It is to be noted that for a maximization problem, the above equations are changed and the given equations are

best(t) = $\max_{j\hat{I}\,\acute{a}1\ldots N\tilde{n}} fit_j(t)$ …….. (27)

worst (t) = $\min_{j\hat{I}\,\acute{a}1\ldots N\tilde{n}} fit_j(t)$ …… (28)

One way to perform a good compromise between exploration and exploitation is to reduce the number of agents with lapse of time. Hence, we propose only a set of agent with bigger mass apply their force to the other. However, we should be careful of using this policy because it may reduce the exploration power and increases the exploitation capability.

Main steps of GSA techniques are given below:

Step I Search space identification.

Step II Randomized initialization.

Step III Fitness evaluations of agents.

Step IV Update *G(t), best(t), worst(t)* and $M_i(t)$ for *i*= 1,2….,N.

Step V Calculation of the total force in different directions.

Step VI Calculation of acceleration and velocity.

Step VII Updating agent's position.

Step VIII Repeat steps 3 to steps 7 until the stop criteria are reached.

Step IX End.

V COMPARATIVE FEATURES OF GSA AND MPSO

In both GSA and MPSO the optimization is obtained by agent movement in the search space, however the movement strategy is different. Some important differences as follows:

1. In MPSO the direction of an agent is calculated using only two best positions *pbest* and *gbest*. But in GSA, the agent direction is calculated based on the overall force obtained by all other agents.
2. In MPSO updating is performed without considering the quality of the solutions and the fitness value are not important in the updating procedure while in GSA the force is proportional to the fitness value and so the agents see the search space around themselves in the influence of the force.
3. MPSO uses a kind of memory for updating the velocity (due to *pbest* and *gbest*). However, GSA is memory less and only the current position of the agents plays a role in the updating procedure.
4. In MPSO, updating is performed without considering the distance between solutions while in GSA the force is inversely proportional to the distance between solutions.
5. Finally, note that the search ideas of these algorithms are different. MPSO simulates the social behaviour of birds and GSA in inspires by a physical phenomenon.

VI RESULT AND DISCUSSION

To verify the feasibility of the proposed method, two different defined power systems data were tested. The proposed MPSO technique has been applied to two different test systems e.g. IEEE 14-bus system and IEEE 30 bus system. The software programs were written in MATLAB 7.10.0(R2010a) language and executed on a 2.5GHz core i-5 personal computer with 4GB RAM. The following MPSO (Modified Particle Swarm Optimization) and PSO parameters have been used after a no. of careful experimentation.

Table I Optimization parameters

Techniques	MPSO	GSA
Population	100	30
Iteration	200	100
C_1	2.00	0.5
C_2	2.05	1.5
w_{max}	0.9	0.9
w_{min}	0.4	0.4

A. IEEE 14-bus test system

There are 5 generator buses present in IEEE 14 bus test data and the remaining are the load bus. The total load demand of IEEE 14 bus system is 259 MW. The parameters required for the testing are given in Table I and the result obtained by applying MPSO and GSA to given parameters are shown in Table II. The convergence characteristic is also coming within the desired limits and are particularly seen in Fig. I and Fig. II.

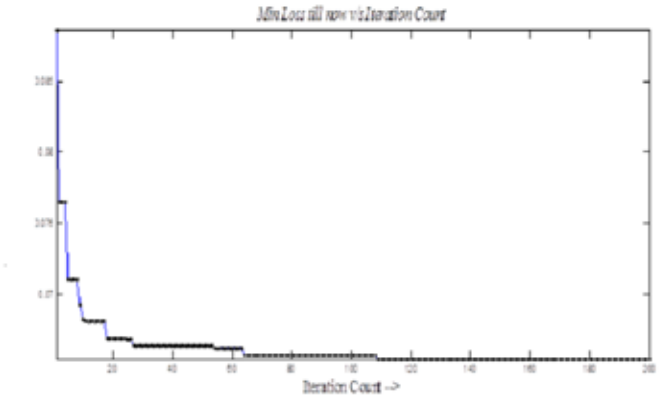


Fig.1 Convergence Characteristic for optimal loss of IEEE 14-bus system using MPSO

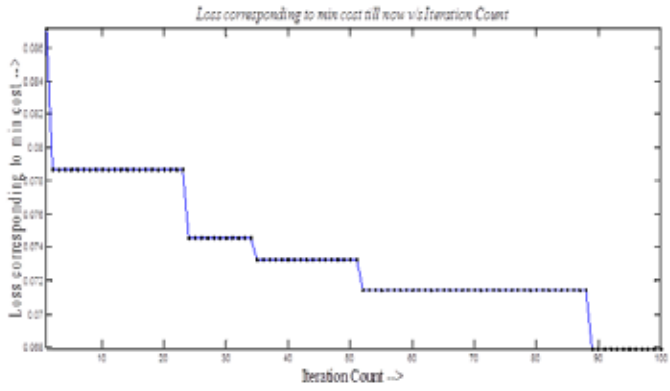


Fig.2 Convergence Characteristic for optimal loss of IEEE 14-bus system using GSA.

Table II Optimal setting of control variables for IEEE 14-Bus test system

Techniques	MPSO	GSA
V_1	1.0600	1.0600
V_2	1.0762	1.0989
V_3	1.0841	1.1142
V_6	1.1416	1.2079
V_8	1.1323	1.1997
Q_2	0.4049	0.5000
Q_3	0.2001	0.4000
Q_6	0.1581	0.1365
Q_8	0.2004	0.2176
Y_{sh12}	0.0568	0.0500
Y_{sh13}	0.0167	0.0500
Y_{sh14}	0.0362	0.0500
T_8	0.9349	0.9500
T_9	1.0224	0.9500
T_{11}	0.9615	0.9500
Loss(MW)	**6.544**	**6.793**
SVD	1.4498	2.1176
Time(sec)	25.529	31.278

B. IEEE 30-Bus test system

There are 6 generator buses present in IEEE 30-bus test data and the remaining are the load bus. The total load demand of IEEE 30-bus system is 283.40 MW. The parameters required for the testing are given in Table I and the result obtained by applying MPSO and GSA to given parameters are shown in Table III and consequently it is compared with the previous recorded result. The convergence characteristic is also coming within the desired limits and are shown in Fig. 3 and Fig.4.

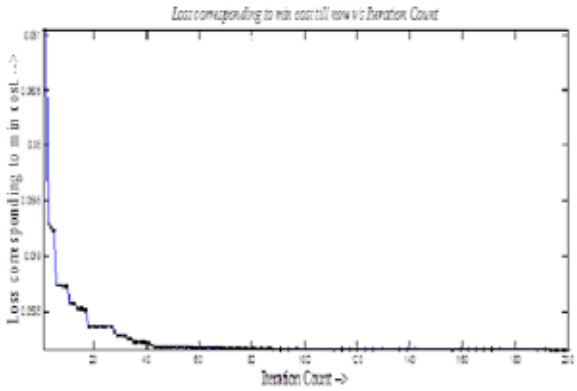


Fig.3 Convergence Characteristic for optimal loss of IEEE 30- bus system using MPSO.

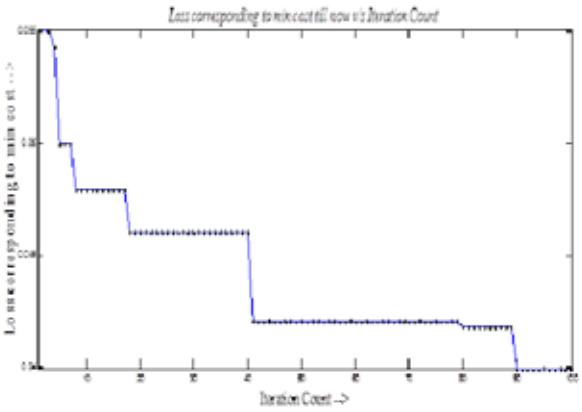


Fig. 4 Convergence Characteristic for optimal loss of IEEE 30-bus system using GSA

VI CONCLUSION

In this work, the proposed novel Gravitational search algorithm (GSA) method and Modified particle swarm optimization (MPSO) method are proposed to solve OPF problems for different objective functions. The feasibility of the methods for solving single objective OPF problem is demonstrated using IEEE 14-bus and IEEE 30-bus system. Both results are compared with each other and superiority of GSA over MPSO in terms of quality, convergence characteristic is shown.

REFERENCES

[1] Mohammad Sadegh Norouzzadeh, Mohammad Reza Ahmadzadeh, Maziar Palhang "Plowing PSO: A Novel Approach to Effectively Initializing Particle Swarm Optimization" IEEE International Conference on Computer Science science and technology,Volume 1, Publication Year: 2010 , Page(s): 705 - 709

[2] Sourav Mallick, S.P. Ghoshal, P. Acharjee, S.S. Thakur "Optimal static state estimation using improved particle swarmoptimization and gravitational search algorithm" Electrical Power and Energy Systems 52 (2013) 254–265

[3] Sanjeev Kumar, D.K. Chaturved i" Optimal power flow solution using fuzzy evolutionary and swarm optimization " Electrical Power and Energy Systems 47 (2013) 416–423

[4] M. N. Suharto, M. Y. Hassan, M. S. Majid, M. P. Abdullah, F. Hussin "Optimal Power Flow Solution Using Evolutionary
Computation Techniques" Power Engineering and Optimization Conference (PEOCO) IEEE International conference,date 6-7 June 2011, pages 77-88

[5] Belgin Emre TURKAY, Rengin Idil CABADAG "Optimal Power Flow Solution Using Particle Swarm Optimization Algorithm" EUROCON,IEEE, July 2013,page(s) 1418-1424

[6] Tohid Ghanizadeh, Ebrahim Babaei "Optimal Power Flow Using Iteration Particle Swarm Optimization" 2012 IEEE

[7] A. Chatterjee, V. Mukherjee, S.P. Ghoshal "Velocity relaxed and craziness-based swarm optimized intelligent PID and PSS controlled AVR system" Electrical Power and Energy Systems 31 (2009) 323–333

[8] P.K. Roy, S.P. Ghoshal, S.S. Thakur "Optimal VAR control for improvements in voltage profiles and for real powerloss minimization using Biogeography Based Optimization" Electrical Power and Energy Systems 43 (2012) 830–838

[9] Shanhe Jiang, Zhicheng Ji, Yanxia Shen "A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints" Electrical Power and Energy Systems 55 (2014) 628–644

[10] Mohammad Rasoul Narimani "A novel approach to multi-objective optimal powerflow by a new hybridoptimization algorithm considering generator constraints and multi-fuel type" Energy 49 (2013) 119e136

[11] S.K. Goswami, P. Acharjee "Multiple low voltage power flow solutions using hybrid PSO and optimalmultiplier method" Expert Systems with Applications 37 (2010) 2473–2476

[12] Mathew M. Noel "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum" Applied Soft Computing 12 (2012) 353–359

[13] Nampetch Sinsuphan, Uthen Leeton, Thanatchai Kulworawanichpong "Optimal power flow solution using improved harmony search method" Applied Soft Computing 13 (2013) 2364–2374.

[14] V. Mukherjee, S.P.Ghoshal, "Craziness based and velocity relaxed swarm optimized intelligent PID controlled AVR system," IEEE powercon, pp. 1-8, Oct 2008.

[15] A. Bhattacharya and P.K.Roy, "Solution of multi-objective optimal power flow using gravitational search algorithm" in *Int. J. IET Gen.Trans. & Dis.*, vol. 6, pp. 751-763, Feb. 2012.

[16] S. Duman, Y. Sonmez, U. Gulvenc and N. Yorukeren, "Optimal reactive power dispatch using gravitational search algorithm," in *Int. J. IET Gen.Trans. & Dis.*, vol. 6, pp. 563-576, Jan. 2012.

A LITERATURE BASED SURVEY ON SWARM INTELLIGENCE INSPIRED OPTIMIZATION TECHNIQUE

Anukaran Khanna[1], Akhilesh Mishra[2], Vineet Tiwari[3] and P.N.Gupta[4]

[1]Assistant Professor, Department of Electronics Engineering,

[2,3]Assistant Professor, Department of Electrical Engineering

United College of Engineering & Research, Allahabad, (India)

[4]Associate Professor, Department of Electronics & Communication Engineering,

University of Allahabad, Allahabad, (India)

ABSTRACT

Nature is of course a great and enormous source of inspiration for solving hard and composite problems in the field of computer science, transportation engineering, mechanical engineering, management and so on, since it exhibits exceptionally diverse, dynamic, robust, and complex phenomenon. It always finds the optimal solution to resolve its problem establishing perfect balance among its components. This is the driving force behind bio inspired computing. Nature inspired algorithms are meta heuristics that mimics the nature for solving optimization problems thus opening a new era in computation .For the past decades ,numerous research efforts has been concentrated in this particular area. Still being infantile and the results being very astonishing, broadens the scope and feasibility of Nature Inspired Algorithms (NIAs) exploring new areas of application and more opportunities in computing. This paper highlights the comparative analysis of nature inspired swarm Intelligence based optimization techniques based on literature analysis and the areas where these algorithms have been most successfully applied.

Keywords*: *Nature Inspired Algorithm, Optimization technique, Swarm Intelligence

I. INTRODUCTION

Every one of us experiences the magnificence of nature. The way it is constructed and managed scientifically is appreciable. However, the complexity beneath it is extremely difficult to understand. For example, functioning of brain and its communication with different parts of the body, relocation of birds every year from one subcontinent to the other and evolution of nature and human being, everything looks incredible and wonderful but simple. In reality, the so-called simple nature is very much complex. Technological advancements provided us with computers to carry out complex tasks. Therefore, if the computers are competent of performing tasks with reasonable degree of complexity, why cannot we take inspiration from nature and mimic it to solve some of the complicated problems that are hard to solve? The answer to this question is yes. It is possible but it is very difficult to mimic nature since nominal information is available in direct form. In spite of these technical hitches, researchers have tried to connect the nature with computation and the nature-inspired algorithms have resulted as an outcome of some of the finest research work. This paper discusses the classification of set of nature-inspired algorithms particularly brief explanation of swarm based algorithm.

II. CLASSIFICATION OF NATURE INSPIRED ALGORITHMS

Nature presents many diverse phenomenons. These can be transformed into mathematical algorithms, which, formulates to mathematical algorithms. However, based on the applicability some of these algorithms have reached a stage of development while some are still in the intangible phase or in a situation where the application mapping is not found. Nature inspired algorithms are broadly categorized into four categories based on the fundamental natural process that they possess : swarm intelligence (SI) based, bio-inspired (but not SI-based), physics/chemistry-based, and others. However, we will focus here on swarm intelligence (SI) based algorithms.

It is worth pointing out the classifications here are not distinctive as some algorithms can be categorized into different categories at the same time. The categorization depends on the real perspective and inspirations. Therefore, the categorization here is just one probable attempt, though the stress is placed on the sources of inspiration.

2.1 Swarm intelligence based

Swarm intelligence (SI) concerns the cooperative, promising performance of multiple, interacting agents who follow some plain rules [1]. While each agent may be measured as unintelligent, the whole system of multiple agents may show some self-organization performance and thus can behave like some sort of cooperative intelligence. Many algorithms have been developed by drawing inspiration from swarm-intelligence systems in nature.

All SI based algorithms use multi-agents, inspired by the cooperative actions of social insects like ants, bees, as well as from other animal societies like flocks of birds or fish. A list of swarm intelligence algorithms and there principle is grouped in Table 1.

SI-based algorithms are amongst the most accepted and extensively used. There are several reasons for such recognition; one of the reasons is that SI-based algorithms usually shared information among multiple agents, so that self-organization, co-evolution and learning all through iterations may help to provide the high efficiency of most SI-based algorithms. Another reason is that multiple agent can be parallelized easily so that large extent optimization becomes more realistic from the implementation point of view.

2.2 Bio-inspired (not SI-based)

In true sense, bio-inspired algorithms form a popular of all nature-inspired algorithms. From the set theory point of view, SI-based algorithms are a subset of bio-inspired algorithms, while bio-inspired algorithms are a subset of nature-inspired algorithms [1]. That is

$$\text{SI-based} \subset \text{bio-inspired} \subset \text{nature-inspired}$$

On the other hand, not all nature-inspired algorithms are bio-inspired, and some are wholly physics and chemistry based algorithms as we will see below. Many bio-inspired algorithms do not use directly the swarming behaviour. Therefore, it is better to call them bio-inspired, but not SI-based like Invasive weed optimization, Paddy Field Algorithm, Flower pollination algorithm etc.

2.3 Physics and Chemistry based

Not all algorithms are bio-inspired, since their sources of inspiration often come from physics and chemistry. For the algorithms that are not bio-inspired, most have been developed by mimicking certain physical and/or chemical laws, including electrical charges, gravity, river systems, etc. As diverse natural systems are relevant to this category, we can even subdivide these into physics and chemistry based algorithms.

2.4 Other algorithms

While researchers develop new algorithms, some may look for idea away from nature. As a result, some algorithms are not bio-inspired or physics/chemistry-based, it is sometimes difficult to put some algorithms in the above three categories, because these algorithms have been developed by using various characteristics from diverse sources, such as social, emotional, etc. In this case, it is better to put them in the other category.

III APPLICATIONS

Nature Inspired Computing techniques are so elastic that they can be applied to broad range of problems, so flexible that they can deal with undetected data and capable of learning, so robust that they can handle imperfect data.. There are three key differences between conventional computing systems and biological information processing systems: components of biological systems react slowly but implement much higher-level operations. The capacity of biological systems to assemble and develop on their own enables much higher interconnection densities. The implementation of biological systems is not a premeditated one.

One who wishes for solutions from nature for complex problems has to first examine the nature's behaviour cautiously. The subsequent step is to use models and record all the behaviours observed so far. The above steps should be repeated till an ideal operational model is obtained. As a by-product some unidentified mechanisms may be found. Based on the examination from nature a problem-solving strategy is formulated. Two main computational applications of NIC are Clustering and Optimization.

3.1 Clustering

Clustering is the unverified categorization of patterns into groups (clusters). An explanation of clustering could be the method of organizing objects into groups whose members are similar in some way. A cluster is therefore a group of objects which are "similar" among them and are "dissimilar" to the objects belonging to other clusters.

The objective of clustering is to determine the intrinsic grouping in a set of unlabeled data. It can be shown that there is no supreme "best" criterion which would be independent of the final objective of the clustering. Consequently, it is the user which must provide this criterion, in such a way that the result of the clustering will suit their needs.

3.2 Optimization

An optimization problem is the problem of finding the best solution from all possible solutions. Optimization problems can be divided into two categories depending on whether the variables are continuous or discrete. Classification of optimization algorithm can be carried out in many ways. A straightforward way is to look at

the nature of the algorithm, and this divides the algorithm into two categories: deterministic algorithms, and stochastic algorithms.

Deterministic algorithms pursue a rigorous process, and its path and values of both design variables and the function are repeatable. On the other hand, stochastic algorithms always have some randomness and every individual path towards a possible solution is not exactly repeatable.

Several techniques are suitable only for specific types of problems. Thus, it is significant to identify the characteristics of a problem and to identify a suitable technique in the context of given problem to find the optimal solution, such that for each class of problems there are different minimization methods, varying in computational requirements, convergence properties, and so on. Optimization problems are classified according to the mathematical characteristics of the objective function, the constraints and the control variables. The most significant characteristic is the nature of the objective function. The relationship between the control variables is of a particular form, such as linear, e.g.

$f(x) = b^t x+c$ ……(1)

Where b is a constant-valued vector and c is a constant, or quadratic, e.g.

$f(x) = x^t Ax\ b^t x+c$ ……...…(2)

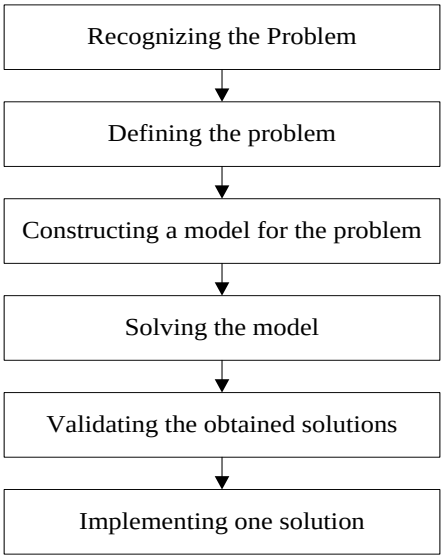


Fig. 1 Flow chart of Optimization Technique formulation

Where A is a constant-valued matrix, special methods exist that are guaranteed to locate the optimal solution very efficiently.

The optimization technique formulates the problem in given below steps:

1. Make a basic configuration
2. Recognize the decision variables
3. Establish the objective function
4. Recognize any constraints
5. Choose and apply an optimization method

IV. SWARM INTELLIGENCE BASED OPTIMIZATION TECHNIQUES

The different SI based optimization techniques are given as:-

4.1 Ant Colony

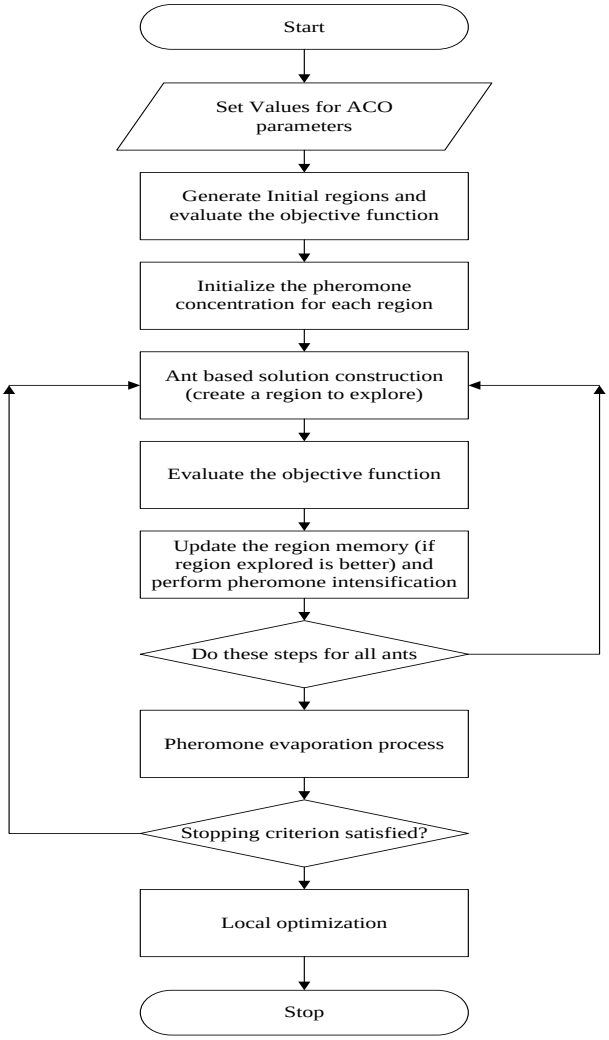


Fig. 2 Flow chart of Ant Colony Optimization Algorithm

ACO is among the most successful swarm based algorithms propounded by Dorigo & Di Caro in 1999 [12] .It is a meta heuristic inspired by the foraging actions of ants in the wild, and moreover, the phenomena known as stigmergy, term used by Grasse in 1959.Stigmergy refers to the indirect communication amongst a self-organizing emergent system via individuals modifying their local environment. The most interesting aspect of the collaborative behavior of several ant species is their ability to find shortest paths between the ants' nest and the food sources by tracing pheromone trails Then, ants choose the path to follow by a probabilistic decision biased by the amount of pheromone: the stronger the pheromone trail, the higher its desirability. Because ants in turn deposit pheromone on the path they are following, this behavior results in a self-reinforcing process leading to the formation of paths marked by high pheromone concentration. By modeling and simulating ant foraging behavior, brood sorting, nest building and self-assembling, etc. algorithms can be developed that could be used for complex, combinatorial optimization problems.

The first ant algorithm, named Ant System (AS), was developed in the nineties by Dorigo et al. (1996) and tested successfully on the well known benchmark Travelling Salesman Problem. The ACO meta heuristic was developed (Dorigo & Di Caro, 1999;) to generalize, the overall method of solving combinatorial problems by approximate solutions based on the generic behavior of natural ants. ACO is structured into three main functions as Ant Solutions Construct, Pheromone Update, and Deamon Actions.

4.2 Artificial bee colony

Artificial bee colony (ABC) Algorithm is an optimization algorithm based on the intelligent behavior of honey bee foraging. This model was introduced by Dervis Karaboga in 2005, and is based on inspecting the behaviors of real bees on finding nectar amounts and sharing the information of food sources to the other bees in the hive. These specialized bees try to maximize the nectar amount stored in the hive by performing efficient division of labour and self-organization [7]. The three agents in Artificial Bee Colony are: The Employed Bee, The Onlooker Bee , The Scout. The employed bees are associated with the specific food sources, onlooker bees watching the dance of employed bees within the hive to choose a food source, and scout bees searching for food sources randomly [4]. The onlooker bees and the scout bees are the unemployed bees. Initially, the scout bees discover the positions of all food sources, thereafter, the job of the employed bee starts. An artificial employed bee probabilistically obtains some modifications on the position in its memory to target a new food source and find the nectar amount or the fitness value of the new source. Later, the onlooker bee evaluates the information taken from all artificial employed bees and then chooses a final food source with the highest probability related to its nectar number. If the fitness value of new one is higher than that of the previous one, the bee forgets the old one and memorizes the new position [5]. This is called as greedy selection. Then the employed bee whose food source has been exhausted becomes a scout bee to search for the further food sources once again.

In ABC, the solutions represent the food sources and the nectar quantity of the food sources corresponds to the fitness of the associated solution. The number of the employed and the onlooker bees is same, and this number is equal to the number of food sources [8]. Employed bees whose solutions cannot be improved through a predetermined number of trials, specified by the user of the ABC algorithm and called ―limit‖, become scouts and their solutions are abandoned [4], [6].

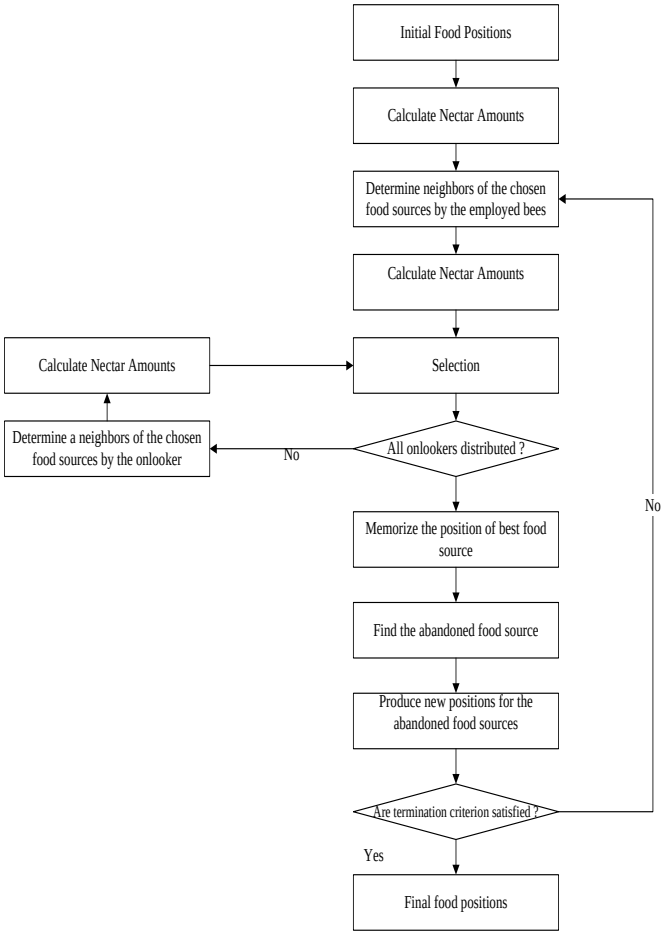


Fig. 3 Flow chart of Artificial Bee Colony Optimization Algorithm

4.3 Bacterial foraging

BFA is a newly introduced evolutionary optimization algorithm that mimics the foraging behavior of Escherichia coli (commonly referred to as E. coli) bacteria. [19] There are successful applications of BFA in optimization problems such as economic load dispatch and power systems. BFA models the movement of E. coli bacteria that thrive to find nutrient-rich locations in human intestine. An E. coli bacterium moves using a pattern of two types of movements: tumbling and swimming. Tumbling refers to a random change in the direction of movement, and swimming refers to moving in a straight line in a given direction. A bacterium in a neutral medium alternates between tumbling and swimming movements [19].

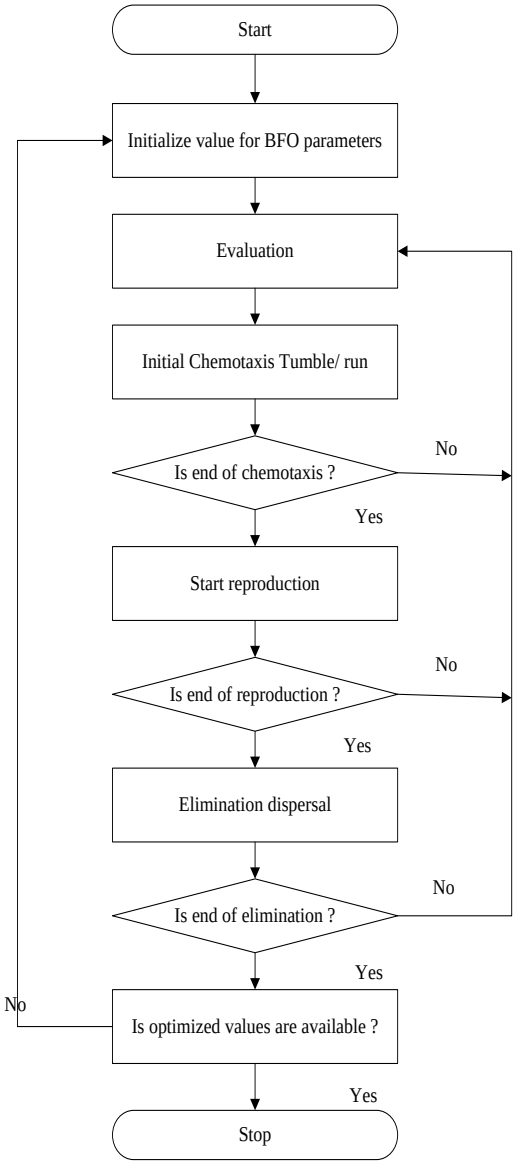


Fig. 4 Flow chart of Bacterial Foraging Algorithm

4.4 Bat algorithm

Xin-She Yang (2010) [3] propose the Bat Algorithm (BA), BA is inspired by the research on the social behavior of bats. The BA is based on the echolocation behaviour of bats. Microbats use a type of sonar (echolocation) to detect prey, avoid obstacles, and locate their roosting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species [3]. Based on the above description of bat process, Xin-She Yang proposes the Bat algorithm. The structure of the pseudo code of the Bat Algorithm is as follows [3]

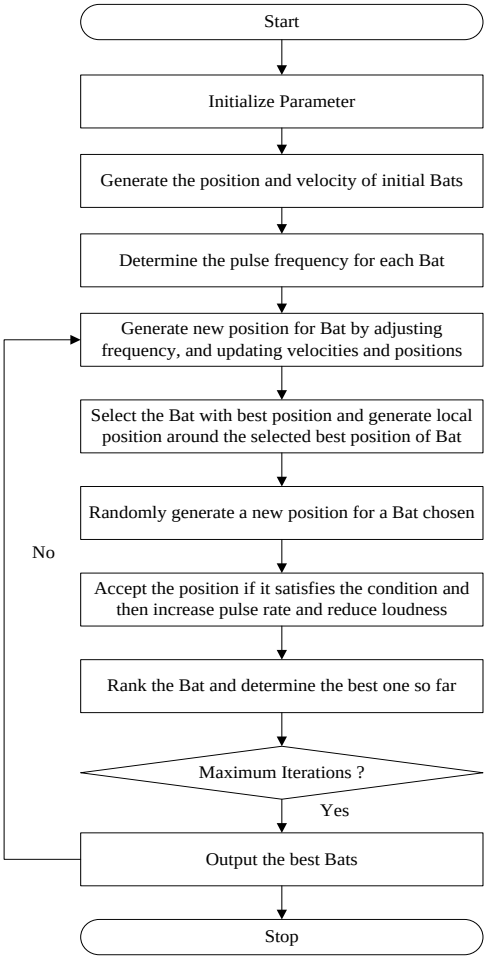


Fig. 5 Flow chart of Bat Algorithm

4.5 Cuckoo search

The CSA is introduced by yang and Deb in 2009 [11-12]. The CS was inspired by compel brood parasitism of cuckoo species by laying their eggs in the nests of host birds. Some cuckoos have evolved in such a way that female scrounging cuckoos can reproduce or rather imitate the colors and patterns of the eggs of a few chosen host species. This reduces the probability of the eggs being abandoned. Yang and dev also suggested in their research that levy flights is useful for improve solution quality besides on random walk method. A Levy flight is a random walk in which the step-lengths are distributed according to a heavy-tailed probability distribution. Each egg in a nest represents a solution .the objective it to employ good quality solution in the nest and replace those which are not so good solution. The algorithm based on three idealized rule:

• Each Cuckoo laid one egg only, and dumps the egg in a randomly chosen nest.

• The best nest (with quality solutions) will carry over the next generation.

• The number of available nest is fixed and a host can identify an alien egg with probability Pa [0, 1].

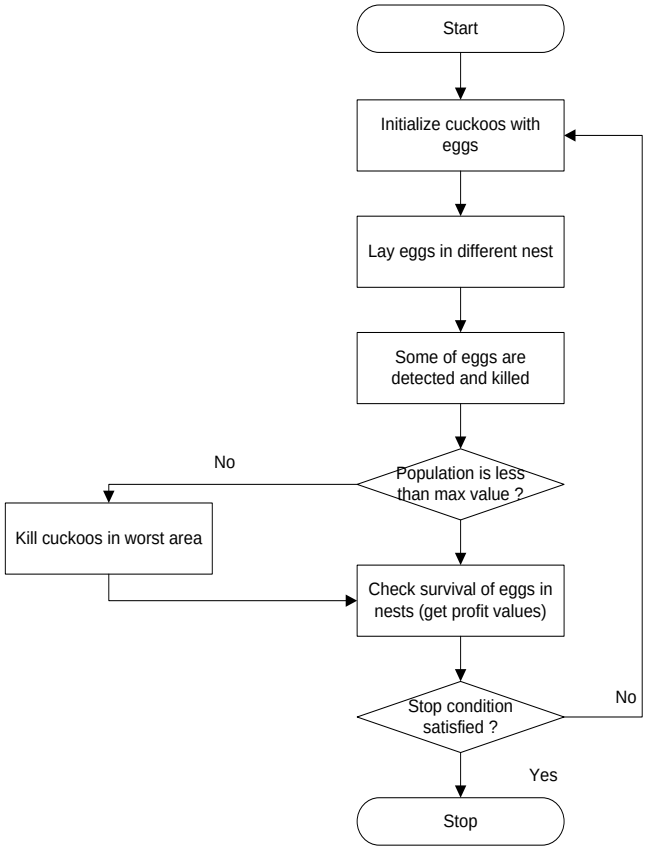


Fig. 6 Flow chart of Cuckoo Search Algorithm

4.6 Firefly

Fire flies, also called glowworms or lightning bugs are found all over the world. There are around two thousand firefly species and most fireflies produce short and rhythmic flashes of light. The pattern of flashes is often unique for a particular species. The fundamental functions of such flashes are to attract mating partners and preys. Females respond to male's unique pattern of flashing within the same species. The light emitting from their body strictly obeys the inverse square law i.e. as the distance between two flies increases, the intensity of light decreases. The air absorbs light, which becomes weaker and weaker as the distance increases. The bioluminescence from the body of the fireflies is due to 'luciferin', which is a heterocyclic compound.

Xin-She Yang [2] put forward firefly algorithm with inspiration from collective social behavior of fireflies promoted by communication through bioluminescence of characteristic different glittering patterns of flashes. It is another population based metaheuristic algorithm and used in nonlinear multimodal optimization in dynamic environment. The algorithm is formulated by assuming (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 decreases as the distance between them increases. (iii) If there are no fireflies brighter than a given firefly, it will move randomly. The brightness is associated with the objective function and the associated constraints along with the local activities carried out by the fireflies.[13]

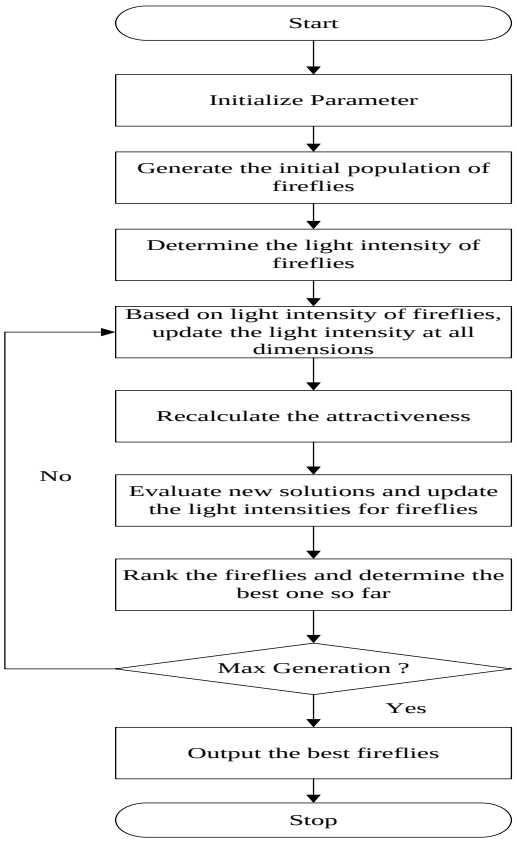


Fig. 7 Flow chart of Firefly Algorithm

4.7 Krill herd

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 [14,15].The significant mechanisms identified are feeding ability, protection from predators, enhanced reproduction and environmental condition.

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 [16, 17].Yet there are number of uncertainties about the mechanism that lead to distribution of krill herd [18].There are proposed conceptual models to explain observed formation of krill heard [19] 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 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.

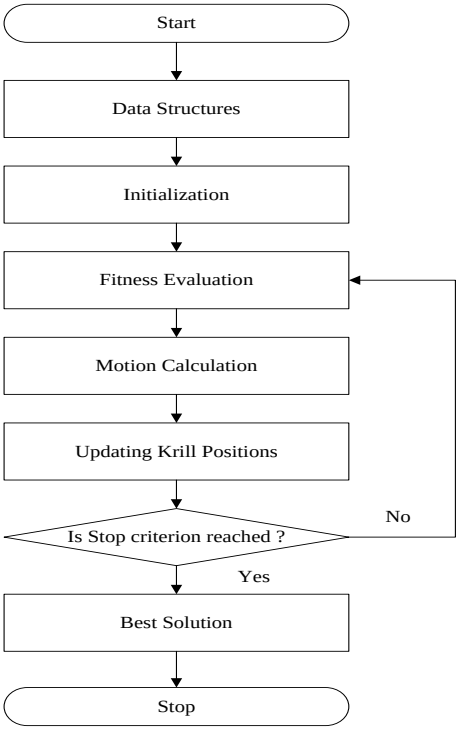


Fig. 8 Flow chart of Firefly Algorithm

Table I. A list of swarm intelligence algorithms

S.No.	Algorithm	Principle	Propounded By
1	**Ant Colony**	works on foraging behaviour of the ant for the food	Dorigo
2	**Artificial Bee Colony**	works on foraging behaviour of a honey bee	Karaboga and Basturk
3	**Bacterial Foraging**	works on behaviour of bacteria	Passino
4	**Bat Algorithm**	works on echolocation behaviour of microbats	Yang
5	**Cuckoo Search**	works on breeding behaviour of a cuckoos	Yang and Deb
6	**Firefly Algorithm**	works on lighting behaviour of firefly	Yang
7	**Krill Herd**	works on herding behaviour of krill individual	Gandomi and Alavi

V. CONCLUSION

Nature inspired algorithms are going to be a new revolution in the field of computer science, transportation engineering, mechanical engineering, management and so on. The scope of this area is really vast since as compared to nature, computer science problems are only a subset, opening a new era in next generation computing, modelling and algorithm engineering. This paper provides an overview of a range of nature inspired swarm Intelligence based optimization techniques. Generally speaking, almost all of the SI algorithms perform with heuristic population-based search procedures that incorporate random variation and selection. It has been witnessed that the applications and growth of natural computing in the last years is very drastic and has been applied to numerous optimization problems in computer networks, control systems, bioinformatics, data mining, game theory, music, biometrics, power systems, image processing, industry and engineering, parallel and distributed computing, robotics, economics and finance, forecasting problems, applications involving the security of information systems etc. Nature inspired computing still has much room to grow since this research community is quite young. There still remain significantly challenging tasks for the research community to address for the realization of many existing and most of the emerging areas in technology.

REFERENCES

[1]. Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Dušan FisterBbb(2013), A Brief Review of Nature-Inspired Algorithms for Optimization, Elektrotehni\v{s}ki vestnik, 80(3), 2013

[2] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, 2008.

[3] Xin-She Yang.: A New Metaheuristic Bat-Inspired Algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence Vol. 284 (2010) 65-74.

[4] Karaboga, D. Artificial Bee Colony Algorithm. Scholarpedia 2010,5,6915.Availableonline : http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm/ (accessed on 27 May 2011).

[5] Chaotic Bee Swarm Optimization Algorithm for Path Planning of Mobile Robots Jiann-Horng Lin and Li-Ren Huang Department of Information Management I-Shou University, Taiwan 2009

[6] An Effective Refinement Artificial Bee Colony Optimization Algorithm Based On Chaotic Search and Application for PID Control Tuning Gaowei YAN †, Chuangqin LI College of Information Engineering, Taiyuan University of Technology, Taiyuan, 030024, China

[7] Artificial bee colony (abc), harmony search and bees algorithms on Numerical optimization D. Karaboga, b. Akay Erciyes University, the dept. Of computer engineering, 38039, melikgazi, kayseri, turkiye

[8] Elitist artificial bee colony For constrained real-parameter optimization Efr´en mezura-montes member, ieee and ramiro ernesto velez-koeppel

[9] M. Marinaki, Y. Marinakis, and C. Zopounidis, "Honey bees mating optimization algorithm for financial classification problems," Appl. Soft Comput., vol. 10, pp. 806-812, 2010.

[10] Y. Marinakis, M. Marinaki, and G. Dounias, "Honey bees mating optimization algorithm for large scale vehicle routing problems," Nat. Comput., vol. 9, pp. 5-27, 2010.

[11] X S Yang and S Deb, Cuckoo Search Via Lévy Flights, IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 09), 2009, p. 210–214.

[12] X S Yang, Nature-Inspired Meta heuristic Algorithms, 2ndEdition, Luniver Press, 2010.

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

[14] 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.

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

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

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

[18] 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.

[19] Raghavendra V. Kulkarni, and Ganesh Kumar Venayagamoorthy, "Bio-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes", PART C: APPLICATIONS AND REVIEWS(to appear in issue 5, volume 40, 2010)

[20] K. M. Passino, ―Biomimicry of bacterial foraging for distributed optimization and control,‖ IEEE Control Syst. Mag., vol. 22, no. 3, pp.52–67, Jun. 2002

[21] S.He,Q.H.Wu, Senior Member IEEE and J.R Saunders.Anovel group search optimizer inspired by Animal Behavioral Ecology;2006,IEEE Congress on Evolutionary Computation,1272-1278

[22] Bonabeau, E., Dorigo, M. and Theraulaz, G.1999:Swarm intelligence. Oxford University Press

DEVELOPMENT OF ENZYMATIC BIOSENSOR FOR THE DETECTION OF AMMONIUM

[1]Ushmaben Chandrakantbhai Dave, [2]Murlidhar Meghwal ,

[3]Ravi Kumar Kadeppagari

[1,2,3] Food Science and Technology Division, Centre for Emerging Technologies, Jain University, Jain Global Campus, Karnataka, (India)

ABSTRACT

Monitoring of ammonium is essential since excess presence of it either in the air or aquatic environment or food samples is toxic. Use of biosensors for monitoring ammonium is advantageous due to their high selectivity and sensitivity. In the present study the enzyme alanine dehydrogenase was used for the development of ammonia biosensor since this enzyme is more stable and responds to wide range of ammonium concentrations. This enzyme converts pyruvate to alanine in the presence of ammonium and NADH. During this reaction two electrons will be generated and they will be transferred to the electrode and the signal will be amplified. Alanine dehydrogenase was isolated from Bacillus subtilis and it showed optimum activity at pH 8.5 and 35^{0}C. The enzyme was stable up to 45^{0}C and lost most of its activity at 50^{0}C. By using this enzyme a biosensor was developed by adopting amperometry. Here the working electrode coated with enzyme, reference electrode and counter electrodes were used and the signal generated at the working electrode was measured by using potentiostat. Here, we used screen printed electrode design since it's easy to carry and sensitive to low signal. The biosensor responded to the different concentrations of ammonium suggesting the suitability of this enzyme for the screen printed design of the biosensor.

Keywords : Alanine Dehydrogenase, Amperometry, Biosensor, Pyruvate, Bacillus Subtilis

I. INTRODUCTION

Higher levels of ammonium is hazardous to mammals and aquatic organisms. Elevated levels of ammonium in human blood is indicator of kidney disorder, stomach bacterial infection or liver dysfunction. Consumption of ammonium contaminated food can chronically elevate the concentration of the ammonium ion in the blood which can severely affect the function of brain, cause hyper excitability, coma, convulsions and finally death. Ammonium ions are responsible for aerosol formation and ammonium nitrate is highly used in landmine blasting and explosive preparation. Hence, the sensor which can monitor the ammonium will be helpful in environmental, food, health and defense applications. Metal oxide ammonia sensors are mostly based on SnO_2 and they operate based on the change of conductance due to the chemisorption of gas molecules on the sensing layer. The major drawback with these sensors is they are not selective to the particular gas. Optical sensors depend on the colorimetry or spectrophotometry. Indophenol blue based colorimetric method was used for determining the ammonium concentration. Here ammonium reacts with phenolic agent to form indophenol blue that absorbs light between 630- 720 nm under oxidizing conditions. Other optical sensors based on Berthelot's reaction [1], silver nanoparticles [2] and Nessler's reagent were developed for sensing ammonia. High-

performance liquid chromatography coupled with fluorescence detector was used for the ammonium detection. These conventional methods for the detection of ammonium are time consuming and tedious. In the colorimetric method, phenolic reagent and its by-product of the reaction are highly toxic. Spectrophotometric method will be interfered by other photoactive compounds and chromatographic methods require pre-column derivatization in order to treat the sample for detection. Both optical and chromatographic methods require big instrumentation for the detection of signal.

Potentiometric ammonia sensor based on zirconium titanium phosphate ion exchanger was developed [3]. It requires big instrumentation even though it has long life. An ammonium electrode based on the poly vinyl chloride (PVC) membrane containing palmitic acid and nonactin as an ammonium ionophore for the determination of ammonia was constructed [4]. However this sensor is not specific for the ammonia. Multiwalled carbonnanotube/copper composite paste electrode was developed for the detection of ammonia [5]. But reproducibility was very poor with this electrode. The major problem with above described sensors is they are not very specific to the ammonium and in this context biosensors become important since they depend on the highly specific bioreceptors. Biosensor based on the immobilized nitrifying bacteria (*Nitrosomonas europea*) was used for the detection of ammonia in waste waters. However, lifetime of this sensor is only 2 weeks since living organism itself was used as bioreceptor. Enzyme based biosensors were widely used since their stability can be easily manipulated using stabilizing agents or molecular approaches. Enzyme-based amperometric biosensors gained lot of importance in the last decade due to high selectivity of the bioreceptor and the sensitivity of electrochemical signal transduction. Two enzyme based sensor was constructed for the detection of ammonium by immobilizing glutamate dehydrogenase and glutamate oxidase on a Clark-type oxygen electrode [6]. But this sensor lost around 60% of its response after 18days. Another two enzyme based ammonia sensor was developed based on the glutamate dehydrogenase and diaphorase [7] . But it showed linear response only at lower concentrations of ammonium (2.5-500 µM). Two enzyme based sensors are difficult to manage since each enzyme needs different optimum conditions and it's difficult to maintain both the conditions simultaneously and it affects the stability of the sensor. Hence, single enzyme system i.e. alanine dehydrogenase (AlaDH) was used as a bioreceptor in the present work. This enzyme converts pyruvate to alanine and generates 2 electrons in the presence of ammonia and NADH. The reaction is written below.

$$\text{Pyruvate} + \text{NADH} + NH_4^+ \xrightleftharpoons{\text{AlaDH}} \text{L-alanine} + NAD^+ + H_2O + 2e^-$$

II.MATERIALS AND METHODS

2.1. Microbial growth and conditions

The bacterium, *Bacillus subtilis* (ATCC 6633) was maintained on the nutrient agar (pH 7.0) slants at 4^0C. For the production of AlaDH, the culture was grown at pH 7.0 and 37^0C by using nutrient broth supplemented with L- alanine (0.1 mg/ml) for 24 h.

2.2. Enzyme Assay

The assay mixture (1.2 ml) contained 0.125 mM NADH, 2 mM pyruvate, 100 mM NH_4Cl and 300 µl of appropriately diluted enzyme in the 50 mM Tris-HCl (pH 8.5) buffer. The assay was carried out at room

temperature by recording the decrease in absorbance of NADH at 340 nm. Enzyme activity units were measured as µmol of NADH hydrolyzed min^{-1}.

2.3. Stabilization studies

Different stabilizers like glycerol, EDTA, poly vinyl alcohol, poly vinyl acetate, poly vinyl pyrrolidone etc.were used for studying their effects on the enzyme activity at the storage temperature of 37^0C for different periods of time.

2.4. Enzyme Purification

Bacterial cells were lysed by sonification and the cell lysate was subjected to 80% ammonium sulfate precipitation and the precipitated protein was ran on DEAE-cellulose and Sephadex-G-100 chromatographic columns. The final sample was ran on 10% SDS-polyacrylamide gel to check the purity of the enzyme.

2.5. Biosensor construction and response studies

This biosensor was constructed based on amperometry. A mixture of polyHEMA, NADH, sodium pyruvate, EDTA and alanine dehydrogenase enzyme was prepared. The mixture was then deposited onto the carbon working electrode of screen printed electrode system and allowed to dry at 4^0C overnight. Carbon counter electrode and silver reference electrode were used in the system. The electrolyte (50 mM Tris-HCl, pH 8.5) containing different concentration of ammonium was placed on the electrode system such a way that it covers all 3 electrodes. Then the electric signal was measured with the help of potentiostat. Empty electrolyte was used as a control.

III. RESULTS AND DISCUSSION

This ammonia biosensor depends on the activity of AlaDH and this enzyme generates two electrons in the presence of ammonium and NADH while converting pyruvate to alanine. Generated electrons were detected by adopting amperometry since it's very sensitive technique and three electrodes were used during this process. They were working electrode, counter electrode and reference electrode. These electrodes were screen printed and coating of AlaDH along with other reagents on the working electrode will enhance the transfer of generated electrons to the working electrode and this will lead to the sensitive detection of electrons generated in response to the ammonia. The pure AlaDH is required for the construction of this ammonia biosensor in order to get high selectivity to the ammonium. Hence this enzyme was purified from *B.subtilis* and characterized.

For the production of AlaDH, a loopful of *B.subtilis* culture was inoculated in to the 100 ml nutrient broth (NB) containing L-alanine (0.1 mg/ml) and grown at 37^0C (pH 7.0, 150 rpm) for 15h. Four ml of inoculum was added to the 200 ml of NB broth with alanine and grown at the same conditions for 17h. Completely grown culture was pelleted by centrifuging at 5000 rpm for 20 min. Pellet was washed twice with 100 mM Tris-HCl buffer (pH 8.5) and resuspended in the same buffer containing PMSF and EDTA. Then cells were lysed by sonification (250 W, 30 KHz) and cell debris was removed by centrifuging at 10000 rpm. The supernatant was checked for the AlaDH activity by following the standard enzyme assay. For the purification of AlaDH, the supernatant was precipitated with 80% of ammonium sulphate and the precipitate was dissolved and dialysed against the Tris-HCl buffer (100 mM,pH 8.5) at 4^0C. The dialysed sample was loaded on to the DEAE-cellulose

ion exchange column and eluted with 0.2-1.0 M NaCl. The fractions showing the major enzyme activity were pooled together and dialysed against the buffer and ran on the Sephadex-G-100 column and the fractions showing major AlaDH activity were pooled and lyophilised. The purity of the enzyme was checked on the 10% SDS-polyacrylamide gel (Fig.1).

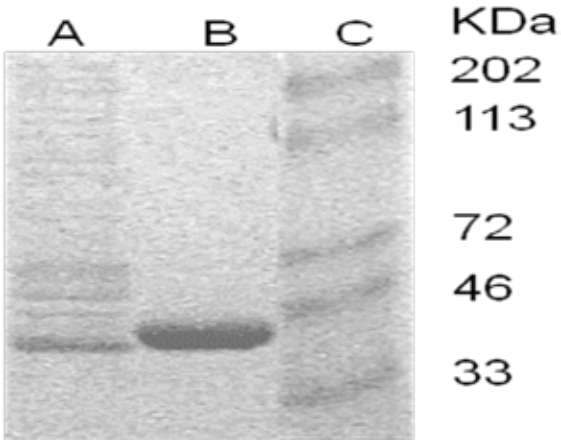


Fig.1. 10% SDS-polyacrylamide gel showing the purified protein (lane B). Lane A: crude extract,

The presence of single band (Fig.1, lane B) in the purified sample suggest the homogeneity of the AlaDH and this enzyme moves at the molecular mass of 42 KDa on the 10% SDS-polyacrylamide gel. The purified AlaDH showed the optimum pH of 8.5 (Fig.2) and the optimum temperature of 35^0C (Fig.3) when the effect of different pH and temperatures on the activity of the enzyme was studied. Enzyme lost its activity significantly at acidic pH and its activity started decreasing after pH 8.5 (Fig.2). Enzyme showed the increase in activity up to 35^0C and it remains more or less constant up to 45^0C and it showed drastic decrease in activity afterwards (Fig.3). Optimum enzyme activity around ambient temperatures is advantageous for the sensor since ammonium measurements will be done at ambient temperatures in real time.

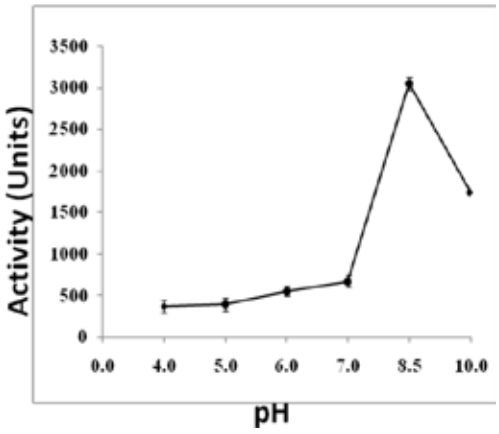


Fig.2. Graph showing the effect of pH on the activity of alanine dehydrogenase

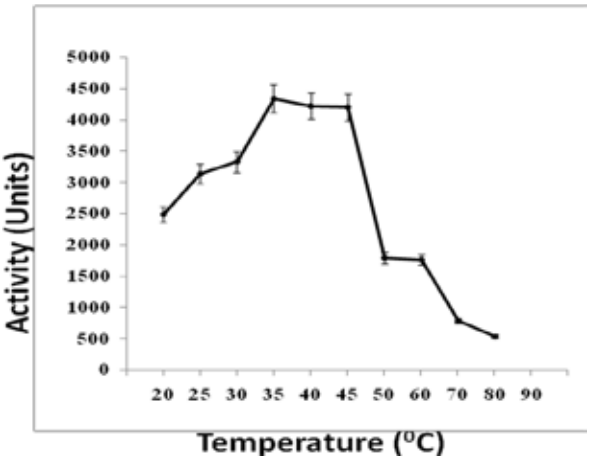


Fig.3. Graph showing the effect of temperature on the activity of alanine dehydrogenase

Studies on the storage stability of AlaDH for different periods around ambient temperature are required since this biosensor depends on the activity of AlaDH. When these studies were conducted in the presence of different stabilizers at 37^0C, the enzyme could retain the 97.9% and 76.44% of original activity in the presence of glycerol and EDTA respectively on the 30^{th} day of storage (Table 1). However, in the absence of stabilizer the enzyme could retain only 68.08% activity on the 30^{th} day and other stabilizers (poly vinyl alcohol, poly vinyl acetate and poly vinyl pyrrolidone) could also retain the more or less same kind of activity. On the 75^{th} day, the enzyme alone could retain only 6.86% of original activity, whereas in the presence of glycerol and EDTA it could retain 43.97% and 46.76% activity respectively (Table 1). These results suggest that glycerol and EDTA acted as good stabilizers for AlaDH. However, calcium chloride, sucrose, poly ethylene glycol, glycine and tween 80 acted as inhibitors for AlaDH (data not shown).

Table 1: Effect of stabilizers on the storage stability of alanine dehydrogenase at 37°C

S. No.	Stabilizer	Residual activity (%) on				
		0 day	15^{th} day	30^{th} day	45^{th} day	75^{th} day
1	Control	100	70.44	68.08	63.45	6.86
2	EDTA	100	87.72	76.44	65	46.76
3	Glycerol	100	85.14	97.9	60.11	43.97
4	Poly vinyl alcohol	100	65.75	62.28	58.99	3.33
5	Poly vinyl acetate	100	63.49	60.11	49.72	5.80
6	Poly vinyl pyrrolidone	100	64.22	62.15	60.66	7.43
7	Poly vinyl alcohol + Poly vinyl pyrrolidone	100	71.41	69.75	67.71	7.12

8	Poly vinyl alcohol+ Poly vinyl acetate	100	67.85	77.92	66.41	9.45

Ammonia biosensor was constructed by coating a layer of solution containing AlaDH, polyHEMA, NADH, sodium pyruvate and the stabilizer EDTA on the working electrode of the screen printed electrode system. After drying, the electrolyte containing different concentrations (0-100 mM) of ammonium was placed on the working electrode and the electric signal generated at the working electrode was detected by plotting cyclic voltagram with the help of potentiostat. When there is no ammonium, the cyclic voltagram was at basal level (Fig.4, CV 1) and it didn't show any signal, whereas at 5 mM ammonium the cyclic voltagram was above the basal level (Fig.4, CV 2) and the signal proportionately increased with the increasing concentrations of ammonium (Fig 4; CV 3, 4, 5 and 6 correspond to 10, 25, 50 and 100 mM ammonium respectively) suggesting the biosensor response to the different ammonium concentrations.

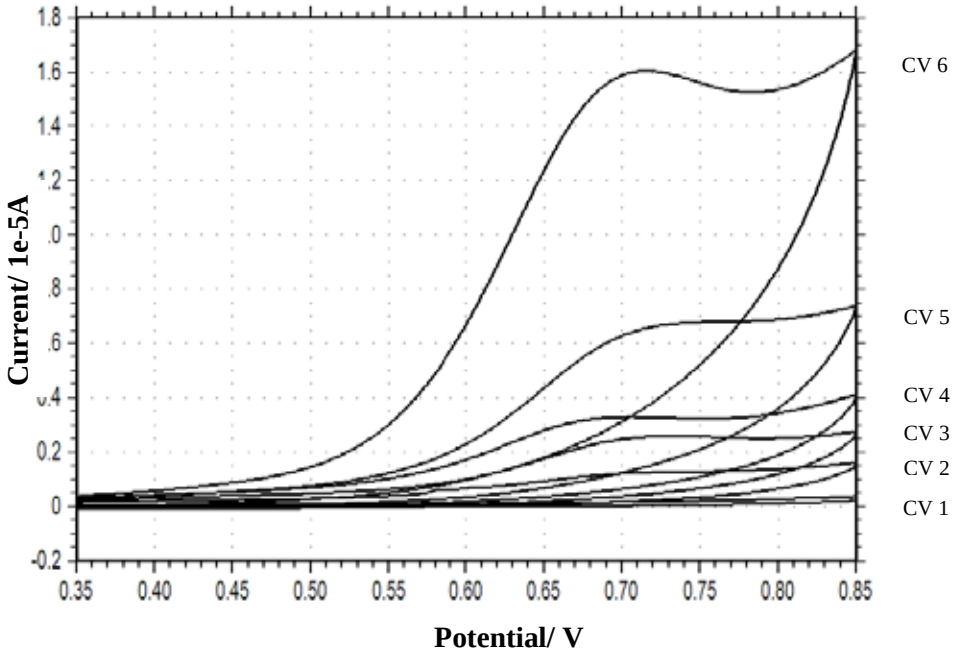


Fig.4. Cyclic voltagrams of biosensor in response to the different concentrations of ammonium. CV 1, 2, 3, 4, 5 and 6 are the cyclic voltagrams corresponding to the 0, 5, 10, 25, 50 and 100 mM ammonium.

IV.CONCLUSIONS

The developed biosensor for the detection of ammonia works at ambient conditions. Improvising the biosensor for working at extreme conditions could be advantageous, even though ammonia levels were commonly checked at ambient conditions. The electric signal obtained while detecting the ammonium was in terms of 10^{-5} amps and this signal should be improved in order to calibrate the biosensor for wide range of ammonium concentrations. Nevertheless, this sensor clearly differentiated the ammonium concentrations in the range of 0-100 mM. Stability of this sensor after different storage periods need to be checked even though this study was carried out for the bioreceptor, AlaDH.

V. ACKNOWLEDGEMENTS

Authors thank Dr. Chenraj Roychand, President, Jain University Trust and Dr. Krishna Venkatesh, Director and Dean, Centre for Emerging Technologies, Jain University for providing the support and facilities. Authors are thankful to Science and Engineering Research Board, New Delhi and National Programme on Micro Air Vehicles for their financial support.

REFERENCES

[1] K.T. Laua, S. Edwards and D. Diamond, Solid-state ammonia sensor based on Berthelot's reaction, *Sensors and Actuators B, 98*, 2004, 12–17.

[2] S.T. Dubas and V. Pimpan, Green synthesis of silver nanoparticles for ammonia sensing, *Talanta*, *76*, 2008, 29–33.

[3] S.S.M. Hassan, S.A. Marei, I.H. Badr and H.A. Arida, Novel solid-state ammonium ion potentiometric sensor based on zirconium titanium phosphate ion exchanger, *Analytica Chimica Acta*,*427*, 2001, 21–28.

[4] E.Karakuş, S. Pekyardimci and E. Kiliç, A new potentiometric ammonium electrode for biosensor construction, *Artif Cells Blood Substit Immobil Biotechnol, 34(5)*, 2006, 523-534.

[5] F. Valentini, V. Biagiotti, C. Lete, G. Palleschi and J. Wang, The electrochemical detection of ammonia in drinking water based on multi-walled carbon nanotube/copper nanoparticle composite paste electrodes, *Sensors and Actuators B, 128*, 2007, 326–333.

[6] R.C.H.Kwan, P.Y.T. Hona and R. Renneberg R, Amperometric determination of ammonium with bienzyme/poly(carbamoyl) sulfonate hydrogel-based biosensor, *Sensors and Actuators B, 107*, 2005, 616–622.

[7] S.A.Rahman, J. Abdullah, H.Sidek and N.E. Azmi, Mediated Amperometric Biosensor for the Determination of Ammonium, *Analytical and Bioanalytical Electrochemistry*, 4, 2012, 262 – 276.