A SIMPLE PROCEDURE FOR EXTRACTING QUADRATICS FROM A GIVEN ALGEBRAIC POLYNOMIAL.

S.N.SIVANANDAM
Professor and Head: Department of CSE
PSG College of Technology
Coimbatore, TamilNadu, India 641 004.
sns@mail.psgtech.ac.in, snsivanandam@yahoo.co.in

RAMANI BAI.V
Assistant Professor: Department of CSE
Saintgits College of Engineering
Kottayam, Kerala, India
ramanibaiv@saintgits.org, ramanisivan@gmail.com

***Abstract*—Any linear time-invariant continuous or discrete system can be represented either in terms of characteristic algebraic polynomial or in the form of open-loop transfer function. The open-loop transfer function will consist of numerator polynomial and denominator polynomial. To understand the transient characteristics of the system, the denominator polynomial is analysed for its roots. Any nth order polynomial will have $\frac{n}{2}$ quadratics if n is even, there will be $\frac{n-1}{2}$ quadratics if n is odd. Various procedures are available for extracting quadratics, each having its own merits and limitations. In this paper, a simple-novel scheme is proposed for extracting quadratics for even order polynomial.**

***Keywords-Coefficients - Quadratic factors - Polynomials – Synthetic Division – Bisection Principle -Iterations – Convergence*.**

I. INTRODUCTION

Developments of numerical methods to solve algebraic polynomials are very much essential because the analytical method fails to solve the polynomial equations of degree greater than four.

Most of the methods, used to solve an algebraic polynomial are based on iterative techniques. Different numerical methods such as, Newton-Rapson [1], Muller [2], Graeffe's root-squaring method [2], Lehmer's Method, Birge-Vieta [2], Bairstow [3],[4] and Laguerre's method [2], etc are available to solve the polynomials. But, each method has some advantages and disadvantages over another method. Generally, the following aspects are considered to compare the methods: convergence or divergence, rate of convergence, applicability of the method, computational complexity of the method, and amount of pre-calculations needed before application of the method.

The iterative algorithms for obtaining the roots of an equation are essentially begin with the initial approximation; the algorithms then provide a sequence of iterates converging to a root in the limit. This initial approximation should be sufficiently close to one of the roots; otherwise the iterates may diverge. Hence, some idea about the location of root is important [6-20].

From the fundamental theorem of algebra [17], it is known that every polynomial of degree n has exactly n roots in the complex plane. Furthermore, if the coefficients of the polynomial are real, then complex roots appear in conjugate pairs. Complex roots can be easily determined closed form formula [2] from the quadratic factors of the polynomial. Bairstow method [3],[4], extracts quadratics from the polynomial, thus solves it. The present work follows the same direction but involves simple procedure to extract initial quadratic approximation. Since the initial approximation is found closer to the actual quadratic, then the procedure does a simple iterative process to refine this approximation unlike the high computational iterative process involved in Bairstow method.

II. PPROPOSED SCHEME

Consider a monic polynomial,

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0.\ a_i > 0. \qquad (1)$$

where n is even degree and a_is' are real coefficients with $a_n=1$.

The roots of the polynomial equation can be determined by extracting a quadratic factor from the polynomial. The roots of a quadratic equation can be determined using a known closed form formula [1].

Let x^2+px+q be a factor of equation (1). If equation (1) is divided by the factor x^2+px+q then we obtain a polynomial $Q_{n-2}(x)$ of degree (n-2) and a remainder $R(x) = (e_1x+e_2)$ of degree one, where e1 and e2 are independent of x. Thus the polynomial $P_n(x)$ can be written as

$$P_n(x) = (x^2+px+q)\, Q_{n-2}(x) + R(x). \qquad (2)$$

$$\text{where } Q_{n-2}(x) = b_n x^{n-2} + b_{n-1} x^{n-3} + \ldots\ldots + b_2;\ b_n=1. \qquad (3)$$

The values of e_1 and e_2 depend on p and q. If x^2+px+q is a factor of $P_n(x)$, then e_1 and e_2 should be zero. Thus our aim is to determine the values of p and q such that e1=0 and e_2=0.

2.1 *New Scheme for Initial Quadratic Factor*

Without the loss of generality and for simplicity, we introduce a heuristic strategy to extract the initial approximate quadratic factor by a triangle formation from the coefficients of the polynomial $a_n, a_{n-1}, a_{n-2\ldots}\ a_1, a_0$ as shown in the Table I.

Every element in each row is determined by

$a_{ij} = a_{(i-1)(j)} + a_{(i-1)(j-1)}$.
: the row index i = 1, …. n
: the coefficient index j = n, n-1, …,0. (4)

The 0th row of the triangle is formed with the coefficients of the polynomial. The each element in the first row of the triangle is formed as

$a_{1j} = a_{0j} + a_{0(j-i)}$: j = n, n-1,…0.

This reduces the degree of the polynomial to (n-1). The elements in the second row are determined as

$a_{2j} = a_{1j} + a_{1(j-1)}$: j = n-1, n-2,…0.

Similarly, the elements in the third row are determined as

$a_{3j} = a_{2j} + a_{2(j-1)}$: j = n-2, n-3,…0.

And so on, until j is reduced to zero.

The initial approximate quadratic factor is extracted at the (n-2)th row. i.e. the elements $a_{(n-2)2}$, $a_{(n-2)1}$ and $a_{(n-2)0}$ form the coefficients of the quadratic factor as

$D(x) = a_2x^2+a_1x+a_0$. (5)

The above equation (5) is scaled by a_2 as

$= x^2+(a_1/ a_2)x+ (a_0/ a_2)$.

Let the initial approximate quadratic factor be

$D(x) = x^2+p_0x+q$. (6)

where $p_0 = (a_1/ a_2)$ and $q_0=(a_0/ a_2)$.

2.1.1) *Algorithm for Initial Quadratic Factor*

The computation involves only two rows. The second row is determined from the first row and the first row is updated with the second row. We deduce the pseudocode of the algorithm for this scheme as:

```
Algorithm Init_Quad(P_n(x),n)
/*The algorithm extracts the initial approximate quadratic factor from the given polynomial $P_n(x)=a_nx^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} +…+a_1x+a_0$ , where n is the even degree and $a_n, a_{n-1}, a_{n-2}, …, a_1$ and $a_0$ are positive coefficients.*/
for i = 0 to n-2 do
    for j = n-i to 0 do
        Compute bj = aj + aj-1;
                aj = bj;
    end for
end for
Compute p = $a_1/ a_2$ ;
        q = $a_0/ a_2$;
Write p, q;
End Init_Quad.
```

The algorithm performs n(n+1)/2 additions only. The procedure is terminated at (n-2)th row to obtain the three elements to form the initial quadratic, thus saves three addition operations. Therefore, the total addition operations performed by this procedure is ((n(n+1)/2)-3); assume addition operation takes unit time.

2.2 *Quadratic Synthetic Division*

The division of the polynomial equation (1) by the initial quadratic factor equation (6) is carried out by quadratic synthetic division scheme as shown in the Table II.

Hence the quotient $Q_{n-2}(x) = b_nx^{n-2}+b_{n-1}x^{n-3}+……+b_2$ and the remainder $R(x)=b_1x+b_0$ are determined from the Table II. The remainder coefficients $e_1=b_1$ and $e_2=b_0$ are errors caused due to the approximation of p_0 and q_0 values.

2.3 *Error Correction and Convergence*

Let (p_t, q_t) be the true values of p and q and Δp and Δq the corrections to p and q.

Then $p_t = p + \Delta p$ and $q_t = q+ \Delta q$.

Let $\Delta p = e_1/n$ and $\Delta q = e_2/n$. (7)

Therefore, the improved values of p and q are p + Δp and q+ Δq.

Then if p_0, q_0 be the initial values of p and q then the improved values are $p_1= p_0+ \Delta p$ and $q_1= q_0+ \Delta q$.

Once $p_1$and q_1 are evaluated, the synthetic division is repeated, and the next improved p_2 and q_2 are determined from the relation $p_2 = p_1+ \Delta p$ and $q_2 = q_1+ \Delta q$.

In general,

$p_{k+1}=p_k+ \Delta p$
$q_{k+1}=q_k+ \Delta q$. (8)

The values of Δp and Δq are determined at $p=p_k$ and $q=q_k$.

Any sign changes from p_k to p_{k+1} or q_k to q_{k+1} is observed in each iteration and then bisection technique is employed to enhance the convergence rate.

The midpoint is found to update p_{k+1} and q_{k+1} values as

$p_{k+1}= (p_{k+1}+ p_k)/2$
$q_{k+1}= (q_{k+1}+q_k)/2$. (9)

The repetition is to be terminated when p and q have been obtained to the desired accuracy.

The convergence rate for p and q are determined

Table I: Extracting the initial approximate quadratic factor

n	a_{0n}		$a_{0(n-1)}$		$a_{0(n-2)}$		$a_{0(n-3)}$	…		a_{01}		a_{00}
		+		+		+					+	
n-1		$a_{1(n-1)}$		$a_{1(n-2)}$		$a_{1(n-3)}$		…	a_{11}		a_{10}	
			+		+					+		
n-2			$a_{2(n-2)}$		$a_{2(n-3)}$					a_{20}		
:			:		:			…		:		
:				:		:		…	:			
2				$a_{(n-2)2}$		$a_{(n-2)1}$		$a_{(n-2)0}$				
					+		+					
1					$a_{(n-1)1}$		$a_{(n-1)0}$					
						+						
0						a_{n0}						

Table II: Quadratic Synthetic Division

	1	a_{n-1}	a_{n-2}	…	…	a_k	…	…	a_1	a_0
$-p_0$		$-p_0$	$-p_0b_{n-1}$	…	…	$-p_0b_{k-1}$	…	…	$-p_0b_2$	
$-q_0$			$-q_0$	…	…	$-q_0b_{k-2}$	…	…	$-q_0b_3$	$-q_0b_2$
	1	b_{n-1}	b_{n-2}	…	…	b_k	…	…	b_1	b_0

from equation (8) as

$\Delta p_k = p_{k+1} - p_k$

$\Delta q_k = q_{k+1} - q_k$

Let $p_{k+1} = g(p_k)$, $k = 0,1,2,\ldots$

$q_{k+1} = g(q_k)$, $k = 0,1,2,\ldots$

Let $g(p_k) = p_k + (e1/n)$

$g(q_k) = q_k + (e1/n)$

Therefore $g'(p_k) = g'(q_k) = 1$.

Hence, the scheme converges linearly.

2.4 *The Deflated Polynomial*

The polynomial

$Q_{n-2}(x) = P_n(x) / (x^2+px+q)$

$= b_nx^{n-2}+b_{n-1}x^{n-3}+\ldots\ldots+b_3x+b_2$; $b_n=1$. (10)

$Q_{n-2}(x)$ is called the deflated polynomial. The next quadratic factor can be obtained in the similar process from the deflated polynomial.

2.5 *Algorithm for Extracting all Quadratics*

The algorithm which extracts a quadratic factor from a polynomial of degree n and also determines the deflated polynomial is proposed as:

```
Algorithm Quad_Poly($P_n(x)$,n)
/* Extracts a quadratic factor $x^2+px+q$ from a polynomial $P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x+a_0$ of degree n and determines the deflated polynomial $Q_{n-2}(x)=b_nx^{n-2}+b_{n-1}x^{n-3}+\ldots + b_3x +b_2$.*/
Read n, $a_n, a_{n-1}, a_{n-2}, \ldots, a_1, a_0$;
// The degree and Coefficients
Set $\xi = 10^{-7}$;        // Error Tolerance
// Extract the Initial Quadratic Factor
1. Call Init_Quad($P_n(x)$,n);
/* Polynomial division by synthetic scheme and bisection technique for convergence.*/
2. for k=(n-1) to 0 do
        Compute $b_k=a_k-pb_{k-1}-qb_{k-2}$;
   end for
   Compute $\Delta p = b_1/n$;
           $\Delta q = b_0/n$;
   Compute $p_{new} = p + \Delta p$ ;
           $q_{new} = q + \Delta q$;
   if (sign(p) $\neq$ sign($p_{new}$) or
       sign(q) $\neq$ sign($q_{new}$)) then
        Compute $p_{new} = ( p_{new} + p ) / 2$ ;
                $q_{new} = ( q_{new} + q ) / 2$ ;
   end if
   if ( ( $|p_{new} - p| > \xi$ ) or ( $|q_{new} - q| > \xi$ ) ) then
        Set p = $p_{new}$ ;
            q = $q_{new}$ ;
        goto 2;
   else
        Print "The values of p and q are" , $p_{new}$, $q_{new}$;
        Print "The coefficients of the deflated polynomial are", $b_n, b_{n-1}, \ldots\ldots, b_2$;
   end if
3. Set n=n-2;
End Quad_Poly.
```

2.6 *Generalised Algorithm for Solving Polynomial of any degree*

The proposed procedure satisfies the following cases also:

Case 1: Odd degree

If the n is odd, then $P_n(x)$ is multiplied by the single factor (x+1) to convert it into even degree polynomial. Multiplying the equation(1) with (x+1), we get

$(x+1)P_n(x)=a_{n+1}x^{n+1}+a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x+a_0$ (11)

The algorithm is written as

```
Algorithm Even_Poly($P_n(x)$,n)
for i=0 to n do
    Set c[i]=a[i];
end for
for i=0 to n do
    Compute a[i+1] = a[i+1]+c[i];
end for
Set n=n+1;
End Even_Poly
```

Case 2: Negative Coefficients

If any coefficient a_i $(0 \leq i \leq n)$ is found negative, then the following procedure is to be followed before starting the step 1 in the algorithm Quad_Poly.

Let $S = [a_n+a_{n-1}+a_{n-2}\ldots a_1]$

where S is the sum the coefficients except a_0.

If $a_0 \geq S$, then inverse the polynomial.

Now the equation (1) becomes

$P_n(1/x) = a_n(1/x)^{n+1}+a_{n-1}(1/x)^{n-1} +\ldots..+a_0$ (12)

$= a_0x^n+ a_1x^{n-1}+\ldots..+a_{n-1}x+a_n$ (13)

Scale the equation (13) by a_0 to convert it into monic polynomial as

$= x^n +(a_1/a_0)x^{n-1}+\ldots+(a_{n-1}/ a_0)x+(a_n/ a_0)$. (14)

The equation (14) can be easily determined from equation (12) by the following procedure.

The equation (14) can be easily determined from equation (12) by the following procedure.

```
Algorithm Inverse_Poly(P_n(x),n)
/*This algorithm inverses the polynomial and scale it to convert into monic polynomial*/
for i=0 to n do
        Set b[n-i]=a[i];
end for
for i= 0 to n do
        Compute a[i]=b[i]/b[n];
end for
End Inverse_Poly
```

Thus we develop an algorithm which extracts all quadratics of polynomial of any degree with real coefficients as

```
Algorithm Solve_Poly(P_n(x),n)
/*Solves a polynomial $P_n(x) = a_nx^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} +…+a_1x+a_0$ of any degree n by extracting (n/2) quadratic factors $x^2+px+q$.*/
Read n, $a_n, a_{n-1}, a_{n-2} …  a_1, a_0$;
// The degree and Coefficients
Set $\xi = 10^{-7}$;                // Error Tolerance
if n is odd then
        Call Even_Poly(P_n(x),n);
end if
if $a_i<0$ $(0 \le i \le n)$ then
        Call Inverse_Poly(P_n(x),n);
end if
while(n>2)
        Call Quad_Poly(P_n(x),n);
End Solve_Poly.
```

This procedure is simple and straightforward to be implemented using 'C' language.

III ILLUSTRATIONS

3.1 Illustration 1

Consider the given polynomial to be:

$$P(x) = x^8+9x^7+39x^6+103x^5+183x^4+227x^3+205x^2+133x+60. \qquad (15)$$

The degree of the above polynomial is even and all the coefficients are positive. All the quadratics are extracted, thus the polynomial is solved using the proposed algorithm Quad_Poly(Pn(x),n) as explained in 2.5. The results are tabulated in the Table III. The execution time is measured for solving the polynomial (15) in terms of milliseconds.

3.2 Illustration 2

Consider the polynomial

$$P(x) = x^4+6.2x^3+13.915x^2+13.33x+4.62 \qquad (16)$$

The equation has even degree and positive fractional coefficients. The polynomial is solved using the algorithm 2.5 Quad_Poly and the results are tabulated in Table IV.

The two quadratic factors and four cluster roots are obtained in 0.156 milliseconds using the algorithm 2.5.

3.3 Illustration 3

Consider the polynomial

$$P(x) = x^4+4x^3-7x^2-22x+24 \qquad (17)$$

The order of the polynomial equation (17) is even and it has negative coefficients. It is found that the coefficient a_0 is greater than the sum of other coefficients i.e. $(a_4+a_3+a_2+a_1)$. In this case, the algorithm 2.6 Inverse_Poly is applied to solve this above polynomial (17). And the results are tabulated in the Table V.

The polynomial (17) is inverted and the roots of the inverted extracted quadratics are obtained. The roots are again inverted to get the actual roots. This task is accomplished in 0.171milliseconds.

VI DISCUSSIONS

The observations made from the illustrations 3.1, 3.2 and 3.3 are discussed here. The execution times measured in illustrations 3.1 and 3.2 are comparable with Bairstow method. The initial approximate quadratic is obtained by the proposed scheme from the original given polynomial. Then quadratic synthetic division is carried out and error correction is applied to refine the original guess quadratic. Thus, the actual quadratic factor and the deflated polynomial are derived. Again the initial approximate quadratic is obtained from the deflated polynomial and the process is continued until all the quadratics are extracted. This guarantees that the method converges on accurate results. The bisection technique deployed during error correction process enhances the

Table III: Solving the Polynomial equation (15)

Polynomial P(x)	*Initial Approximate Quadratic $D_a(x)$*	*Deflated Polynomial Q(x)*	*Extracted Quadratic Factor D(x)*	*Roots of D(x)*	*Total Exe.Time (ms)*
$x^8+9x^7+39x^6+103x^5+183x^4+227x^3+205x^2+133x+60$	$x^2+1.456931x+1.693525$	$x^6+6x^5+17x^4+28x^3+31x^2+22x+15$	x^2+3x+4	$-1.5 \pm j1.322878$	0.307
	$x^2+1.437037x+1.548148$	$x^4+4x^3+6x^2+4x+5$	x^2+2x+3	$-1 \pm j1.414214$	
	$x^2+1.333333x+1.266667$	x^2+4x+5	x^2+1	$0 \pm j1$	
			x^2+4x+5	$-2 \pm j2$	

Table IV : Solving the Polynomial equation (16)

Polynomial P(x)	*Initial Approximate Quadratic $D_a(x)$*	*Deflated Polynomial Q(x)*	*Extracted Quadratic Factor D(x)*	*Roots of D(x)*	*Total Exe. Time (ms)*
$x^4+6.2x^3+13.915x^2+13.33x+4.62$.	$x^2+1.733797x+1.654705$	$x^2+4.1x+4.2$	$x^2+2.1x+1.1$	-1,-1.1	0.156
			$x^2+4.1x+4.2$	-2,-2.1	

convergence rate. But, even though Bairstow method has high convergence rate, in some cases it is observed that it may converge on inaccurate results.

In illustration 3.3, it is found that the polynomial has negative coefficients. Then the coefficient value a_0 i.e. 24 is compared with the sum of other coefficients which is -24. i.e. a_0 is not lesser than the sum of remaining coefficients. Thus, the polynomial is inverted and the proposed scheme is applied. In order to get the actual roots, the obtained roots are again inverted.

V CONCLUSION

The proposed method is reliable in finding both real and complex roots of a given polynomial up to the accuracy of 10^{-7}. Since the proposed method extracts initial approximate quadratic from the given polynomial itself that enhances the quadratic synthetic division to converge on accurate results. The method is also suitable for odd degree polynomials and polynomials having negative coefficients. In case of odd degree polynomial, the given polynomial is multiplied with (x+1) factor to convert it in to even degree, and then the proposed scheme is applied. Finally, the root x = -1 must be eliminated from the obtained list of roots. For the ill-conditioned polynomial [18],[19] such as polynomial having very large coefficients, it is suggested that the polynomial can be inverted and scaled, and then the proposed method can be used to extract all the quadratics. The illustrative examples show that the method is very simple to be implemented using any programming language.

REFERNECE

[1] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, “Numerical Recipes in C: The Art of Scientific Computing,” MA: Cambridge University Press, pp. 279-282, (1988).

[2] A. Ralston,”First Course in Numerical Analysis,” New York: McGraw-Hill, pp. 362-371, (1965).

[3] D.W.Arthur,”Extension of Bairstow’s Method for Multiple Quadratic Factors,” IMA Journal of Applied Mathematics, 9(2), pp.194-197, (1972).

[4] Herbert E.Salzer,”Some Extensions of Bairstow’s Method”, Journal of Numerische Mathematik, Vol 3, pp.120-124, (1961).

[5] Hyung W.Paik, “C Program for Finding the Roots of Real-Coefficient Polynomials,” IEEE Transactions on Education, VOL. 37, NO.1, pp 72, (1994).

[6] S.N.Sivanadam, S.N.Deepa, “A Novel Approach for Root Distribution Analysis of Linear Time-Invariant Systems Using Routh and Fuller Tables,” Asian Journal of Control, Vol. 11, No.3, pp.1-10, (2009).

[7] Takehiro Mori, “Note on the Absolute value of the Roots of a polynomial,” IEEE Transaction on Automatic Control, Vol.AC-29, No.1, pp.54-56, (1984).

[8] Gray A.Sitton, C.Sidney Burrus, James W.Fox, and Sven Treitel, “Factoring Very-High-Degree Polynomials,” IEEE Signal Processing Magazine, pp.28-42, (2003).

[9] F.Kraus,L.Guzzella,A.Astolfi and R.Tempo, “Invarient Roots of Polynomials with uncertain Coefficients”, IEEE proceedings of 32nd conference on Decision and Control,pp.498-499, (1993).

[10] Chyi Hwang, Shih-Feng Yang, “The Robust Root Locus of Polynomial Families with Multilinear Parameter Dependence,” Proceedings of IEEE International Conference on Control Applications. pp.847-852, (2001).

[11] H. W. Lenstra, “Algorithms in algebraic number theory,”Bulletin of the AMS, 26:211-244, (1992).

[12] A. K. Lenstra, H. W. Lenstra, and L. Lovisz, “Factoring polynomials with rational coefficients,” Math.Ann., 261:515-534, (1982).

[13] Thomas A. Rice and Leah H. Jamieson, “A Highly Parallel Algorithm

Table V : Solving the Polynomial equation (17)

Polynomial P(x)	*Initial Approximate Quadratic $D_a(x)$*	*Deflated Polynomial Q(x)*	*Extracted Quadratic Factor D(x)*	*Roots of D(x)*	*Total Exe. Time (ms)*
$x^4+4x^3-7x^2-22x+24$ $P(1/x)=x^4-0.916667x^3-0.291667x^2+0.166667x+0.041667$	$x^2+1.185185x-0.074074$	$x^2-1.5x+0.5$	$x^2+0.58333x+0.083333$	-0.25, -0.333333 Inverted Roots: -4 , -3	0.171
			$x^2-1.5x+0.5$	1, 0.5 Inverted Roots: 1, 2	

for Root Extraction," IEEE Transactions on Computers, Vol. 38, No. 3, pp-443-449, (1989).

[14] S. M. Pizer, "Numerical computing and mathematical analysis, Science Research Associates," Inc., Palo Alto, CA, pp. 231-243, (1977) .

[15] E. Bach, G. Miller, and J. Shallit, "Sums of divisors, perfect numbers, and factoring," Proceedings of the 16th ACM Symposium on Theory of Computing, pp.183-190, (1984).

[16] Hamming, Richard W, "Numerical Methods for Scientists and Engineers," 2nd edition, McGraw-Hill Book Company, New York, (1973).

[17] W. S. Burnside, "The Theory of Equations", vol. 2. New York, Dover Publications, (1970).

[18] V.Y. Pan, "Solving a polynomial equation: Some history and recent progress," SIAM Rev., vol. 39, no. 2, pp. 187–220, (1997).

[19] V.Y. Pan, "Solving polynomials with computers," Amer. Sci., vol. 86, no. 1, pp. 62–69, (1998).

[20] M. A. Jenkins and J. F. Traub, "A three-stage algorithm for real polynomials using quadratic iteration," SIAM J. Nurner. Anal., pp.545-566, (1970).

AUTHORS PROFILE

S. N. Sivanandam received the Ph.D degree in Electrical Engineering from Madras University, Chennai in 1982. He is currently Professor and Head, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore. He has a total teaching experience (UG and PG) of 41 years. He has published 12 books. He has delivered around 150 special lectures of different specialization in Summer/Winter school and also in various Engineering Colleges. He has guided and co-guided 30 Ph.D research works and at present nine Ph.D research scholars are working under him. The total number of technical publications in International/National journals/Conferences in around 700. He has chaired 7 International conferences and 30 National conferences. His research areas include modeling and simulation, neural networks, fuzzy systems and genetic algorithm, pattern recognition, multi dimensional system analysis, linear and nonlinear control systems.

Ramani Bai.V is a research scholar under the guidance of Dr. S. N. Sivanandam, Professor and Head, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore. She has 10 years of teaching experience and 2 years of Industrial experience. She is an Assistant Professor in Computer Science and Engineering at SAINTGITS College of Engineering, Kottayam, Kerala .