Appendix A

The Various Object Oriented Classes
The concepts discussed so far have been developed as a MS window's application. It provides a generalized interface to solve Optimal Control Problems, from the simplest unconstrained problems to the more complicated multivariable problems involving nonlinear plant equation, non quadratic cost function and problems with physical and boundary constraints. It implements object oriented concepts making use of Microsoft Foundation Classes (MFC) [107-111]. Generic classes for models DE/DENN, AGA/AGANN form the core of the application implementing the algorithms proposed in the earlier chapters.

**Features of the MS-Windows Application**

1) The interface has been ergonomically designed to accept inputs for any type of Optimal Control Problems.

2) It allows both maximization and minimization of the performance index.

3) Separate data entry screens for inputs specific to a model have also been provided.

4) Saving and retrieval of the entered problems and their corresponding results have been facilitated.

5) The results saved can be easily exported to EXCEL and MATLAB for analysis.

6) Auto Scale graph for run time plotting of fitness values Vs iterations has been provided.

7) The graphs for the optimized values of the control inputs and state variables can be chosen and displayed at any time during the iteration process.

8) A tabulated result is provided at end of simulation. Selected values of the results can be copied to clipboard.
The application is created with the help of MFC Wizard project using Multiple Document Interface (MDI) in VC++. The object relationship flow diagram is shown in Fig. A.1. The important classes, which drive the application, are

1) CMainFrame - The Mainframe and the application tool bar
2) CChildFrame - Frame for Child windows
3) C1stGridView - Applications view class. Implements the grid control view used for displaying tabulated results.
4) COScopeCtrl - Implements the graph displays.
5) CTestoscope - It has all the models as member variables along with Oscope control for displaying graphs. It is responsible for the initialization of the corresponding model chosen by the user, and running the model for specified number of iteration, updating the fitness value and iteration display.
6) CProjectDoc - Document class which handles loading, storing, initializing of parameters for a problem.
7) CExpression - Expression evaluator class which parses the input equations and returns the evaluated result.
8) CValue - Class for storing values of variables in the equations.
9) Cmap Variable - Class contains a key (variable name ) and CValue to store variable value.
10) CProjectApp - Main Application Class.
11) CProgressST - Owner drawn Progress bar control.
12) Backprob - Class to implement backpropagation Neural Network.
13) DE - Class to implement Differential Evolution model.
14) DENN - Class to implement Differential Evolution with Neural Networks model.

15) AGA - Class to implement Adaptive Genetic Algorithm model.

16) AGANN - Class to implement Adaptive Genetic Algorithm with Neural Network model.

17) CExpressionDlg - Data Entry dialog box for input of Objective function and constraints.

18) CStateInsertDlg - Data Entry dialog box for input of state variables.

19) CVarInsertDlg - Data Entry dialog box for input of control variables.

20) CNNInputDlg - Data Entry dialog box for input of Neural Network parameters.

Fig. A.1 Objects Relationship Diagram