Using DiagramEng

An Introduction to Modelling and Simulation with DiagramEng

INTRODUCTION

The DiagramEng software application provides users the means to efficiently represent mathematical equations, perform interactive and intuitive model building and conduct control engineering experiments. The software incorporates block icons, representing model components, and adjoining connections, signifying a relationship between them.

DiagramEng has been built with Microsoft Visual C++ 6.0 [1] using the Microsoft Foundation Classes (MFC) and the Standard Template Library (STL). The graphical user interface involves: menus, toolbars, a block library tree-like browser for block selection, and a palette upon which a diagram, consisting of blocks and connections, can be drawn. The computation of a system model is performed either directly, or iteratively through the use of a Newton-method-based nonlinear solver. A range of blocks, including the Derivative and Integrator blocks, used for numerical differentiation and integration, respectively, allow the user to model time-based linear and nonlinear differential equations.

The remainder of the Using DiagramEng document discusses functionality provided in the menus, and toolbars, and provides examples of how to model typical real-world engineering problems and compute results that may be visualized or saved to an output data file.

1. MENUS

There are two frame-based sets of menus used in the DiagramEng application: 1) the Main frame-based menus, which include the File, View and Help menus (Fig. 1), and 2) the Child frame-based menus (Fig. 2) which include the File, Edit, View, Model, Simulation, Format, Tools, Window and Help menus. A listing of menu-based functionality is presented in the tables below.

1.1 Main Frame Menus

The following tables present information on the File, View and Help Main frame-based menus as shown in Fig. 1.

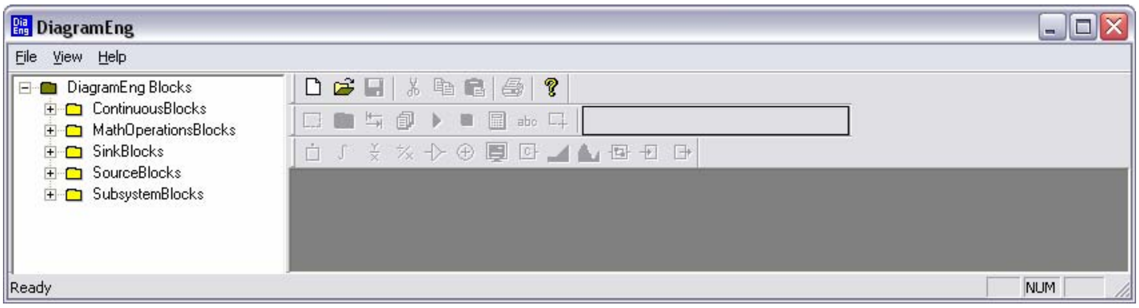

Figure 1 Main frame-based menus: File, View and Help.

Table 1 File menu entries and function.

File Menu Entry	Function
New	Create a new empty child document.
Open	Open an existing document previously saved to a model data file.
Print Setup	Set up the printer properties.
Recent Files	List the four most recent files.
Exit	Exit the application prompting the user to save unsaved files.

Table 2 View menu entries and function.

View Menu Entry	Function
Toolbar	Show or hide the Toolbar.
Status Bar	Show or hide the Status Bar.

Table 3 Help menu entry and function.

Help Menu Entry	Function
About DiagramEng	Display program, version number and copyright information.

1.2 Child Frame Menus

The following tables present information on the File, Edit, View, Model, Simulation, Format, Tools, Window and Help, Child frame-based menus, as shown in Fig. 2 below.

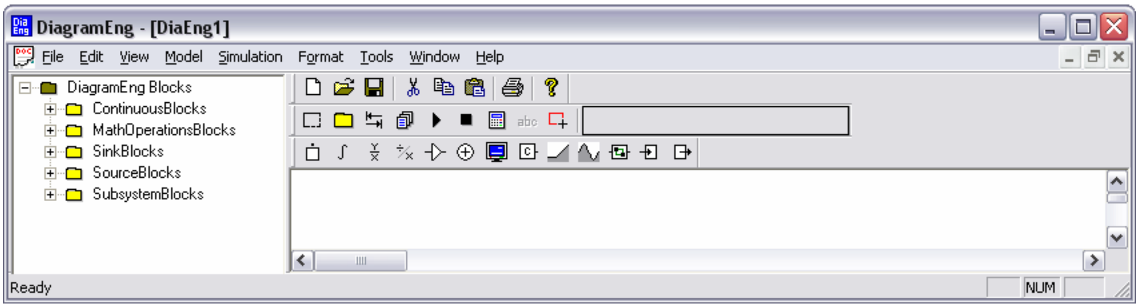

Figure 2 Child frame-based menus: File, Edit, View, Model, Simulation, Format, Tools, Window and Help.

Table 4 File menu entries and function.

File Menu Entry	Function
New	Create a new document in a new child document window.
Open	Open and existing document prompting the user to save the current document if it already exists in the child document window.
Close	Close a document prompting the user to save if the document content has not already been saved.
Save	Save the active document to a file.
Save As	Save the active document to a new file.
Print	Print the active document.
Print Preview	Preview the active document prior to printing.
Print Setup	Set up the printer properties.
Recent File	List the four most recent files.
Exit	Exit the application prompting the user to save unsaved files.

Table 5 Edit menu entries and function.

Edit Menu Entry	Function
Undo	Undo the last system model based editing action.
Redo	Redo the last system model based editing action.
Cut	Cut the selection and place it on the Clipboard.
Copy	Copy the selection and place it on the Clipboard.
Paste	Insert Clipboard contents onto the system model diagram.
Delete Grouped Items	Delete items grouped by an enclosing rectangular region.
Select All	Select all document content with an enclosing rectangular region.
Add Multiple Blocks	Present a block library dialog window for multiple block selection.

Table 6 View menu entries and function.

View Menu Entry	Function
Toolbar	Show or hide the toolbar.
Status Bar	Show or hide the status bar.
Common Ops. Toolbar	Show or hide the Common Operations toolbar.
Common Blocks Toolbar	Show or hide the Common Blocks toolbar.
Block Directory	Show or hide the block directory tree.
Auto Fit Diagram	Automatically fit diagram to view.
Zoom In	Zoom into detail enlarging the size of the diagram.
Zoom Out	Zoom out of detail reducing the size of the diagram.
Reset Diagram	Reset diagram to original size prior to zooming operations.

Table 7 Model menu entries and function: the shaded entry denotes a non-functional item.

Model Menu Entry	Function
Build Model	Build the active model
Build Subsystem	Build the selected model subsystem (not functional).

Table 8 Simulation menu entries and function.

Simulation Menu Entry	Function
Start	Start the simulation, invoking build model if model not already built.
Stop	Stop the simulation.
Numerical Solver	Set the numerical solver parameters (not all fields are functional).

Table 9 Format menu entry and function.

Format Menu Entry	Function
Show Annotations	Show or hide diagram annotations if present.

Table 10 Tools menu entry and function.

Tools Menu Entry	Function
Diagnostic Info.	Present process and system memory utilization statistics.

Table 11 Window menu entries and function.

Window Menu Entry	Function
New Window	Open another window for the active document.
Cascade	Arrange windows so they overlap.
Tile	Arrange windows as non-overlapping tiles.
Arrange Icons	Arrange icons at the bottom of the window
Close All Documents	Close all documents and prompt the user to save if necessary.
Name of child windows	Show names of windows and activate the selected window.

Table 12 Help menu entries and function.

Help Menu Entry	Function
About DiagramEng	Display program, version number and copyright information.
Using DiagramEng	Display information about using the DiagramEng application.

1.3 Context Menu

The Context menu is invoked by right-clicking on a diagram entity, e.g. a block or block port, or upon the palette, and an item may be chosen from the list to perform some action. The location of the cursor at which the Context menu is invoked is used to determine the applicability of the selected action for the object concerned. Some of the invoked functions need to be preceded or followed by other necessary steps to complete the whole interactive action. For example, to perform fine movement of a diagram entity, the Fine Move Item entry should be first selected, followed by the usage of the arrow keys to move the item. The Context menu entries and their function are listed in Table 13 below.

Table 13 Context menu entries and function.

Context Menu Entry	Function
Delete Item	Delete selected block, connection or connection bend point.
Delete Grouped Items	Delete items grouped by an enclosing rectangular region.
Fine Move Item	Move an item using the arrows keys.
Format Annotation	Format an existing annotation or insert a new one.
Insert Bend Point	Insert a bend point upon a connection object.
Reverse Block	Reverse the direction of a block.
Set Output Signal	Set block output connection-based signal.
Set Properties	Set block, port and numerical solver properties.

2. TOOLBARS

The three application toolbars are: 1) the standard Main frame-based toolbar and the Child frame-based toolbars, 2) Common Operations and 3) Common Blocks, as may be seen in Fig. 2 above. The following tables present information about the toolbar buttons and their associated functionality.

2.1 Main Frame-Based Toolbar

The standard Main frame-based toolbar is that shown in Fig. 3 below: as a child window is not open, the Child frame-based toolbars are not visible and only the Main frame-based functionality for the toolbar is enabled.

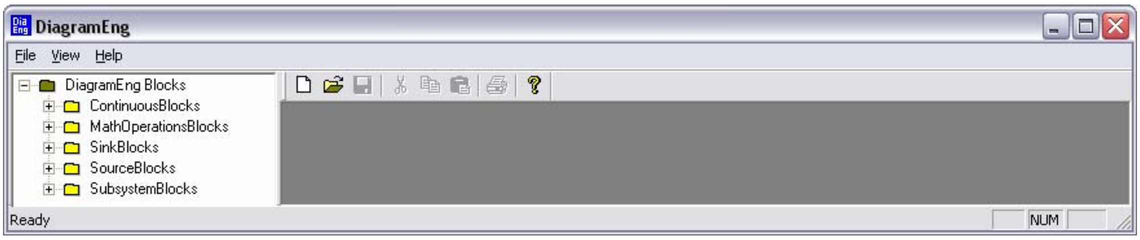

Figure 3 Main frame-based toolbar with the enabled buttons (no child document is present).

Table 14 Standard Main frame-based Toolbar: the shaded entries are inactive since they relate to the Child frame-based document (in the order displayed on the toolbar from left to right).

Toolbar Button	Function
New	Create a new, empty child document.
Open	Open an existing document previously saved to a model data file.
Save	Save the active document to a file.
Cut	Cut the selection and place it on the Clipboard.
Copy	Copy the selection and place it on the Clipboard.
Paste	Insert Clipboard contents onto the system model diagram.
Print	Print the active document.
About	Display program, version number and copyright information.

2.2 Child Frame-Based Toolbars

The Child frame-based toolbars, visible in Fig. 4 below, are the Common Operations and Common Blocks toolbars, second and third from the top respectively, where the Main frame-based toolbar is still present since some of its functionality becomes active in the presence of a child document, in particular, the Save, Cut, Copy, Paste and Print items (shown disabled in Fig. 3). The Common Operations and Common Blocks toolbar-based buttons and their function are shown in tables 15 and 16 respectively.

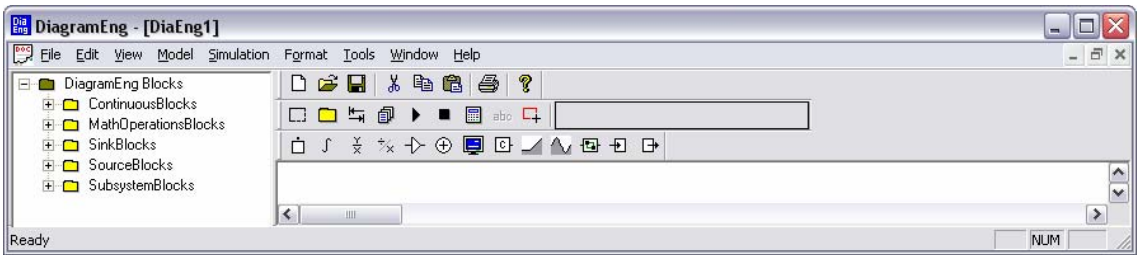

Figure 4 Child frame-based toolbars: Common Operations toolbar (second from the top) and the Common Blocks toolbar (third from the top), where the top Main frame toolbar is still shown.

Table 15 Common Operations toolbar buttons and their associated functionality (in the order displayed on the toolbar from left to right).

Toolbar Button	Function
Select All	Select all document content with an enclosing rectangular region.
Add Multiple Blocks	Present a block library dialog window for multiple block selection.
Auto Fit Diagram	Automatically fit diagram to view.
Build Model	Build the active model
Start Simulation	Start the simulation, invoking build model if not already built.
Stop Simulation	Stop the simulation.
Numerical Solver	Set the numerical solver parameters (not all fields are functional).
Show Annotations	Show or hide diagram annotations if present.
Track Multiple Items	Select and move multiple items.
Edit Box Control	Display the current simulation time and final execution time.

Table 16 Common Blocks toolbar buttons and their associated functionality (in the order displayed on the toolbar from left to right): the shaded entries are non-functioning blocks.

Toolbar Button	Function
Derivative Block	Add a Derivative block to the system model.
Integrator Block	Add an Integrator block to the system model.
Transfer Function Block	Add a Transfer Function block to the system model.
Divide Block	Add a Divide block to the system model.
Gain Block	Add a Gain block to the system model.
Sum Block	Add a Sum block to the system model.
Output Block	Add an Output block to the system model.
Constant Block	Add a Constant block to the system model.
Linear Function Block	Add a Linear Function block to the system model.
Signal Generator Block	Add a Signal Generator block to the system model.
Subsystem Block	Add a Subsystem block to the system model.
Subsystem In Block	Add a Subsystem In block to the system model.
Subsystem Out Block	Add a Subsystem Out block to the system model.

The blocks available on the Common Blocks toolbar are shown placed on the palette in Fig. 5 below. The user may double-click the centre of a block and enter block-specific parameters through the use of a block-parameter-input dialog window.

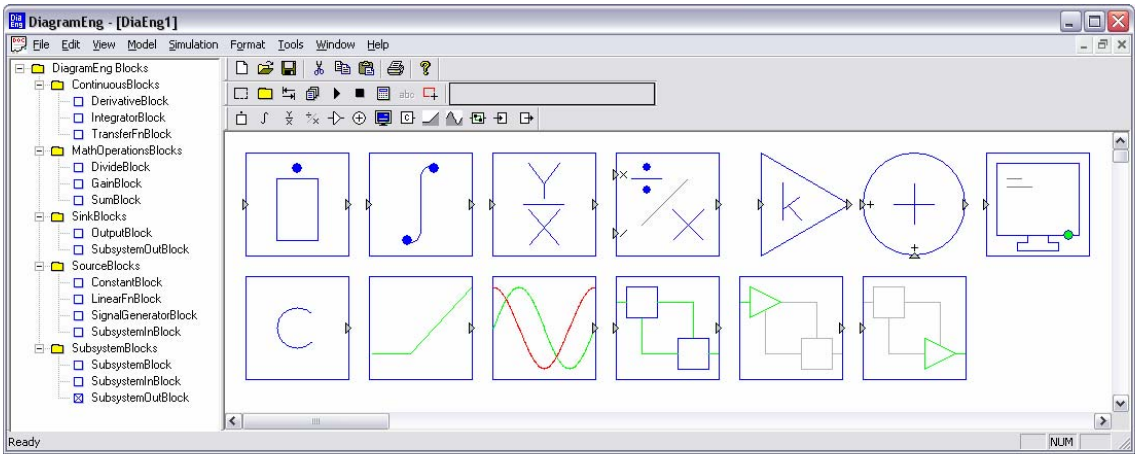

Figure 5 All blocks of the Common Blocks toolbar displayed on the palette.

3. EXAMPLES

Real-world engineering problems typically involve second order linear differential equations that need to be converted to first order equations using order reduction, prior to their numerical integration, to obtain the trajectories of the dependent variables and their time derivatives. In addition, nonlinear dynamical systems often possess coupling and oscillatory dynamics that can be conveniently modelled using feedback loops and computed with the Netwon-method-based nonlinear solver and the Integrator block. The following examples show the user how to model differential equations and nonlinear dynamical systems.

3.1 Second Order Linear Ordinary Differential Equations

Consider a simple second order linear differential equation representing a mechanical mass-spring-damper system (Example 3-3 p. 73 [2])

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t) \tag{1}$$

where m, b, k, $y(t)$ and $u(t)$, are the mass, damping constant, spring constant, output mass displacement from the equilibrium position, and external force input to the system, respectively. An order reduction is used to reduce the second order system to two first order equations, where $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$, and results in the following system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix} u(t) \tag{2a}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{2b}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}, \; \boldsymbol{B} = \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix}, \; \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \text{ and } D = 0$$

[2].

Students of control engineering will recall that the state and output equations in linear form are

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \tag{3a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t) \tag{3b}$$

where $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are the state, control and output vectors, respectively, and $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, $\boldsymbol{C}(t)$ and $\boldsymbol{D}(t)$ are the state, control, output, and direct transmission matrices, respectively [2]. One will notice on comparing equations (2) and (3), that (2a) and (2b) are the state and output equations respectively. The corresponding block diagram representation of the state and output equations (3) is shown in Fig. 6, below.

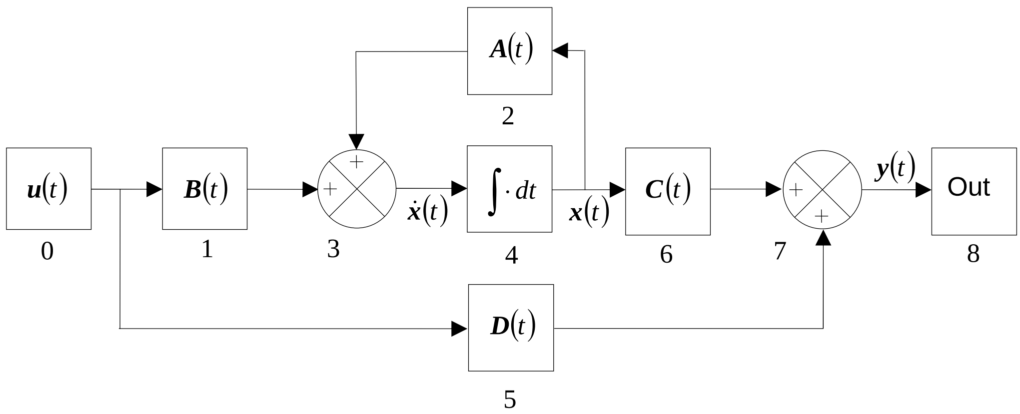

Figure 6 Block diagram representation of the state and output equations (3).

The integration in the above diagram concerns that of $\dot{x}(t)$ to yield $x(t)$, and since $\boldsymbol{x}(t) = [y(t), \dot{y}(t)]^{\mathrm{T}}$, then the initial condition for the Integrator block is $\boldsymbol{x}(0) = [y(0), \dot{y}(0)]^{\mathrm{T}}$ (e.g. if the mass is initially at rest with displacement $2.0\,(\mathrm{m})$, then $\boldsymbol{x}(0) = [2, 0]^{\mathrm{T}}$).

The engineer can draw Fig. 6 using the DiagramEng application, as shown in Fig. 7 below, and enter various selections of mechanical properties, m, b and k and forcing functions $u(t)$, to generate different displacement outputs $y(t)$, to analyse the physical response behaviour of the mass-spring-damper system.

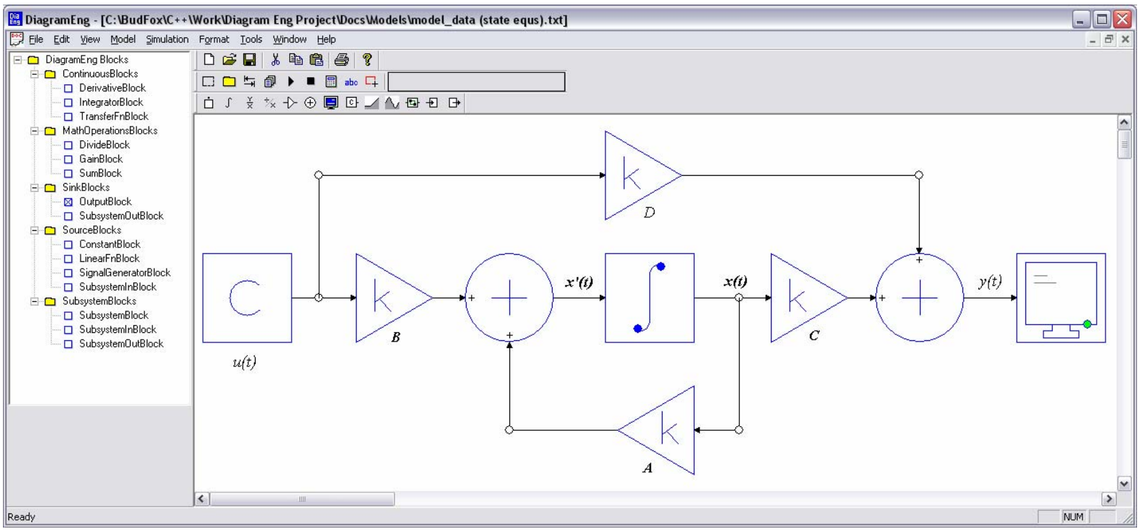

Figure 7 Block diagram model of the state and output equations (3) drawn with DiagramEng.

The general analytic solution $y_g(t) \equiv y(t)$ to (1) is the sum of the homogeneous solution $y_h(t)$ and the particular solution $y_p(t)$, i.e.

$$y_g(t) = y_h(t) + y_p(t). \tag{4}$$

The homogeneous solution is obtained by setting the right hand side of (1) to zero, as follows

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = 0 \tag{5a}$$

$$\Rightarrow \quad \ddot{y}(t) + \frac{b}{m}\dot{y}(t) + \frac{k}{m}y(t) = 0 \tag{5b}$$

and letting $y(t) = e^{rt}$, results in the characteristic equation

$$r^2 + \frac{b}{m}r + \frac{k}{m} = 0 \tag{6a}$$

with roots

$$r_{1,2} = \frac{1}{2m}\left(-b \pm \sqrt{b^2 - 4km}\right). \tag{6b}$$

Underdamped Vibration $\left(b^2 - 4km < 0\right)$

If the discriminant, $b^2 - 4km < 0$, then the motion of the mass-spring-damper system is said to be underdamped and the roots are complex, $r_{1,2} = \alpha \pm j\beta$, where $\alpha = -b/2m$, $\beta = (1/2m)\sqrt{4km - b^2}$, and $j = \sqrt{-1}$, and the homogeneous solution is

$$y_h(t) = C_1 e^{\alpha t}\left(\sin(\omega t) + C_2\right) \tag{7}$$

where $\omega = \beta$, is the angular frequency, and C_i, for $i = 1, 2$, are constants [3]. Values for the constants may be determined with knowledge of quantities $b, k, m, u(t)$ and the initial conditions $\boldsymbol{x}(0)$. The mathematician will notice here that since $\alpha < 0$, $y_h(t) \to 0$ as $t \to \infty$, with a decaying oscillatory motion.

Critically Damped Vibration $\left(b^2 - 4km = 0\right)$

If the discriminant, $b^2 - 4km = 0$, the motion of the system is said to be critically damped [3] and there exists a repeated root $r = -b/2m$, and the homogeneous solution is

$$y_h(t) = C_1 e^{-bt/2m} + C_2 t e^{-bt/2m} \, . \tag{8}$$

Both exponents are negative and hence $y_h(t) \to 0$ as $t \to \infty$, without oscillation.

<u>Overdamped Vibration</u> $(b^2 - 4km > 0)$

If the discriminant, $b^2 - 4km > 0$, the motion of the system is said to be overdamped [3] and there exists two real roots as given by (6b) and the homogeneous solution

$$y_h(t) = C_1 e^{r_1 t} + C_2 e^{r_2 t} \, . \tag{9}$$

Both roots are real but negative and hence $y_h(t) \to 0$ as $t \to \infty$, without oscillation.

The particular solution of (1) may be found by the Method of Undetermined Coefficients, and given initial conditions, the coefficients may be determined and a general solution found.

The following figures illustrate the three different damping conditions where the homogeneous solutions $(y_h(t))$ are those provided by (7-9) and $y_p(t) = 1/k$ (given $u(t) = 1$) for the intial conditions $\mathbf{x}(0) = [2, 0]^T$: 1) Fig. 8 (a) below shows underdamped oscillatory vibration, where $y_g(t) \to 1.0$ as $t \to \infty$, for $u(t) = 1$, $m = 1 = k$ and $b = 0.5$, 2) Fig. 8 (b), shows critically damped nonoscillatory vibration, where $y_g(t) \to 1.0$ as $t \to \infty$, for $u(t) = 1$, $m = 1 = k$ and $b = 2$, and 3) Fig. 8 (c), shows overdamped nonoscillatory vibration, where $y_g(t) \to 1.0$ as $t \to \infty$, for $u(t) = 1$, $m = 1 = k$ and $b = 3$.

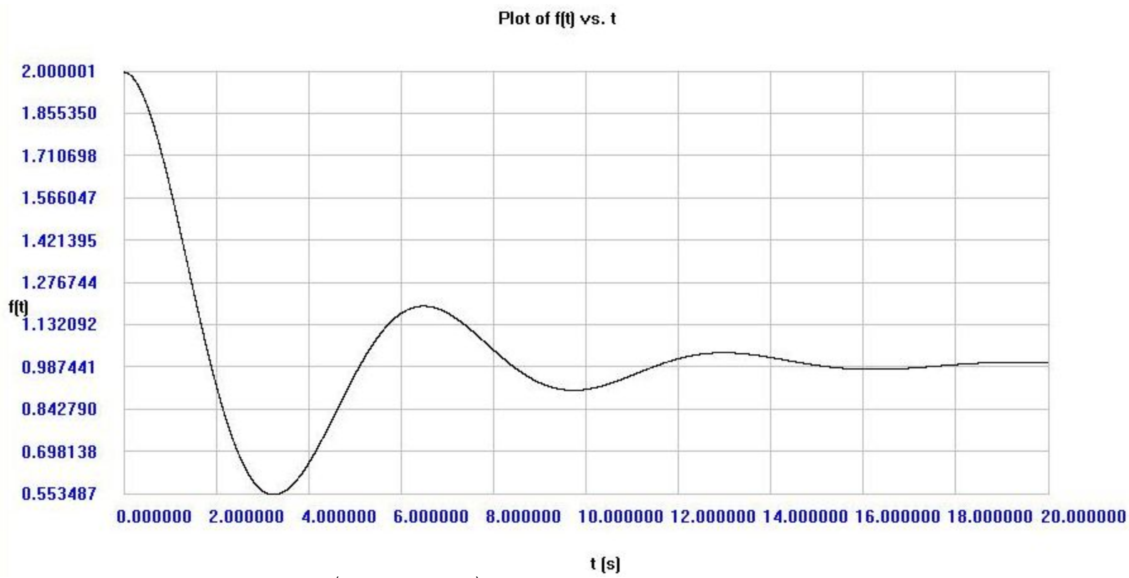

Figure 8 (a) Underdamped $(b^2 - 4km < 0)$ oscillatory vibration, where $y_g(t) \to 1.0$ as $t \to \infty$, for $u(t) = 1$, $m = 1 = k$, $b = 0.5$, $\mathbf{x}(0) = [y(0), \dot{y}(0)]^T = [2, 0]^T$ and $\delta t = 10^{-3}(\mathrm{s})$.

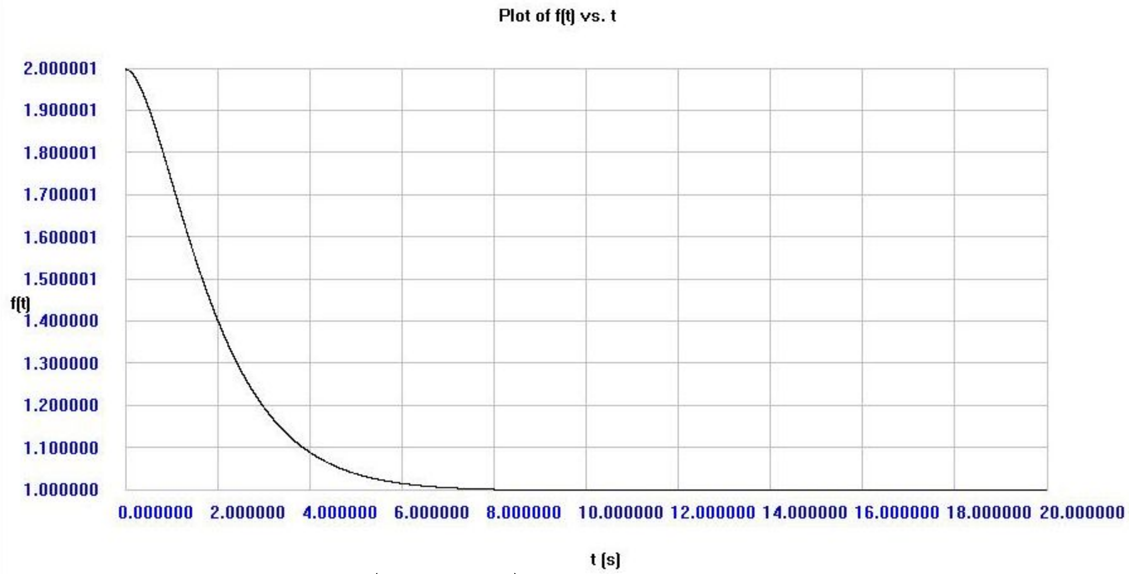

Figure 8 (b) Critically damped $\left(b^2 - 4km = 0\right)$ nonoscillatory vibration, where $y_{\mathrm{g}}(t) \to 1.0$ as $t \to \infty$, for $u(t) = 1$, $m = 1 = k$, $b = 2$, $\boldsymbol{x}(0) = [y(0), \dot{y}(0)]^{\mathrm{T}} = [2, 0]^{\mathrm{T}}$ and $\delta t = 10^{-3}(\mathrm{s})$.

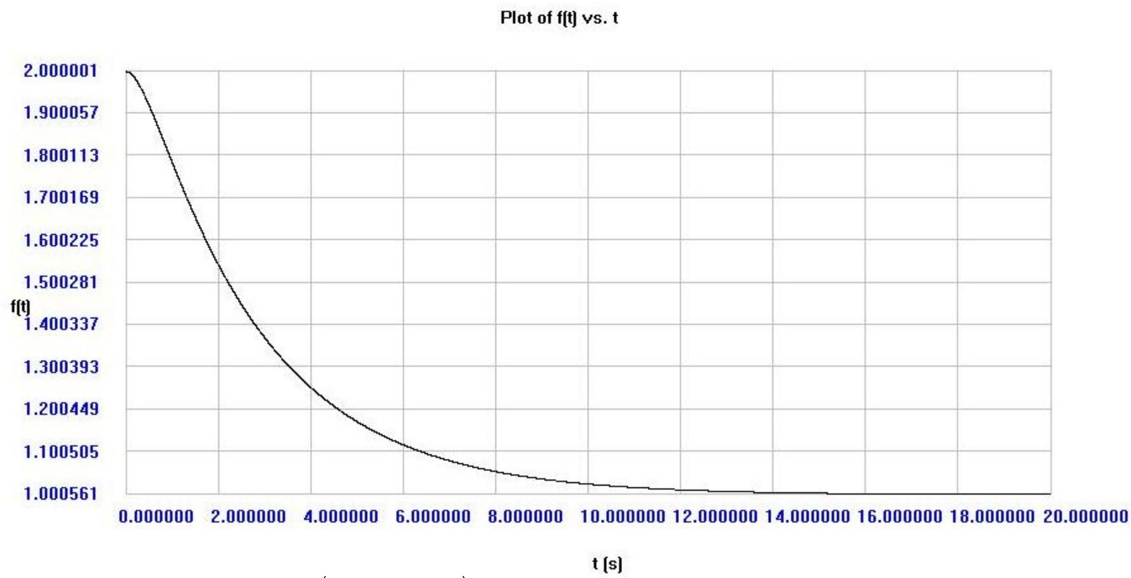

Figure 8 (c) Overdamped $\left(b^2 - 4km > 0\right)$ nonoscillatory vibration, where $y_{\mathrm{g}}(t) \to 1.0$ as $t \to \infty$, for $u(t) = 1$, $m = 1 = k$, $b = 3$, $\boldsymbol{x}(0) = [y(0), \dot{y}(0)]^{\mathrm{T}} = [2, 0]^{\mathrm{T}}$ and $\delta t = 10^{-3}(\mathrm{s})$.

3.2 Nonlinear Dynamical Systems

A coupled nonlinear system involves equations that are nonlinear in the variables for which the system is to be computed. Consider the Lotka-Volterra system consisting of two coupled first order nonlinear differential equations, describing the population dynamics of predator-prey interaction, presented in [4],

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{10a}$$

$$\frac{dy}{dt} = -\gamma y + \delta xy \tag{10b}$$

where, $x(t)$ and $y(t)$ are the populations of the prey and predator respectively, t represents the independent time variable, and α and γ are the rates of growth of the prey and predator respectively, and β and δ are the rates of competitive efficiency for the prey and predator species, respectively, where $\alpha, \beta, \gamma, \delta > 0$. A block diagram representation of this system (10) and its DiagramEng implementation are shown in figures 9 and 10 respectively, where the input signals are the growth rates α and γ, which may be initially chosen given a condition of no interaction $(\beta, \delta = 0)$ between the species (block numbers appear beneath the blocks).

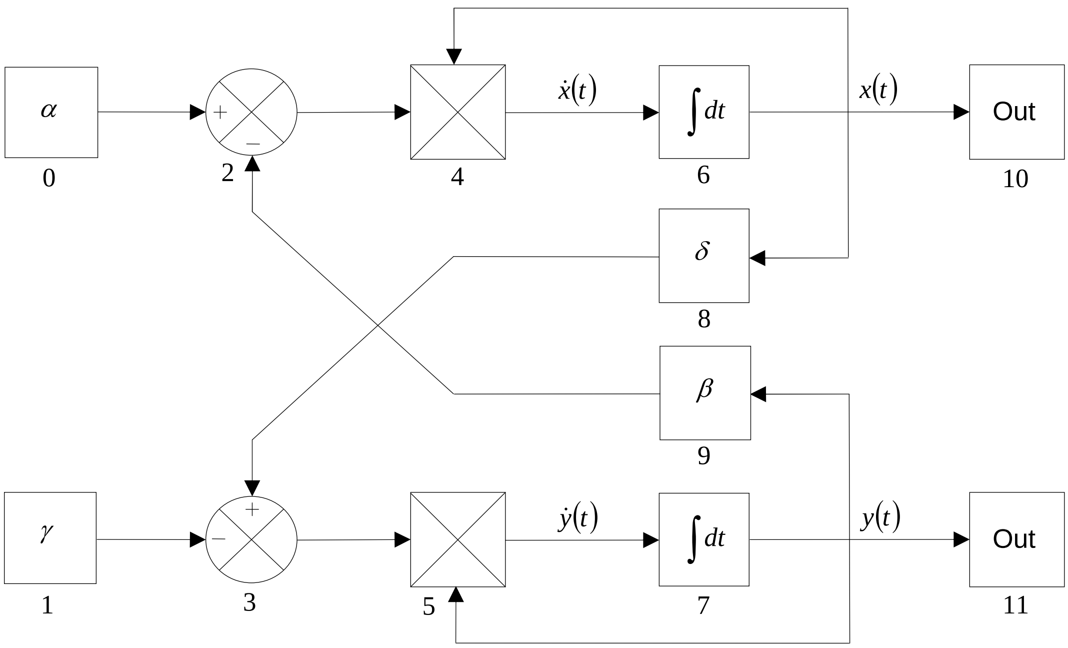

Figure 9 A block diagram representation of the Lotka-Volterra system of two coupled first order nonlinear differential equations (10).

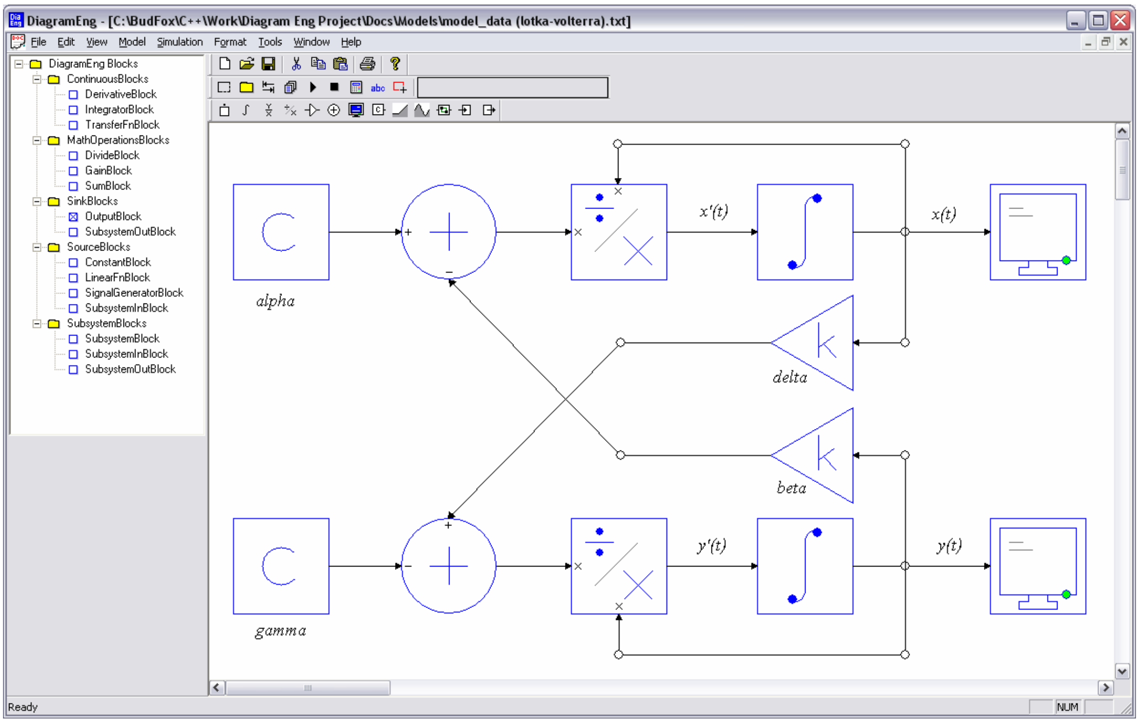

Figure 10 The block diagram representation of the Lotka-Volterra system made using DiagramEng.

In the case where $\beta, \delta = 0$, the equations of the populations are

$$\frac{dx}{dt} = \alpha x \quad \Rightarrow \quad x(t) = C_1 e^{\alpha t} \tag{11a}$$

$$\frac{dy}{dt} = -\gamma y \quad \Rightarrow \quad y(t) = C_2 e^{-\gamma t} \tag{11b}$$

where C_1 and C_2 are constants, and $x(t)$ and $y(t)$ are exponentially increasing and decreasing functions of time, for the prey and predator populations, respectively: the population of the prey in the absence of the predator increases and that of the predator decreases.

Mathematicians familiar with the study of nonlinear dynamical systems and chaos, see e.g., the texts [4] and [5], will recognize that the fixed points occur when the populations are in equilibrium, i.e., when $\dot{x}(t) = 0$ and $\dot{y}(t) = 0$, resulting in two such points: $(x_0, y_0) = (0, 0)$ and $(x_1, y_1) = (\gamma/\delta, \alpha/\beta)$. The stability of these points may be determined by observing the eigenvalues of the Jacobian matrix of the system (10), i.e.

$$\boldsymbol{J}(x, y) = \begin{bmatrix} \partial\dot{x}/\partial x & \partial\dot{x}/\partial y \\ \partial\dot{y}/\partial x & \partial\dot{y}/\partial y \end{bmatrix} \tag{12a}$$

$$= \begin{bmatrix} \alpha - \beta y & -\beta x \\ \delta y & \delta x - \gamma \end{bmatrix}. \tag{12b}$$

For $(x_0, y_0) = (0, 0)$, the eigenvalues of $\boldsymbol{J}(x, y)$ are $\lambda_1 = \alpha$ and $\lambda_2 = -\gamma$, with corresponding eigenvectors $\boldsymbol{v}_1 = [1, 0]^{\mathrm{T}}$ (the unstable manifold, x axis) and $\boldsymbol{v}_2 = [0, 1]^{\mathrm{T}}$ (the stable manifold, y axis), respectively, and hence the critical point is a saddle point and the system is unstable, implying that the extinction of both species is unlikely.

For $(x_1, y_1) = (\gamma/\delta, \alpha/\beta)$, the eigenvalues of $\boldsymbol{J}(x, y)$ are $\lambda_{1,2} = \pm i\sqrt{\alpha\gamma}$, i.e. they are purely imaginary $(\mathrm{Re}(\lambda_{1,2}) = 0)$ indicating the presence of a centre (in the positive quadrant) rather than a spiral, and Kibble states that as a result, there are cyclic variations in $x(t)$ and $y(t)$ which are not in phase [4] (the population of the predators grows whilst that of the prey declines and vice versa).

A simulation of the Lotka-Volterra system (10) was made with the parameters: $\alpha = 2$, $\gamma = 2$, $\beta = 1$ and $\delta = 0.5$, where the initial conditions of integration were $x(0) = 20$ and $y(0) = 10$, the initial output signals for the Divide blocks (which must be set since the Divide blocks are involved in two feedback loops) were $x_4(t_0) = -8$ and $x_5(t_0) = 8$, and a time-step size of $\delta t = 10^{-4}(\mathrm{s})$ was chosen. The cyclical variations in the populations of the prey and predators are shown in figures 13 (a) and 13 (b) respectively, where the population of the prey leads that of the predator, i.e., the variations are in fact not in phase.

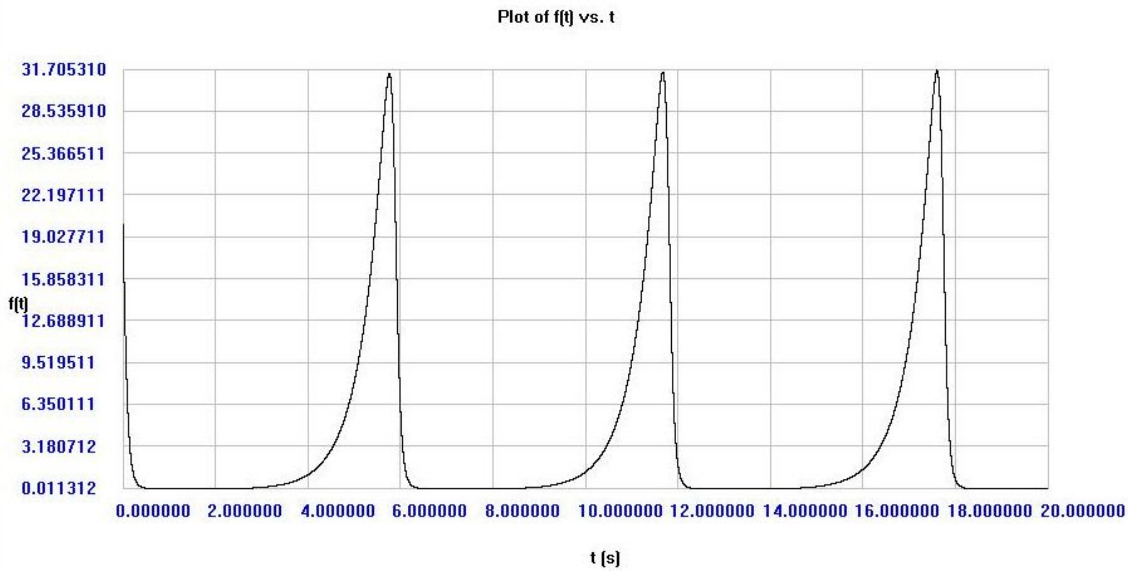

Figure 11 (a) Cyclic variations in the prey population for the Lotka-Volterra system (10) with paramters: $\alpha = 2$, $\gamma = 2$, $\beta = 1$ and $\delta = 0.5$, where $x(0) = 20$ and $y(0) = 10$, and $\delta t = 10^{-4}(\mathrm{s})$.

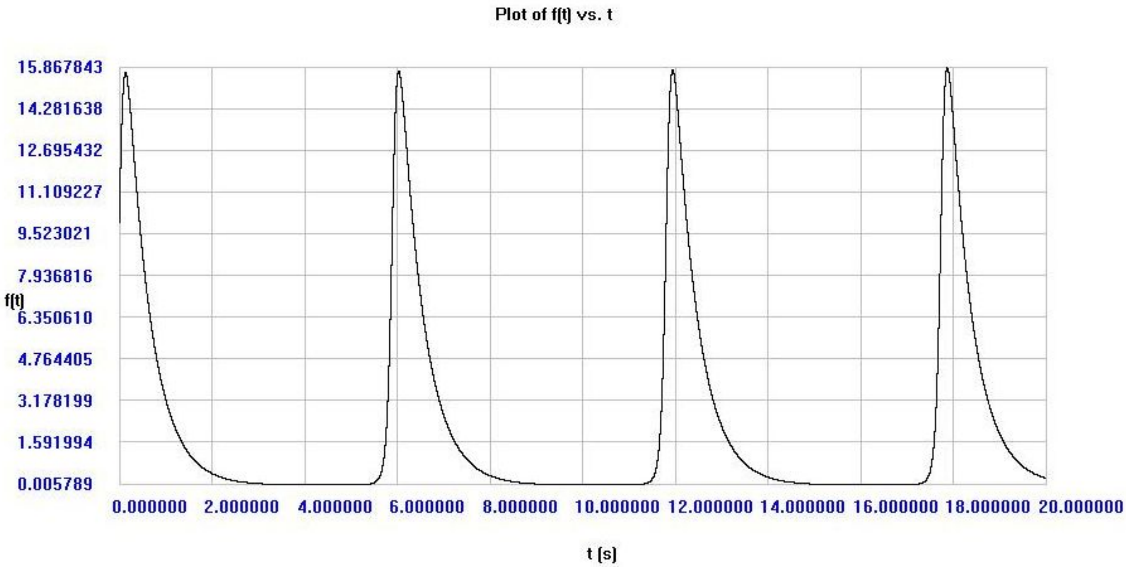

Figure 11 (b) Cyclic variations in the predator population for the Lotka-Volterra system (10) with paramters: $\alpha = 2$, $\gamma = 2$, $\beta = 1$ and $\delta = 0.5$, where $x(0) = 20$ and $y(0) = 10$, and $\delta t = 10^{-4}(\mathrm{s})$.

The phase portrait of the population of the predator vs. the prey, i.e., $y(t)$ vs. $x(t)$ may be generated by saving the output data through the Output blocks and plotting the two population values against each other as shown in Fig. 12 below (using a third-party graphical application). A saddle point resides at the origin $(x_0, y_0) = (0, 0)$ and a centre is present at $(x_1, y_1) = (\gamma/\delta, \alpha/\beta)$. As the prey declines in number, the predator grows, and vice versa, and neither species becomes extinct. As $t \to \infty$ it is observed that the trajectories do in fact form a centre.

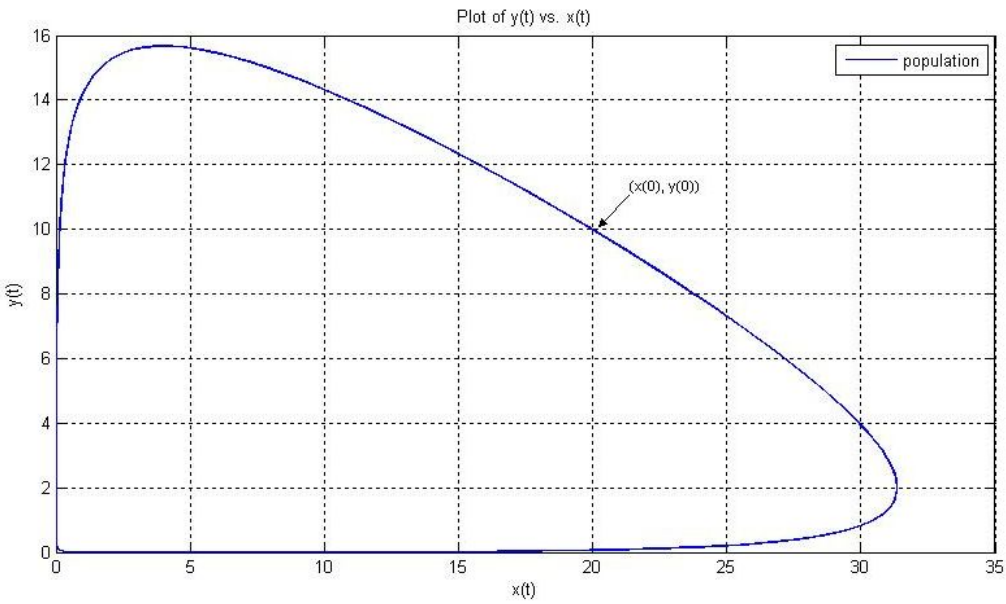

Figure 12 Phase portrait of $y(t)$ vs. $x(t)$ showing the change in population of the predator vs. the prey, where $(x(0), y(0)) = (20, 10)$: the saddle point is at $(0, 0)$ and the centre at $(\gamma/\delta, \alpha/\beta)$.

4. OUTPUT

The output shown in the figures above is obtained by double-clicking the Output block and selecting the Show Graph button on the OutputBlockDialog dialog window shown in Fig. 13 below. If the underlying numerical data are desired then these may be saved by selecting the Save Data button and specifying the appropriate file name and location. Then, a third-party application may be used to plot the data of different Output blocks against each other, as has been done to produce the phase portrait of Fig. 12 above.

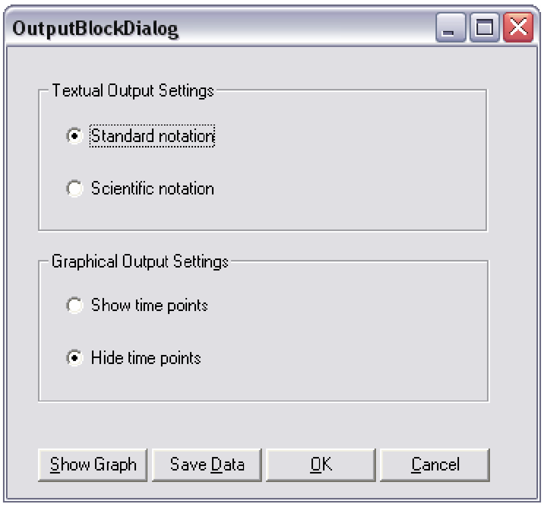

Figure 13 Output block dialog window allowing the user to view the graphical results (Show Graph) or save the underlying data (Save Data).

4.1 Output Block Data File

Consider an Output block-based recorded data matrix of the following form

$$\boldsymbol{M} = \left[\begin{bmatrix} f_1(t_0) & f_2(t_0) \\ f_3(t_0) & f_4(t_0) \end{bmatrix} \begin{bmatrix} f_1(t_1) & f_2(t_1) \\ f_3(t_1) & f_4(t_1) \end{bmatrix}, \ldots, \begin{bmatrix} f_1(t_n) & f_2(t_n) \\ f_3(t_n) & f_4(t_n) \end{bmatrix} \right] \tag{13}$$

where, $f_s(t)$, for $s \in \{1,..,4\}$ (four signals are used here for simplicity), are the individual signals being recorded for each time point $t_i \in [t_0, t_n]$, for initial and final simulation time points, t_0 and t_n, respectively. The data are written to an output file, with default name, “output_data.txt”, where each row of data in the output file corresponds to all signal output for a particular time point t_i and the file is of the following form

$$\begin{array}{c} t_0,\ f_1(t_0),\ f_2(t_0),\ f_3(t_0),\ f_4(t_0) \\ t_1,\ f_1(t_1),\ f_2(t_1),\ f_3(t_1),\ f_4(t_1) \\ \vdots \\ t_n,\ f_1(t_n),\ f_2(t_n),\ f_3(t_n),\ f_4(t_n). \end{array} \tag{14}$$

If there is no data in the output matrix, then the number of rows and columns are zero and only the time points corresponding to the system model simulation parameters will be written to the output file.

4.2 Model Data File

The model data file used to record all system model elements that specifies the geometry of the model and its underlying properties, is different to the aforementioned numerical output data file. The user need not be concerned with the model data file format but should be aware that the saving and restoring of a system model is performed using a user-specified text file, which has a default name: “model_data.txt”.

5. NON-FUNCTIONAL ITEMS

The current initial version of the DiagramEng software application allows the user to perform general modelling and simulation activities with the essential mathematical features. However, some functional elements exist that require additional work for their completion and are left to the second version of the software. The incomplete application items are shown in Table 17 below.

Table 17 Incomplete application items.

Item	Status
Blocks	Subsystem, Subsystem In, Subsystem Out and Transfer Function blocks currently do not perform data operations.
Model Menu	The Build Subsystem entry does not function as subsystem blocks are not implemented.
Numerical Solver	The absolute and relative error tolerance parameters are not currently used, the Time Step Type is “Fixed-step” and the Integration Method is “Euler (1st Order).”

SUMMARY

The topics covered in the “UsingDiagramEng.pdf” document include: 1) the Main frame-based menu, Child frame-based menu and Context menu, 2) the Main frame-based toolbar and the Child frame-based Common Operations and Common Blocks toolbars, 3) two examples concerning ordinary differential equations and nonlinear dynamical systems, 4) forms of output data including numerical simulation data (“output_data.txt”) and system model data (“model_data.txt”) and 5) non-functional elements that will be completed in the next version of the software.

REFERENCES

1. Microsoft Visual C++ 6.0, Microsoft® Visual Studio™ 6.0 Development System, Professional Edition, Microsoft Corporation, 1998.
2. Ogata, K., *Modern Control Engineering*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2002.
3. Salas, S.L. and Hille, E., *Calculus: One and Several Variables*, *(Complex Variables, Differential Equations Supplement)* 6th ed., John Wiley and Sons, New York, NY, 1990.
4. Kibble, T.W.B. and Berkshire, F.H., *Clasical Mechanics*, 5th ed., Imperial College Press, London, 2004.
5. Strogatz, S.H., *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Addison-Wesley, Reading MA, 1994.