Visual Simulation of Dynamic Economic Models

Richard D. HERBERT

Department of Computing, MPCE
Macquarie University
Sydney, 2109
Australia
email: ric@mpce.mq.edu.au

Rodney D. BELL

Department of Computing, MPCE
Macquarie University
Sydney, 2109
Australia
email: rod@mpce.mq.edu.au

Abstract This paper examines the problem of simulating a wide range of dynamic economic models in a coherent and easy to use software environment. It considers the use of visual simulation techniques with dynamic economic models. It shows how such techniques can aid in building both the model and its simulation environment. It also shows how the visualisation can aid in the comprehension of these in an analogous manner in the way it can aid to the comprehension of data. Using three sample models, the papers examines how visual simulation can be useful for dynamic models. The first illustration uses a difference equation model in state space form; the second uses a low-order, nonlinear, continuous-time model; and, the third a large, nonlinear, discrete-time macroeconometric model of the Australian economy.

1. INTRODUCTION

This paper considers the visual simulation of dynamic economic models. Such simulation is considered through illustration with a variety of dynamic economic models and a particular visual simulation environment software.

The paper uses the MATLAB (Mathworks, 1992) and SIMULINK (Mathworks, 1994) simulation environment. The MATLAB and SIMULINK software has powerful data manipulation and visualisation features. Also available are software libraries for specific use in control, statistics and neural networks, for example. One of its most useful features is that programs can be ported to different computing hardware platforms without change. This software environment is typical of a number of visual simulation environment software packages. Others include BUILDSIM by *Tangent Systems* and VISSIM by *Visual Solutions*.

Three illustrations are presented in this paper. The first illustration uses a low-order, difference equation model. The model is constructed in the MATLAB environment and then simulated in the SIMULINK environment in a state space form. The second illustration uses a low-order, nonlinear model. It constructs a simulation environment where the parameters of the model may be easily altered, so that their implications on model can be easily examined visually. The final illustration uses a large macroeconometric model based on the Murphy Model of Australia.

2. DISCRETE LINEAR MODEL

This illustration shows the construction and simulation of a low-order discrete model. The construction is undertaken in MATLAB, and the model is simulated in SIMULINK. The model is a two input, two output model, with the inputs being contemporaneous with the outputs. The model is quarterly, and formed by lagging each output for one year.

The outputs are the inflation rate (INF) and the unemployment rate (URT); and the inputs are a fiscal policy (F) and a monetary policy control (M). All variables are stationary except the money supply, M. It is transformed into first differences in logs.

The model is

$$\begin{aligned} \mathbf{y}_t = A_1\mathbf{y}_{t-1} + A_2\mathbf{y}_{t-2} + A_3\mathbf{y}_{t-3} \\ + A_4\mathbf{y}_{t-4} + B_0\mathbf{u}_t \end{aligned} \tag{1}$$

where $A_i, B_0 \in \Re^{2\times 2}, i = 1,2,3,4$; $\mathbf{y}_t = [\mathrm{INF}_t, \mathrm{URT}_t]^T$; and, $\mathbf{u}_t = [\mathrm{F}_t, \Delta \log \mathrm{M}_t]^T$.

The model may be written in standard state space

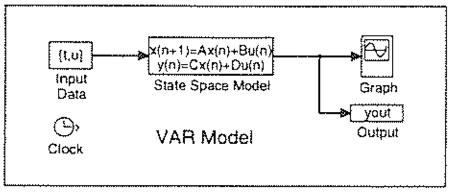

Figure 1. Simulation Environment for Linear Model

form (Herbert *et al.*, 1997) as

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t \quad (2)$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t \quad (3)$$

with $\mathbf{x}_t = [\mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \mathbf{y}_{t-3}, \mathbf{y}_{t-4}]^T$. The system matrices are

$$A = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ A_4 & A_3 & A_2 & A_1 \end{bmatrix} \quad (4)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ B_0 \end{bmatrix} \quad (5)$$

$$C = [A_4 \quad A_3 \quad A_2 \quad A_1] \quad (6)$$

$$D = B_0 \quad (7)$$

with I and 0 being the 2×2 identity and zeros matrices respectively.

The SIMULINK visual simulation environment for the model is given in Figure 1. This Figure is a simulation model within the overall simulation environment, and exists within a window on the user's workstation. This simulation environment window contains a series of menus for the control of the simulation. The Figure shows the simulation environment to be made up of a number of objects and data flow lines between the objects. The simulation takes data from the *Input Data* table object, inputs it into the *State Space Model* object, and the output is plotted as the simulation progresses by the *Graph* object. The *Graph* object opens a window where the data flow is plotted as it receives data. As, in this case, the data flow is a vector, then multiple lines will be plotted. The output is also stored for post simulation analysis in the *yout* storage object. The *Clock* object, if selected, opens a window that displays the simulation time.

The instantiation of object is through dialogues boxes that are obtain when the object is selected. Selecting the *Graph* object opens a dialogue box where the the initial parameters (such as line colours and styles) are set. Similarly, selecting the *State Space Model* object opens a dialogue box where the system matrices and sample time is entered. In this simulation the system matrices are stored as variables in the MATLAB workspace. The simulation environment of Figure 1 can,then, simulate any discrete-time state space model. The objects and data flow lines will automatically adjust to the different dimensionality of the model.

The *Input Data* object contains a table of the exogenous variables for each quarter over the entire time horizon. If the time step for the solution routine is set to 1 quarter, then within time horizon simulation uses data directly from the table. It also allows for shorter time steps, if this is considered appropriate, by interpolating data between quarterly table entries. Model prediction (i.e. simulation beyond the time horizon of table entries) uses extrapolation from the table.

The environment can be easily altered by the 'dragging and dropping' of additional objects and data flow lines. It is a simple task to make another *Graph* object, and connect a data flow line so that the exogenous input data can also be plotted.

3. NONLINEAR MODEL

3.1 Model Simulation

This illustration considers a continuous-time, dynamic model in an exercise to consider the implications of parameter variation on the solution of a nonlinear economic model. A simulation is set up where the parameters may be easily altered while the simulation is in process so the the effects can be examined visually. For this exercise a model is used that has a cyclical steady state. The model can be simulated for a very long time horizon. The model is also highly dependent upon the relationship of the parameter values and the objective is to alter them during the simulation to visually show this relationship.

The economic model is a dynamic nonlinear model based on the Lotka-Volterra equations. The model is

$$\frac{d}{dt}y(t) = \alpha_1 y(t) - \alpha_2 y(t)r(t) + \alpha_3 f(t) \quad (8)$$

$$\frac{d}{dt}r(t) = -\beta_1 r(t) + \beta_2 y(t)r(t) - \beta_3 m(t) \quad (9)$$

where y is economic output; r is interest rates; f is a fiscal policy instrument; m is a monetary policy instrument; and, the the parameters are represented by Greek letters. The model is a 'predator-prey' model with the addition of the fiscal and monetary policy forcing terms. The model is a dynamic IS-LM model and could be used to explain short-run variations around the steady state levels of the variables. The nonlinearity is easily seen by considering $r = f = m = 0$ where y (the

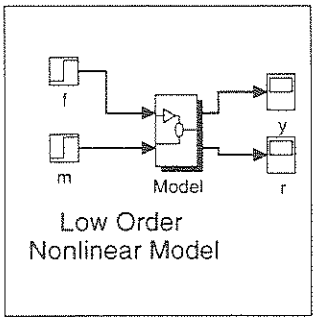

Figure 2. Overall Model Simulation

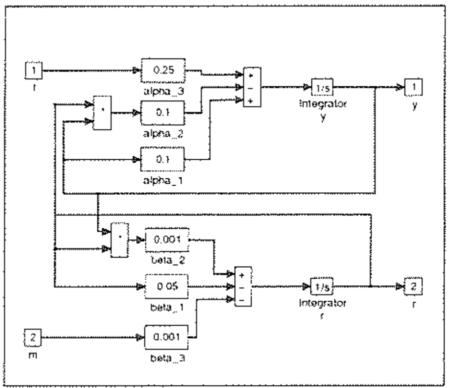

Figure 3. The Model (Equations 8 and 9)

'prey') grows exponentially at a rate α_1: and when $y = f = m = 0$, r (the 'predator') decays exponentially at a rate β_1.

Another example of economic Lotka-Volterra model is Goodwin's (1990) model of the class struggle. More examples can be found in Gabish and Lorenz (1987) and Lorenz (1989).

Figures 2 and 3 present the visual simulation of the model. Figure 2 is the overall simulation and shows the flow of data from the input objects through the model to the output objects. The *Model* object contains the details of the model. Selecting either output object, say y, opens a graphically output window which continually plots the variable. The input objects open dialogue boxes which allow for altering of the input variables, in this case applying step functions to the variables.

The details of the model are in the *Model* object. Here information hiding has been used, so that details do not clutter the environment. This is a major advantage of visual simulation – the overall structure of the simulation can be seen without the complex details.

Figure 3 visually shows the model as given in Equations 8 and 9. It shows the flow of the (scalar) data through the objects. It was developed by simply 'dragging and dropping' the objects, and then connecting them with the data flow lines. Again, objects, when selected, open dialogue boxes or

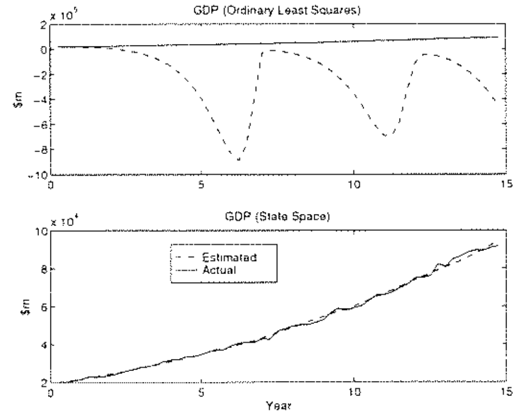

Figure 4. Nonlinear Model: GDP Parameters

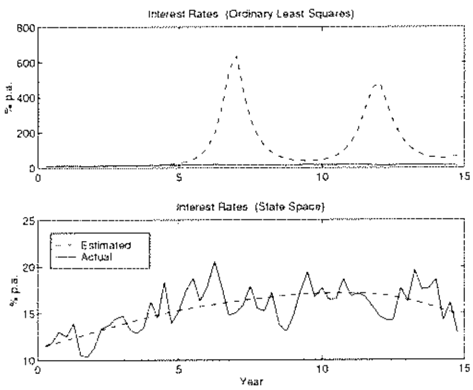

Figure 5. Nonlinear Model: Interest Rate Parameters

windows which allow for the object's instantiation to be altered (eg. the initial value of the integrator). Of particular relevance to this exercise is each parameter object opens a window containing a slider so that the parameter value can be altered by moving the slider.

The simulation model here allows for the parameters to be altered by moving a slider while the simulation is in progress and to visually examine the output as the parameter is altered. It also allows for the easy change of input variables so that their implications may be examined.

Another feature of the simulation model in SIMULINK is that easily allows for changes to be made in the numerical procedure used to solve the model. The solution routine is selected from a menu, and solution parameters (such as time step size) are easily modified. In this manner, the model can be easily used to illustrate the importance of numerical issues like truncation error.

3.2 Parameter Estimation

The issue of parameter estimation has to be addressed to obtain a useable model. The visual environment allows very powerful parameter estima-

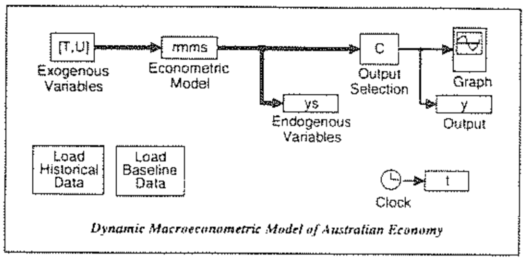

Figure 6. Visual Simulation Environment

tion procedures to be used. An optimisation toolbox is available with MATLAB and this has most of the commonly used linear and nonlinear estimation procedures included. For this nonlinear model a linear least squares estimation procedure based on one step ahead fitting to historical data was used to obtain initial parameter estimates for $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2$ and β_3. The actual historical data used was the ouput obtained from the simulation of the Macroeconometric Model outlined in Section 4 when using historical data as input. These parameter estimates were then used as a starting point for a nonlinear parameter estimation procedure. That procedure fitted the model's output to the data for the whole trajectory when the model is run-out from the initial conditions rather than a one step ahead prediction. Figures 4 and 5 show the fit that has been obtained by the two methods. Another advantage of this simulation environment is that the improvement in fit could be observed as the nonlinear parameter estimation procedure iterated towards the solution. This allowed early intervention if poor conditioning or an error in the algorithm was present.

4. MACROECONOMETRIC MODEL

The previous illustrations used low-order models. In this illustration the simulation is undertaken of a large, nonlinear, practical macroeconometric model. The model is an implementation of the Murphy model. The model is written in MATLAB and can be executed in the normal MATLAB environment. In the visual simulation environment, SIMULINK, it is considered as an object into which data (exogenous variables, and initial estimates of the endogenous variables) flows and from which output data (final endogenous variables) flows. It is possible for such objects to be written in other languages (eg. C) than MATLAB and be used as objects in SIMULINK.

4.1 The Model

The Murphy model is a widely used dynamic macroeconometric model of the Australian economy designed for policy analysis and forecasting. Following most current macroeconometric models, it is a discrete model based upon quarterly data. Details of the model can be found in Murphy (1988*a*), Murphy (1988*b*), Murphy (1990*a*), Murphy (1990*b*), and Powell and Murphy (1995). It has recently (Murphy, 1996) been extended to include much greater regional and industry details.

The Murphy model can be written as

$$\mathbf{X}_t = \mathbf{F}(\mathbf{X}_{t-i}, \mathbf{X}_t^e, \mathbf{Z}_{t-j}, \epsilon_t) \qquad i, j = 0, 1, 2, \cdots, N \quad t = 0, 1, 2, \cdots, T \tag{10}$$

where t represents the discrete-time period of a quarter within the time horizon $[0, T]$; N is the maximum lag length; $\mathbf{X}$ is a vector of endogenous variables; $\mathbf{X}^e$ is a vector of forward looking expectations variables; $\mathbf{Z}$ is a vector of exogenous variables; and, ϵ is a vector of error terms. Initial and lagged values of the endogenous variables are given; exogenous variables are known for the entire time horizon; the error terms statistical properties are known from the estimation procedures, and are treated as exogenous; and, forward looking expectations are model consistent and are predicted by the model. This requires the model to be solved multiple times for a time period.

The model has 96 endogenous variables and 90 exogenous variables. The equations of the model use, besides these variables for the current quarter, lagged exogenous and lagged endogenous variables. These lags vary from a single quarter to a year ($N = 4$). For a given quarter these lagged endogenous and lagged exogenous variables are predetermined, and by a suitable redefinition of variables, can be considered as current states of the model. The endogenous variables and the expectations variables can also be considered as states of the model. The model then consists of current states and current exogenous variables for the quarter. This gives a total of 258 variables for each quarter. The model is then in the form

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{z}_t) \tag{11}$$

where $\mathbf{x}$ is a vector of state variables, and $\mathbf{z}$ is a vector of current exogenous variables.

4.2 Model Solution

A visual simulation environment to run the model is presented in Figure 6. This simulation environment allows for the examination of the effects of exogenous variables on endogenous variables. The Figure shows the objects and the flow of data between the objects. Some data flows are vectors, and, in this illustration, are indicated by the wide lines. Others are scalars and are indicated by the narrow lines.

In Figure 6, the *rmms* object solves the model for a single time period. Its input is the control solution vector of predetermined variables (exogenous and lagged endogenous) and an initial estimate of the endogenous variables and it returns the shocked solution vector where the endogenous variables have been solved. As before, these variables can be considered as the current states ($\mathbf{x}$), and current exogenous variables ($\mathbf{z}$).

The *Load Historical Data* object in Figure 6, if selected, loads the data for the Historical Solution; whereas the *Load Baseline Data* object loads data for the Baseline Solution. The Historical Solution is the the model when solved using historical data, so that historical exogenous data is used as inputs and historical endogenous data is used as the initial estimate for the endogenous variables. The Baseline Solution is where all exogenous variables grow smoothly. The *[T,U]* table object contains this data as a table of the states and current endogenous variables for each quarter over the entire time horizon.

The *ys* storage object is a data sink for storage of the states. This data can then be used in post simulation analysis. It is available in the MATLAB environment. The matrix object C uses matrix manipulation of the endogenous data flow vector to select some of what are considered the model outputs of interest. In the Figure a single variable is selected, so C contains a 258×258 matrix with a single entry of one. The output is plotted over time as the simulation progresses by the *Graph* object; and, the outputs are stored by the *y* storage object for post simulation analysis. The *Clock* object displays the simulation time and the *t* storage object stores it for post simulation analysis.

4.3 Model Prediction

As, in the Baseline Solution variables grow smoothly, the extrapolation properties of the *[T,U]* table object allow the model using Baseline Data to be simulated over as long a time horizon as is desired. The model can be run out indefinitely. This has the substantial benefit that changes can be made to exogenous variables during a simulation, and the effects examined visually. This is not the case with the Historical Solution. Here extrapolation, as would be expected, generates poor estimates of 'real' data and the model quickly explodes.

4.4 Tax Analysis

A different simulation environment is given in Figure 7. This environment considers the implications of tax changes, ceteris paribus, on GDP and Employment as given by the model. The *GST* and *Y Tax* objects each allow for a single (step) change.

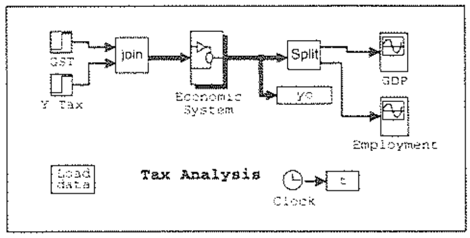

Figure 7. Tax Analysis Visual Simulation Environment

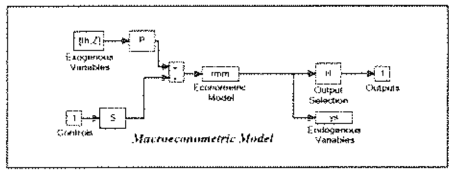

Figure 8. Economic System

These two scalar data flows are combined into a vector by the *join* object and are input into the *Economic System* object. The outputs from this object are stored for post simulation analysis in the *yo* storage object, and split into scalar data flows for individual plotting.

Selecting the *Economic System* object opens a window containing the model given in Figure 8. This differs from that of Figure 6 in that the exogenous variables are divided into those completely outside the control experiment and those that are considered as controls. The selection matrix objects P and S allow for the incorporation of the controls to the inputs into the macroeconometric model. Similarly, the matrix object H selects the endogenous variables that are to be considered as outputs. The extension from Figure 6 to Figure 8 is a simple visual step.

5. CONCLUSION

This paper has illustrated the visual simulation of dynamic economic models. A low-order, continuous-time model, a low-order, discrete-time model, and a high-order, nonlinear macroeconometric model have been illustrated. It has shown how experiments can be undertaken with such models.

The advantage of visual simulation is that it allows for the rapid prototyping of experiments, and the rapid and graphic generation of results. The relationship between the various objects can be seen, and this can aid in the explanation of the experiment.

Visual simulation was illustrated using MATLAB and SIMULINK software. This software is typical of a class of similar software which is currently available and can be of substantial benefit to simulating dynamic models.

6. REFERENCES

Gabish, Günter and Hans-Walter Lorenz (1987). *Business Cycle Theory: A Survey of Methods and Concepts*. Vol. 283 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag. Berlin, Germany.

Goodwin, Richard M. (1990). *Chaotic Economic Dynamics*. Clarendon Press. Oxford, UK.

Herbert, Richard D., Rodney D. Bell and Graham Madden (1997). Linear contemporaneous control models. In: *Third International Conference on Computing in Economics and Finance*. Society of Computational Economics. Stanford University, California, U.S.A.

Lorenz, Hans-Walter (1989). *Nonlinear Dynamical Economics and Chaotic Motion*. Vol. 334 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag. Berlin, Germany.

Mathworks (1992). *MATLAB: Reference Guide*. The Mathworks Inc. Natick, Massachusetts, USA.

Mathworks (1994). *SIMULINK: User's Guide*. The Mathworks Inc. Natick, Massachusetts, USA.

Murphy, Christopher W. (1988a). An overview of the Murphy Model. *Australian Economic Papers* pp. 175–199. Supplementary Conference Volume.

Murphy, Christopher W. (1988*b*). Rational expectations in financial markets and the Murphy Model. *Australian Economic Papers* pp. 61–88. Supplementary Conference Volume.

Murphy, Christopher W. (1990a). The macroeconomics of a macroeconometric model. Technical report. Australian National University. Canberra, Australia.

Murphy, Christopher W. (1990*b*). The Murphy Model of the Australian economy: The model in detail. Technical report. Australian National University. Canberra, Australia.

Murphy, Christopher W. (1996). MM2. In: *Modelling and Control of National and Regional Economies 1995* (Ljubisa Vlačić, Tom Nguyen and Dubravka Ćećez-Kecmanović, Eds.). pp. 9–18. Pergamon. Oxford, U.K.

Powell, Allan A. and Christopher W. Murphy (1995). *Inside a Modern Macroeconometric Model: A Guide to the Murphy Model*. Vol. 428 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag. Berlin, Germany.