J. Appl. Environ. Biol. Sci., **4(8S)192-198, 2014**

© 2014, TextRoad Publication

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Hybrid Computing Approach for Solving Delay Differential Equations

Iftikhar Ahmad[1], Maqsoom Fatima[2] and Muhammad Bilal[3]

[1,2,3]Department of Mathematics, University of Gujrat,

Received: September 1, 2014
Accepted: November 13, 2014

ABSTRACT

A number of nonlinear phenomenas in many branches of the applied sciences and engineering are described in terms of delay differential equations, which arise when the evolution of a system depends both on its present and past time. In this paper, our aim is to present various optimization techniques for solving the delay differential equations with variable coefficients subject to initial conditions. We have used constrained nonlinear minimization in Active Set technique (AST) and Sequential Quadratic Programming (SQP) algorithms to solve delay differential equations. Also we have used Genetic Algorithm (GA) and a hybrid technique GA-SQP. Further, we presented the comparison of proposed numerical results with exact solution to confirm the reliability of our method for delay differential equations. We considered higher order delay differential equations and provided their numerical results on the bases of which we showed their complete graphical picture.

KEYWORDS: Delay differential equation; Active Set; Sequential Quadratic Programming; Genetic Algorithm.

1. INTRODUCTION

Delay Differential Equations (DDEs) are widely used in the mathematical formulation of real life phenomena in many fields especially in engineering and sciences such as control problems, population dynamics, secure communication, infectious disease, economics and traffic control. In a DDE, the system not only depends on a certain time but also depends on the state of the system at an earlier time in contrast with Ordinary Differential Equations (ODEs) where the unknown function and its derivatives are evaluated at the same time [1].

A typical first order single-delay scalar DDE model may be expressed as:

$$y'(t) = f_1\{t, y(t), y(t-\tau)\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

The term $y(t-\tau)$ is called the delay term. In more general form the function $y'(t)$ depends on $y(t)$ and $y'(t)$ itself at the past time $y(t-\tau)$ as defined above, in this case, the above equation can be written as:

$$y'(t) = f_2\{t, y(t), y(t-\tau), y'(t-\tau)\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

DDEs have attracted the attention of researchers in mathematical, biological and physical sciences. This is especially due to the fact that the theory of ODEs does not carry over to DDEs. Among the topics studied for the DDEs, oscillation of the solutions has been resolved the most and complied in the monographs [2]. The general theory of DDEs is developed by Hale [3], Bell-man and Cooke [4], El'sgol'ts and Norkin [5], Driver [6] and Kolmanovskii and Myshkis [7], Hale and Verduyn Lunel [8], Kolmanovskii and Nosov [9], Diekmann et al, [10] and Kuang [11, 12], which also include many real-life examples of DDEs and more general retarded functional differential equation [13].

For solving the initial value problems of DDEs with a constant delay $\tau > 0$ a lot of numerical methods have been presented in recent times. The numerical theory, such as stability and convergence issues, has also been developed.

For example, Linear Multistep Methods (LMMs), Runge-Kutta methods (RKMs) have been investigated and one-leg methods have been studied. These numerical methods for initial value problems of DDEs were obtained by using the corresponding methods to the initial problems of ODEs [14].

From many years, the numerical solution of models of dynamic systems has attracted the researchers. Many physical systems can be approximated by sets of ODEs, and the digital simulation of such models has caught the interest of engineers and applied mathematicians from the invention of the digital computer. There is also a huge number of systems from engineering and science that require the inclusion of delays in their models. The numerical simulation of models described by sets of DDEs has been developed by very few publications and the state of the art of software is not highly developed for dealing with such models.

Good solvers were developed by Shampine and co-workers for simulating DDE models. These include a numerical DDE solver, called dde23, encoded in Matlab. They also include a numerical DDE solver, called dde solver, encoded in Fortran. Both solvers are classical solvers in the sense that they are based on the classical time-slicing algorithms used throughout the numerical ODE, DDE and DAE (Differential Algebraic equation) [15].

Most codes available in solving DDE do not catter for stiff DDEs. Most of them used explicit RKMs to solve DDEs. The only effort so for on stiff DDE is done by Roth [16]. He solved stiff DDEs using three methods, which are the backward differentiation (BDF) method, the Adams method and the Rung-Kutta Fehlberg method. The systems are considered as stiff right from the beginning and Lagrange interpolation is used to approximate the delay term [17].

In this paper, we are going to present various optimization techniques for solving the delay differential equations with variable coefficients subject to initial conditions. We have used constrained nonlinear minimization in Active Set

Corresponding Author: Maqsoom Fatima, Department of Mathematics, University of Gujrat, Pakistan.
Email address: maqsoom.fatima@uog.edu.pk

technique (AST) and Sequential Quadratic Programming (SQP) algorithms to solve delay differential equations. Also we have used Genetic Algorithm (GA) and a hybrid technique GA-SQP. Mathematical modeling is defined in section 2. In section 3, solution technique is provided. Application of method is applied in section 4, where numerical results are provided. Section 5 provides discussion and conclusion.

2. Mathematical Modeling:

A neural network model is provided in detail with satisfying initial conditions for DDE. For the following DDE of order *n* we have,

$$y^{(n)}(t) = f(t, y(t), y(t-\tau), y'(t), y'(t-\tau), \ldots) \text{ .... (3)}$$

With these initial conditions

$$y(0) = y_0, y'(0) = y_1, \ldots, y^{(n-1)}(0) = y_{n-1} \text{ ....(4)}$$

Mathematical model of above DDE in the form of following continuous mapping for the solution $y(t)$, and its first derivative $\frac{dy}{dt}$, second $\frac{d^2y}{dt^2}$, and nth order derivative $\frac{d^n y}{dt^n}$, respectively, written as:

$$\hat{y}(t) = \sum_{i=1}^{n} a_i \phi(b_i t + c_i)$$

$$\frac{d\hat{y}(t)}{dt} = \sum_{i=1}^{n} a_i \frac{d}{dt} \phi(b_i t + c_i)$$

$$\frac{d^2\hat{y}(t)}{dt^2} = \sum_{i=1}^{n} a_i \frac{d^2}{dt^2} \phi(b_i t + c_i)$$

$$\frac{d^n\hat{y}(t)}{dt^n} = \sum_{i=1}^{n} a_i \frac{d^n}{dt^n} \phi(b_i t + c_i), \text{ where } i = 1,2,3,\ldots,n.$$

The model shown in the above equations are normally using log-sigmoid based on logarithmic function $\phi(t)$ and its respective derivatives, where

$$\phi(t) = 1 - \frac{e^{-t}}{1+e^{-t}} \text{ ....(5)}$$

Furthermore, the above mathematical models can also be written as:

$$\hat{y}(t) = \sum_{i=1}^{n} a_i \left(1 - \frac{e^{-(b_i t + c_i)}}{(1 + e^{-(b_i t + c_i)})} \right)$$

$$\frac{d\hat{y}(t)}{dt} = \sum_{i=1}^{n} a_i b_i \left(\frac{e^{-(b_i t + c_i)}}{(1 + e^{-(b_i t + c_i)})^2} \right)$$

$$\frac{d^2\hat{y}(t)}{dt^2} = \sum_{i=1}^{n} a_i b_i^2 \left(-\frac{e^{-(b_i t + c_i)}}{(1 + e^{-(b_i t + c_i)})^2} + 2\frac{e^{-2(b_i t + c_i)}}{(1 + e^{-(b_i t + c_i)})^3} \right)$$

$$\frac{d^n\hat{y}(t)}{dt^n} = \sum_{i=1}^{n} a_i b_i^n \left(-\frac{e^{-(n-1)(b_i t + c_i)}}{(1 + e^{-(b_i t + c_i)})^n} + n\frac{e^{-n(b_i t + c_i)}}{(1 + e^{-(b_i t + c_i)})^{n+1}} \right)$$

2.1 Fitness Function:

A fitness function or error function is given by sum of two errors as;

$$e = e_1 + e_2 \text{ ,..(6)}$$

where e_1 is error function associated with differential equation and it is given as:

$$e_1 = \frac{1}{N} \sum_{m=1}^{N} \left[\frac{d^n \hat{y}_m}{dt^n} - f(t_m, \tilde{y}_m, \hat{y}_m, \tilde{y}'_m, \hat{y}'_m, \ldots) \right]^2 \text{ ,............................(7)}$$

where $\hat{y}_m = \hat{y}(t_m), \tilde{t}_m = t_m - \tau, \tilde{y}_m = \tilde{y}(t_m), N = \frac{1}{h}, t_m = mh.$

Similarly, e_2 is the error function associated with initial conditions, which is defined as

$$e_2 = \frac{1}{n}\left[(\hat{y} - y_0)^2 + \ldots + (\hat{y}^{n-1} - y_{n-1})^2\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

3. Solution Technique

To find the Solution of the problem, we have applied the Active Set Technique, Sequential Quadratic Programming, Genetic Algorithm and hybrid approach GA-SQP by using the MATLAB built-in functions with the parameters setting given below for AST, SQP and GA, respectively.

3.1. *Procedural Steps of Proposed Methods.* The necessary procedure for AST, SQP and GA is given in following steps:

- ***Steps 1: Initialization***: Initial values of parameters are set in this step with random assignment. These settings are provided in Table I and Table II for important parameters.
- ***Step 2: Fitness Evaluation***: Calculate the fitness for each individual using the fitness function defined earlier.
- ***Step 3: Termination Criteria***: When the criteria is achieved the algorithm is terminated, and the criteria is achieved for the following conditions:
  - Maximum number of iterations is completed.
  - Defined fitness function is achieved.
  - Any defined value in optimset for maximum function evaluations, X tolerance or function tolerance is achieved as defined in above tables.
- ***Step 4: Save Results***: If the termination criteria is achieved then save the final optimal weights along with fitness values.
- ***Step 5: Statistical Analysis:*** Perform all these steps mentioned above on a large number of runs to get an effective and reliable statistical analysis. It is shown in Fig.1.

Table 1. Parameter settings for the functions "AST" and "SQP" in MATLAB for optimtool.

Parameters	Values	Parameters	Values
Start Point generation	Randomly between (0,1)	Function Tolerance	10^{-12}
Maximum Iteration	500	SQP Constraint Tolerance	Zero
Start Point	Randomly between (1,30)	Nonlinear Constraint Tolerance	Zero
Maximum Function Evaluations	1000000	Unboundedness Threshold	Default
Hessian	FBGS	Relative Line Search Bound	No Bound
X Tolerance	10^{-12}	Sub problem Algorithm	$\lvert d \rvert$ Factorization
Finite Difference Type	Forward Difference	Scaling	None
Start Point Size	30	Others	Default

Table II: Parameter settings for the function "Genetic Algorithm" in MATLAB simulations

Parameters	Values	Parameters	Values
Number of Variables	30	Direction	Forward
Population Type	Double Vector	Hybrid Function	None
Population Size	[30 30 30 30 30 30 30 30 30 30]	Generations	100
Creation Function	Constraint Dependent	Function Tolerance	10^{-12}
Scaling Function	Rank	Level of Display	Off
Selection Function	Uniform	Others	Default
Mutation Function	Constraint Dependent	Crossover Function	Heuristic

3.2. Procedural Steps for Hybrid Method GA-SQP:

In this optimizer the step 1-3 are same as mentioned above. For step 4, if the termination criteria is achieved then SQP is used for further refinement of results by taking final weights of GA as initial weights in start point of SQP algorithm. SQP is applied then by following the parameter settings defined in table I then save the final weights of the algorithm. The above defined procedural steps are defined in the following flow chart.

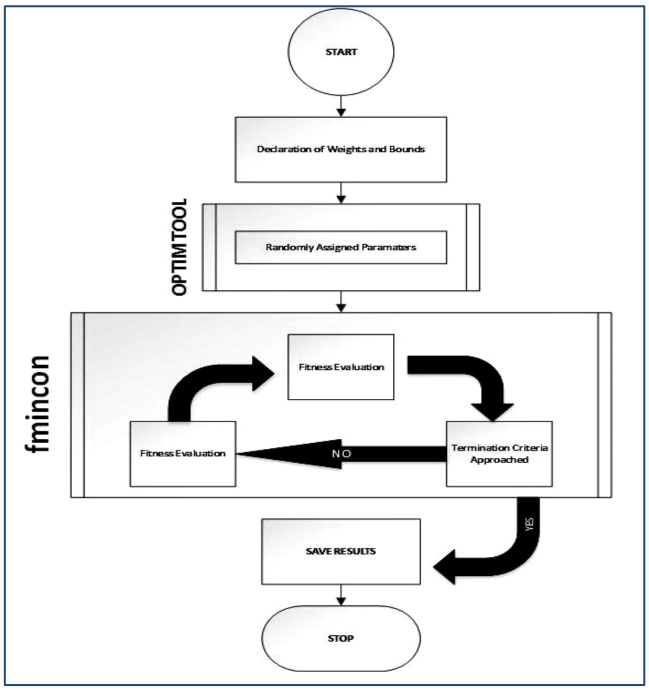
START
Declaration of Weights and Bounds
OPTIMTOOL
Randomly Assigned Paramaters
fmincon
Fitness Evaluation
Fitness Evaluation
NO
Termination Criteria Approached
YES
SAVE RESULTS
STOP

Fig. 1: Flow Chart of procedural steps

4. Application of Proposed Techniques:

Consider the following second order DDE,

$$y''(t) = y(t) - y(t-0.3) + \exp^{(-t+0.3)} \quad \text{......................} \ (9)$$

$$y(0) = 1, y'(0) = -1 \ \text{...}(10)$$

The exact solution of the above equation is

$$y(t) = e^{-t} \ \text{..}(11)$$

For the above DDE with given initial conditions, mathematical model in the form of continuous mapping for the solution $y(t)$, and its first derivative $\frac{dy}{dt}$, second $\frac{d^2y}{dt^2}$, and nth order derivative $\frac{d^n y}{dt^n}$, respectively, written as:

$\hat{y}(t) = \sum_{i=1}^{n} a_i \phi(b_i(t) + c_i)$ and log-sigmoid is using function: $\phi(t) = 1 - \frac{e^{-t}}{1+e^{-t}}$

Error function is given by sum of two square errors as: $e = e_1 + e_2$. We apply the neural network model with 10 neurons to solve this problem. There are total 30 unknown parameters or weights. The error function e for the input span from [0,1] with the step size 0.1 where e_1 is error function associated with differential equation and it is given as:

$$e_1 = \frac{1}{11}\sum_{m=0}^{10}\left[\frac{d^2\hat{y}_m}{dt^2} - \hat{y}_m + \tilde{y}_m - e^{0.3}.e^{-t}\right]^2 \ \text{....................................} \ (12)$$

where $\hat{y}_m = \hat{y}(t_m), \tilde{y}_m = \hat{y}(t_m - 0.3)$. Similarly, e_2 is the error function associated with initial conditions $t = 0, y_0 = 1, y'_0 = -1,$

$$e_2 = \frac{1}{2}\left[(\hat{y}_0 - 1)^2 + (\hat{y}_0' + 1)^2\right] \quad \text{..................................} \ (13)$$

Table III. Comparison of Exact Solution, AST, SQP, GA and GA-SQP for given problem

t	Exact Solution	AST	SQP	GA	GA-SQP
0.0	1.000000	1.000000	1.000000	0.999546	0.999998
0.1	0.904837	0.904835	0.904835	0.904811	0.904832
0.2	0.818731	0.818727	0.818727	0.818850	0.818722
0.3	0.740818	0.740812	0.740813	0.741051	0.740807
0.4	0.670320	0.670312	0.670313	0.670733	0.670305
0.5	0.606531	0.606521	0.606521	0.607218	0.606512
0.6	0.548812	0.548799	0.548800	0.549857	0.548789
0.7	0.496585	0.496571	0.496572	0.498055	0.496560
0.8	0.449329	0.449312	0.449314	0.451272	0.449300
0.9	0.406570	0.406551	0.406552	0.409022	0.406537
1.0	0.367879	0.367858	0.367860	0.370865	0.367843

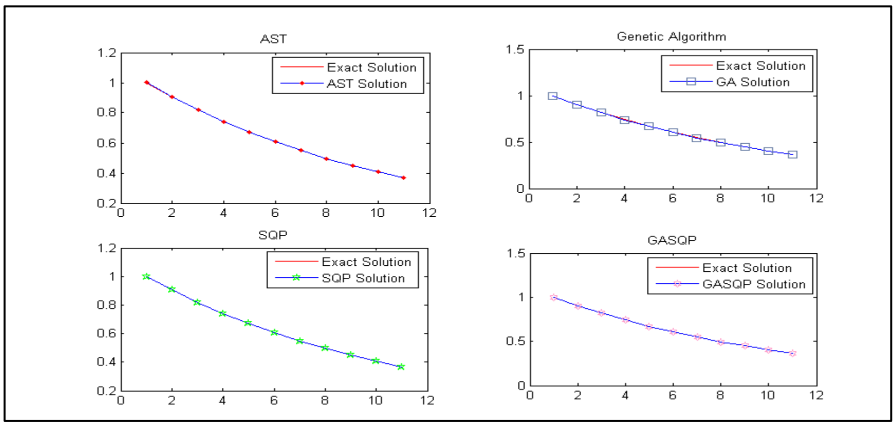

Fig. 2: Comparisons of proposed solutions with exact solution for given problem

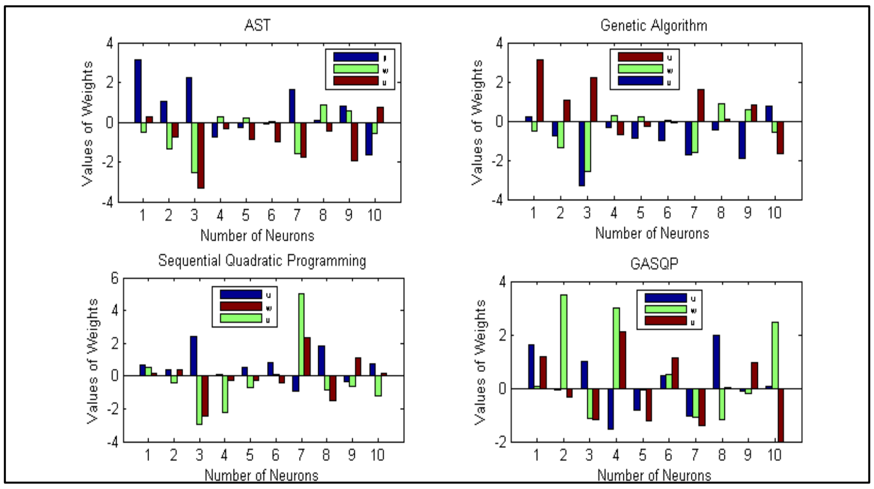

Fig. 3: Set of trained weights by AST, GA, SQP and GA-SQP for given problem

4.1. Numerical Experimentation and Results

The comparison of exact solution and solution obtained by proposed methods are given below in **Table III.** Furthermore, the **Fig. 2** also illustrates the worth of proposed solution by comparing the exact solution and numerical solution obtained by proposed methodology. Set of optimal weights calculated by AST, SQP, GA and GA-SQP are given below in equations. These are also graphically shown in **Fig. 3** and **Fig. 4** in two and three dimensions respectively.

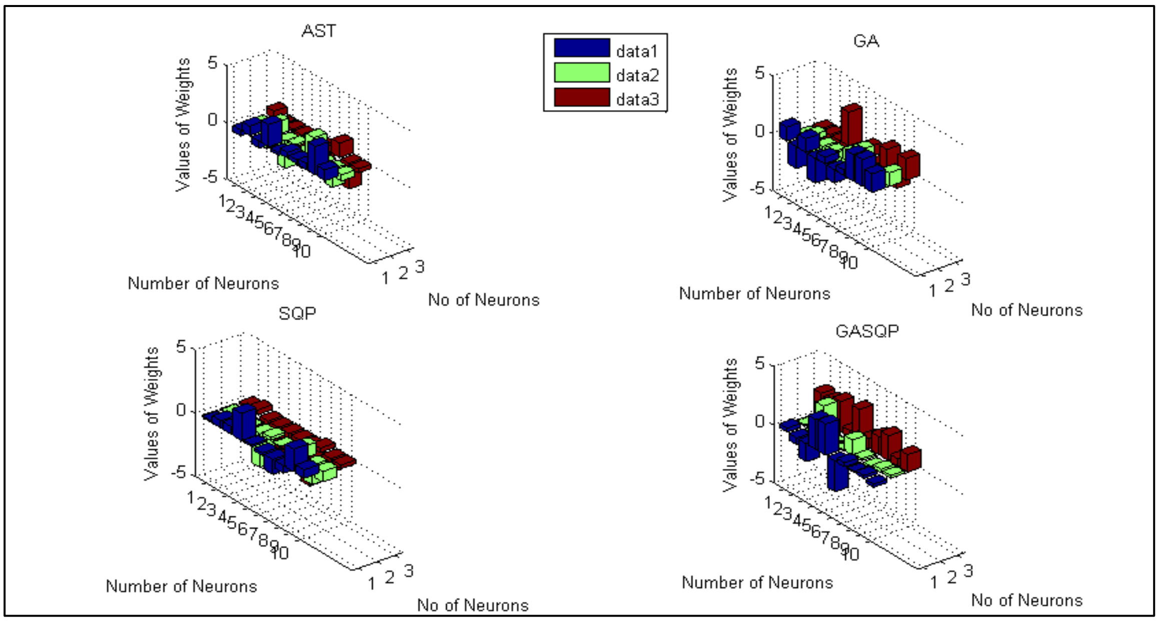

Fig. 4: Set of trained weights by AST, GA, SQP and GA-SQP for given problem

$$
\left(\begin{aligned}
\hat{y}_{AST} =&(-0.560375696662417)\left[1-\frac{e^{-(-0.775569822527061t+1.269023038504775)}}{1+e^{-(-0.775569822527061t+1.269023038504775)}}\right]+(0.017225415158451)\left[1-\frac{e^{-(0.703318272552184t+0.211832681359710)}}{1+e^{-(0.703318272552184t+0.211832681359710)}}\right]+\\
&(0.823737050454974)\left[1-\frac{e^{-(-0.553561263327982t+0.023117514595446)}}{1+e^{-(-0.553561263327982t+0.023117514595446)}}\right]+(1.236421722938689)\left[1-\frac{e^{-(0.142224276322291t+1.828613348392143)}}{1+e^{-(0.142224276322291t+1.828613348392143)}}\right]+\\
&(2.105839952632389)\left[1-\frac{e^{-(-2.656993331896340t+-3.282539425421912)}}{1+e^{-(-2.656993331896340t+-3.282539425421912)}}\right]+(0.239366505543750)\left[1-\frac{e^{-(-0.881875216429771t+0.394764621786012)}}{1+e^{-(-0.881875216429771t+0.394764621786012)}}\right]+\\
&(-0.456102093368282)\left[1-\frac{e^{-(-0.047945157953106t+1.238334555241631)}}{1+e^{-(-0.047945157953106t+1.238334555241631)}}\right]+(-0.468750297682763)\left[1-\frac{e^{-(0.754525208237513t+-0.247354145547301)}}{1+e^{-(0.754525208237513t+-0.247354145547301)}}\right]+\\
&(-0.542542862100195)\left[1-\frac{e^{-(0.201469409148985t+0.010980638546419)}}{1+e^{-(0.201469409148985t+0.010980638546419)}}\right]+(2.533491328484165)\left[1-\frac{e^{-(-1.442004844993111t+-1.262639711813955)}}{1+e^{-(-1.442004844993111t+-1.262639711813955)}}\right]
\end{aligned}\right) \tag{14}
$$

$$
\left(\begin{aligned}
\hat{y}_{SQP} =&(0.662760246297838)\left[1-\frac{e^{-(0.181886155427568t + 0.541435695633353)}}{1+e^{-(0.181886155427568t + 0.541435695633353)}}\right]+(0.353657817230244)\left[1-\frac{e^{-(0.396126942753631t+(-0.442193185355769))}}{1+e^{-(0.396126942753631t+(-0.442193185355769))}}\right]+\\
&(2.440751375340688)\left[1-\frac{e^{-(-2.472628039115358t+ (-2.940959702398721)}}{1+e^{-(-2.472628039115358t+ (-2.940959702398721)}}\right]+(0.087077350776645)\left[1-\frac{e^{-(-0.293096644415935t+(-2.212308189304661)}}{1+e^{-(-0.293096644415935t+(-2.212308189304661)}}\right]+\\
&(0.510313883570241)\left[1-\frac{e^{-(-0.293096644415935t+(-0.700801517458886)}}{1+e^{-(-0.293096644415935t+(-0.700801517458886))}}\right]+(0.836682517866336)\left[1-\frac{e^{-(-0.443322520366752t+0.111149243252744)}}{1+e^{-(-0.443322520366752t+0.111149243252744)}}\right]+\\
&(-0.905631609343865)\left[1-\frac{e^{-(2.367304733038706t + 5.001717921490819)}}{1+e^{-(2.367304733038706t + 5.001717921490819)}}\right]+(1.863465031255667)\left[1-\frac{e^{-(-1.481677073748799t+(-0.883213916691046)}}{1+e^{-(-1.481677073748799t+(-0.883213916691046)}}\right]+\\
&(-0.349757700806720)\left[1-\frac{e^{-(1.097635200399730t+(-0.613553648201361)}}{1+e^{-(1.097635200399730t+(-0.613553648201361)}}\right]+(0.760716282610728)\left[1-\frac{e^{-(0.150697362451676t+(-1.184510703857781)}}{1+e^{-(0.150697362451676t+(-1.184510703857781)}}\right]
\end{aligned}\right) \tag{15}
$$

$$
\left(\begin{aligned}
\hat{y}_{GA} =&(1.969196293207644)\left[1-\frac{e^{-(0.332126936555156t+ 1.007681983199347)}}{1+e^{-(0.332126936555156t+ 1.007681983199347)}}\right]+(2.111724406046478)\left[1-\frac{e^{-(0.444172085431754t+(-0.279450893378288))}}{1+e^{-(0.444172085431754t-0.279450893378288)}}\right]+\\
&(-1.001787992262400)\left[1-\frac{e^{-(3.332947377275055t+1.447706027623595)}}{1+e^{-(3.332947377275055t+1.447706027623595)}}\right]+ (-0.470792649996844)\left[1-\frac{e^{-(2.185712759283145t-0.301501202176470)}}{1+e^{-(2.185712759283145t-0.301501202176470)}}\right]+\\
&(-0.697866455418294)\left[1-\frac{e^{-(0.873077540774102t+2.178047952176411)}}{1+e^{-(0.873077540774102t+2.178047952176411)}}\right]+(-0.743154415343845)\left[1-\frac{e^{-(0.756907555473917t+ 1.529726330379187)}}{1+e^{-(0.756907555473917t+1.529726330379187)}}\right]+\\
&(1.526309603483474)\left[1-\frac{e^{-(-0.753589156996979t+(-0.027084312024924))}}{1+e^{-(-0.753589156996979t-0.027084312024924)}}\right]+(-1.105327764372499)\left[1-\frac{e^{-(0.192131165731807t+1.441594044637437)}}{1+e^{-(0.192131165731807t+1.441594044637437)}}\right]+\\
&(0.664396024451153)\left[1-\frac{e^{-(-1.420074094303599t+1.358487483132616)}}{1+e^{-(-1.420074094303599t+1.358487483132616)}}\right]+ (0.770643222832880)\left[1-\frac{e^{-(0.124593090557029t+0.673115873539305)}}{1+e^{-(0.124593090557029t+0.673115873539305)}}\right]
\end{aligned}\right) \tag{16}
$$

$$
\left(\begin{aligned}
\hat{y}_{GA-SQP} =&(-1.191431060533378)\left[1-\frac{e^{-(0.278911922019130t+1.365935718401084)}}{1+e^{-(0.278911922019130t+1.365935718401084)}}\right]+(0.529614018951745)\left[1-\frac{e^{-(-0.990318507315346t+0.225897485181500)}}{1+e^{-(-0.990318507315346t+0.225897485181500)}}\right]+\\
&(-0.780839802931623)\left[1-\frac{e^{-(1.913383112122880t+2.056665453809710)}}{1+e^{-(1.913383112122880t+2.056665453809710)}}\right]+(2.072564254239857)\left[1-\frac{e^{-(-1.476796702442784t+(-0.983846604535705))}}{1+e^{-(-1.476796702442784t-0.983846604535705)}}\right]+\\
&(-0.068925349662073)\left[1-\frac{e^{-(1.298764347101732t+(-1.801665027545679))}}{1+e^{-(1.298764347101732t+(-1.801665027545679))}}\right]+(-1.231270368367227)\left[1-\frac{e^{-(3.029564055050950t+3.030238468989015)}}{1+e^{-(3.029564055050950t+3.030238468989015)}}\right]+\\
&(2.213368127024384)\left[1-\frac{e^{-(0.100149361415171t+2.080651212645160)}}{1+e^{-(0.100149361415171t+2.080651212645160)}}\right]+(-0.138283640684099)\left[1-\frac{e^{-(-0.619958197695562t+1.310252620309056)}}{1+e^{-(-0.619958197695562t+1.310252620309056)}}\right]+\\
&(0.541804557406408)\left[1-\frac{e^{-(1.245016206814900t+2.173530438199177)}}{1+e^{-(1.245016206814900t+2.173530438199177)}}\right]+(0.651293461630325)\left[1-\frac{e^{-(0.589440455081705t+3.055471334527169)}}{1+e^{-(0.589440455081705t+3.055471334527169)}}\right]
\end{aligned}\right) \tag{17}
$$

5. Conclusion

- ➢ A new approach is developed to solve Delay Differential Equations using Active Set, Genetic Algorithm, Sequential quadratic Programming and their hybrid approach GA-SQP.
- ➢ Comparison of exact solution with the reported solution obtained by the above mentioned techniques is provided for given problem.
- ➢ It is concluded that AST and SQP are more efficient than GA.
- ➢ And GA-SQP provides more quick and better results for optimization and time factor is minimize in this technique.
- ➢ Thus a new artificial intelligence based technique is developed for solving higher order delay differential equations.

In future someone can try these techniques for future delay in differential equations with tan-sigmoid and some application of Bessel's function. One can improve the accuracy and convergence of results by changing the optimization algorithm.

REFERENCES

1 Mehrkanoon, S., Mehrkanoon, S., & Suykens, J.A. K, Parameter estimation of delay differential equations: An integration-free LS-SVM approach, Commun Nonlinear Sci Numer Simulat, 2013.

2 Ravi P. Agarwal, & Fatma K, A survey on oscillation of impulsive delay differential equations, Computers and Mathematics with Applications, 2010. 60 1648-1685.

3 Hale, J. K, Theory of Functional Differential Equations, Springer-Verlag, New York, 1977.

4 Cooke, R., & Bellmanand, K. L, Differential Difference Equations, Academic Press, New York-London, 1963.

5 El’sgol’ts, L. E., &Norkin, S. B, Introduction to the Theory and Applications of Differential Equations with Deviating Arguments, Academic Press, New York, 1973.

6 Driver, R. D, Ordinary and Delay Differential Equations, Springer-Verlag, Berlin, 1977.

7 Kolmanovskii, V., Nosov, V, Stability of Functional Differential Equations, Academic Press, London, 1986.

8 Hale, J. K., Verduyn Lunel, S. M, Introduction to Functional Differential Equations, Applied Mathematical Sciences, 99, Springer-Verlag, New York, 1977.

9 Kolmanovskii, V., & Myshkis, V, Applied Theory of Functional Differential Equations. Kluwer, Dordrecht, 1992.

10 Diekmann, O., Van Gils, S. A., Verduyn Lunel, S. M., & Walter, H. D, Delay Equations: Functional, Complex and Nonlinear Analysis, AMS series 110, Springer-Verlag, Berlin, 1995.

11 Kuang, Y, Delay Differential Equations with Applications in Population Dynamics, Academic Press, Boston, 1993.

12 Kuang, Y, On neutral Delay Logistic Gause-Type Predator-Prey Systems, Dynamics, Academic Press, Boston, 1993.

13 Moghaddam, Behrouz, Zeynab, P., & Mostaghim, S. A, Numerical method based on finite difference for solving fractional delay differential equations, Journal of Taibah University for Science, 2013.

14 Zhang, Chengjian, & Chen, H, Block boundary value methods for delay differential equations, Numerical Mathematics, 2010. 60 915-923.

15 Castro, Rodrigo, Kofman, Ernesto, Cellier, & Franois, E., Quantization-based integration methods for delay-differential equations, Simulation Modelling Practice and Theory, 2011. 19 314-336.

16 Suleiman, Mohamed bin Ismail, & Fudziah, Solving delay differential equations using component wise partition by Rung-Kutta method, Applied Mathematics and Computation, 2001. 122 301-323.

17 Tohidi, E., Bhrawy, A. H., & Erfani, K, A collocation method based on Bernoulli operational matrix for numerical solution of generalized pantograph equation, Applied Mathematical Modelling , 2013. 37 4283-4294.