Multi-rate and Integrated Package Simulation

John G Pearce
ISIM International Simulation Limited
161 Claremont Road
Salford, M6 8PA, UK
johnpearce@isimsimulation.com

Keywords: Multi-rate simulation, Continuous system simulation, ESL, Segments, VTB

Abstract

Many large complex systems can be broken down into a network of interconnected modules, where each module has particular dynamic properties, for example dominant time constants, extent of discontinuities etc. When simulating such systems, it is desirable to treat each module separately applying the most suitable numerical integration algorithm and step-length, and allow modules to communicate with each other at suitable time intervals. Multi-rate simulation is a well established technique which implements such a modular approach to simulation allowing each module to be modelled with the most appropriate simulation parameters. However, great care must be taken with the communication of data between modules to avoid the introduction of excessive time delays leading ultimately to instability. This paper continues a study of multi-rate simulation issues by making use of the ESL simulation language segment structures and the integration of ESL segments into the Virtual Test Bed (VTB) environment. A simple coupled three-network electrical circuit is used as an example to illustrate the multi-rate method and its implementation in ESL. A non-segmented solution is obtained for comparison with the multi-rate simulation to show the effects of choosing different step-lengths. Finally the system is simulated in the VTB environment using ESL generated entities to represent the three networks. This last stage provides an update on the state of an on-going project to integrate ESL with the VTB.

1. INTRODUCTION

The advantages of multi-rate simulation are well documented [1,2]. The technique consists in partitioning a large system into modules, where each module is characterised by a clearly defined dynamic range. Each module is then simulated using an appropriate time-constant. Communication is allowed to take place between the modules at a rate corresponding to the longest time-constant. Care has to be taken in partitioning the original system at appropriate boundaries and the methods used to transfer data between the modules at each communication point to avoid instability and aliasing errors.

This paper outlines to use of the segment feature of the ESL simulation language [3] to implement a multi-rate simulation. Since the objective is to demonstrate a general approach in this environment, a fairly simple three-network electrical circuit is considered for illustrative purposes. Each network is represented by an ESL *emulated* segment. The particular characteristic of this approach is the handling of data transfer between the segments. This involves both using inductive coupling between the segments and returning the values of the network state-variables at communication points and solving the algebraic coupling equations within the segment calling framework.

ESL *embedded* segments may be accessed as simulation entities from the Virtual Test Bed (VTB) environment [4]. The last part of the paper presents a solution of the multi-rate example using the VTB-ESL environment.

2. SEGMENTS IN ESL

ESL is an advanced high-level simulation package for modelling large-scale systems from a wide range of disciplines. ESL comprises two components: the language itself and a graphical user interface - the Integrated Simulation Environment (ISE). ESL is a continuous system simulation language and is used for modelling dynamic systems which are usually described by ordinary and partial differential equations. ISE provides the environment from which all stages of the simulation process can be managed. The software was developed mainly through a series of contracts with ESTEC - the European Space Research and Technology Centre - part of ESA with additional support from various industrial simulation consultancy activities.

A feature of ESL that lends itself to the realization of multi-rate simulation is its *segment* structures. Segments were originally included in ESL as a means of providing a parallel processing capability. The idea was that a large simulation could be broken down into self-contained segments that could be executed in parallel on different processors or networked computers. Communication takes place between segments at pre-determined communication points through a TCP IP protocol. ESL supports three types of segment:

- *Remote segments* – these can be assigned to different processors for truly parallel operation.
- *Emulated segments* – these allow parallel operation to be emulated on a single computer – useful for testing parallel segments before assignment to separate processors or for implementing multi-rate simulations on a single computer.
- *Embedded segments* – used where an ESL model is to be integrated with another application.

Remote segments were not used in the work reported here; however they may be used for implementing large-scale multi-rate simulations where parallel processing would provide a speed advantage. The emulated segment feature was used to implement the tri-rate system described in this paper. The embedded feature is used as the means of integrating ESL models into VTB simulations and is used here to illustrate multi-rate simulation in a multi-package environment.

3. THREE NETWORK CIRCUIT

In order to illustrate the multi-rate technique and its realization using ESL segments, a simple electrical circuit comprising three coupled networks is used. This particular circuit is an extension of one used in earlier studies on the stability of multi-rate methods [5]. The circuit is shown in Figure 1.

Figure 1 Tri-rate circuit

The component values are chosen such that the networks have significantly different time constants. A single-rate simulation would have to be executed with a step-length determined by the shortest time constant whereas the multi-rate simulation allows different step-lengths to be used for each network.

4. ESL SEGMENTED SOLUTION

The three networks may be described by the following equations:

Network A

$$Vc_1' = (i_1 + i_{ba})/C_1$$

$$i_1' = \left(-(R_1 + R_3)i_1 - Vc_1 + Vs_1 - Vs_2 - R_3 i_{ba}\right)/L_1$$

Network B

$$i_2' = \left(-R_2 i_2 - VS_3 + V_{ab} - R_2 i_{cb}\right)/L_2$$

$$Vc_2' = \left(-Vc_2 - Vs_4 + V_{ab}\right)/(R_4 C_2)$$

Network C

$$i_3' = \left(-R_6 i_3 - Vc_3 - Vs_6 + V_{bc}\right)/L_3$$

$$Vc_3' = i_3 / C_3$$

If the networks are modelled as three separate entities, Network A requires an input, i_{ba} form Network B; Network B requires inputs V_{ab} from Network A and i_{cb} from Network C; Network C requires an input V_{bc} from Network B, as shown in Figure 2.

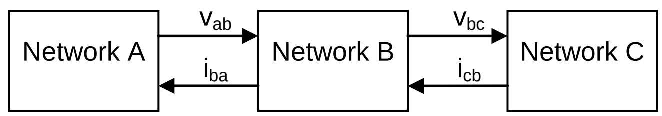

Figure 2 Partitioned system

The variable pairs: V_{ab}/i_{ba} and V_{bc}/i_{cb} are the *across* and *through* components of the natural coupling between the networks, where each variable of the pair depends algebraically on the other. Various techniques are available for resolving the coupling problem, such as the Latency Insertion Method (LIM) [6].

4.1. Inductive Coupling

The first solution presented here is to add an inductive coupling component between the networks. If the value of the inductor is small compared to other component values, it will introduce negligible error to the solution. However a small component value may cause the system to become stiff and necessitate the use of small step-lengths to avoid instability, defeating the object of multi-rate integration. Figure 3 shows the proposed solution.

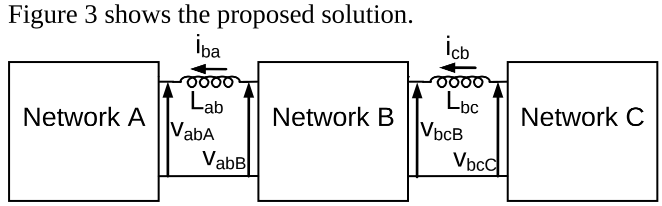

Figure 3 Inductive coupling

Here all three networks are solved to determine their terminal voltages (v_{abA}, v_{abB}, v_{bcB} and v_{bcC}) and the inter-segment currents are determined by solving the following differential equations:

$$i_{ba}' = (V_{abB} - V_{abA})/L_{ab}$$

$$i_{cb}' = (V_{bcC} - V_{bcB})/L_{bc}$$

where L_{ab} and L_{bc} are the values of the coupling inductors.

For convenience, in our solution, the coupling equations are included in the Network B code, leading to the ESL segmented architecture shown in Figure 4.

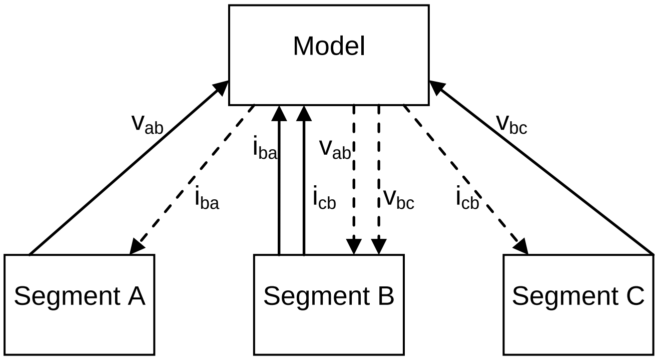

Figure 4 Inductive coupling segment architecture

In this solution, each network is modelled as an ESL emulated segment. In order to verify the architecture, the program is first run with all segments taking the same step-length. The results give good agreement with an un-segmented solution.

ESL segments are called from the top-level model communication region; therefore all inter-segment communication takes place through this region. Different integration step-lengths can be specified for each segment by setting the segment *Nstep* parameter (number of integration steps per communication interval). However, where segment calls are placed directly in the model communication region, all segment communication takes place at the same rate – that specified by the model communication interval *Cint*. This is fine if there are only two segments or if there are three or more segments which communicate at the same rate, but the approach breaks down where different inter-segment communication rates are required.

In a situation where there are three or more segments with *different* intersegment communication rates, the solution is to place *if* statements around the segment calls in the model communication region so that segments can be called every two, three or *n* model communication periods. This is easily achieved through the use of a counter and modulus function. The only restriction is that the inter-segment communication periods must have a common factor, to which the model *Cint* parameter is set.

In our example, suitable step-lengths for networks A, B and C are 0.001*s*, 0.02*s* and 0.01*s* respectively (corresponding the *Nstep* values of 20, 1 and 2). Therefore a single model communication rate of 0.02*s* is appropriate.

Plots of the three network state variables for un-segmented and segmented programs are presented in Figure 5, Figure 6 and Figure 7. Small but significant differences are apparent between the reference un-segmented solution, particularly in the Segment A v_{c1} variable.

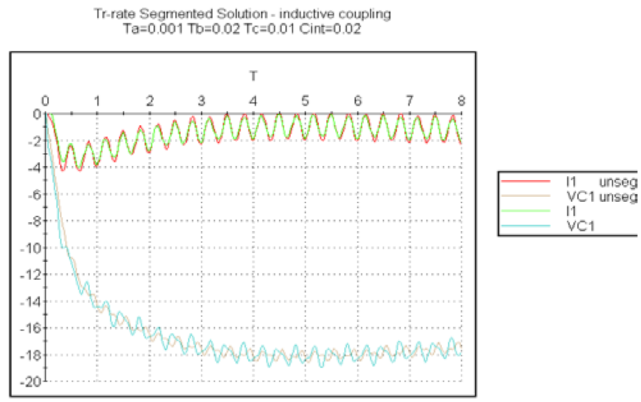

Figure 5 Segmented states for Network A - inductive coupling

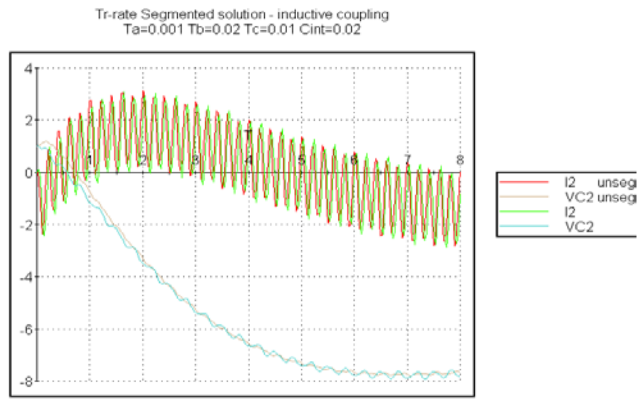

Figure 6 Segmented states for Network B - inductive coupling

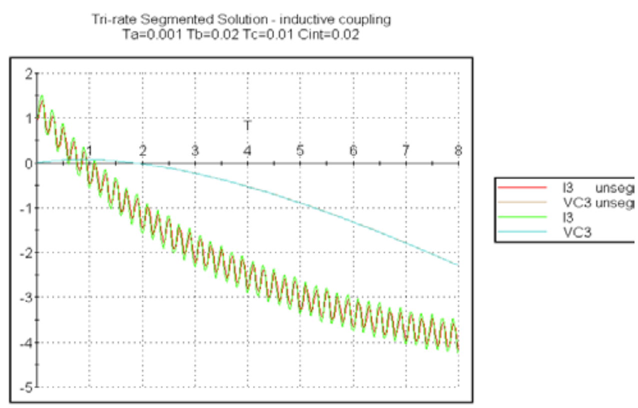

Figure 7 Segmented states for Network C - inductive coupling

4.2. State Variable Coupling

An alternative approach which does not require the addition of coupling components is described. In this approach we simply solve the algebraic relationships between the across and through variables at the network interfaces utilizing the state variables associated with each network, giving the segmented architecture shown in Figure 8. In this arrangement, the following equations are solved at each communication point in the top-level model:

$$V_{ab} = \left(Vc_1 + Vs_2 + R_3(i_1 - i_2) + Vs_4 R_3 / R_4 + Vc_2 R_3 / R_4\right) / \left(1 + R_3 / R_4\right)$$
$$i_{ba} = \left(-i_2 + \left(Vc_2 + Vs_4 - Vc_1 - Vs_2 - R_3 i_1\right) / R_4\right) / \left(1 + R_3 / R_4\right)$$
$$V_{bc} = \left(Vs_3 + R_2(i_2 - i_3) + Vs_5 R_2 / R_5\right) / \left(1 + R_2 / R_5\right)$$
$$i_{cb} = \left(-i_3 - \left(Vs_3 + i_2 R_2 - Vs_5\right) / R_5\right) / \left(1 + R_2 / R_5\right)$$

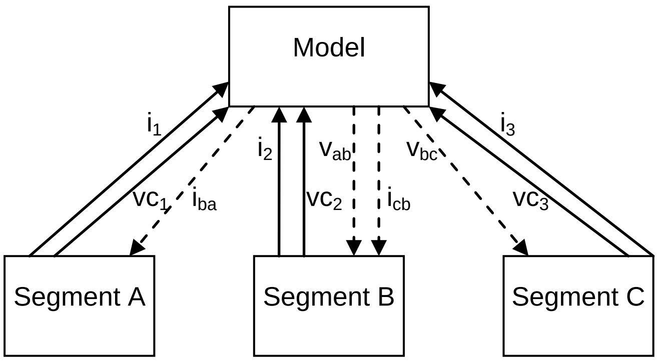

Figure 8 State variable segment architecture

Plots of the three network state variables for un-segmented and segmented programs are presented in Figure 9, Figure 10 and Figure 11. Significant improvement in the agreement between the un-segmented and segmented results is observed.

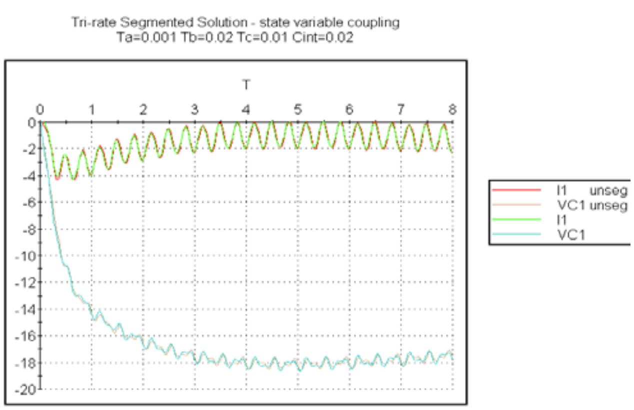

Figure 9 Segmented states for Network A – state variable coupling

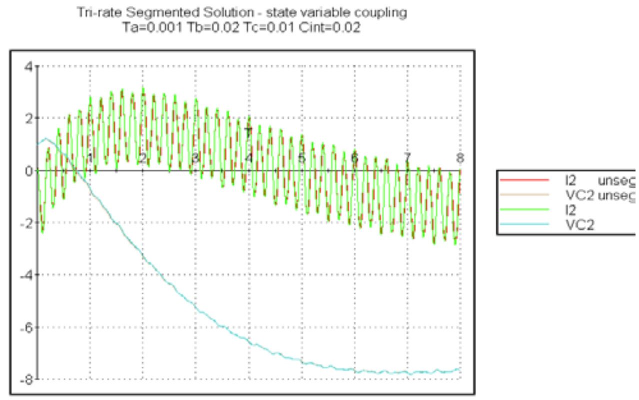

Figure 10 Segmented states for Network B – state variable coupling

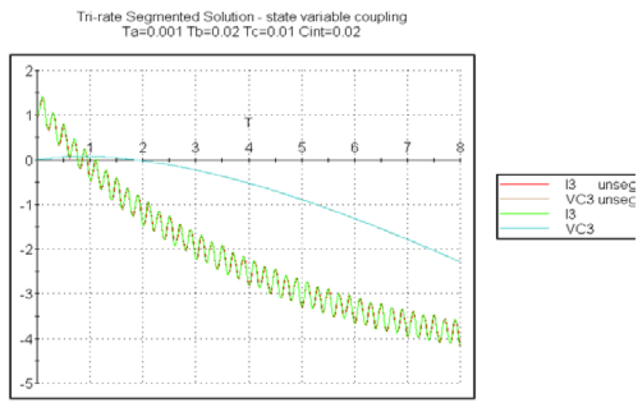

Figure 11 Segmented states for Network C – state variable coupling

5. VTB SOLUTION

The objective of a VTB solution to the current problem is two-fold: (a) to provide an update on the integration of ESL and the VTB and (b) to demonstrate a VTB-ESL multi-rate example. The state variable coupling method only is presented here.

The method of accessing ESL models from the VTB is described elsewhere [7], but summarized here. Basically a VTB simulation entity is created from an ESL embedded segment. This is achieved by creating a .NET CLR assembly from an ESL embedded segment, using a standard utility (*Eslgen*). The VTB entity is then created from the .NET assembly using the VTB *Entity Designer* program. The arrangement is shown in Figure 12.

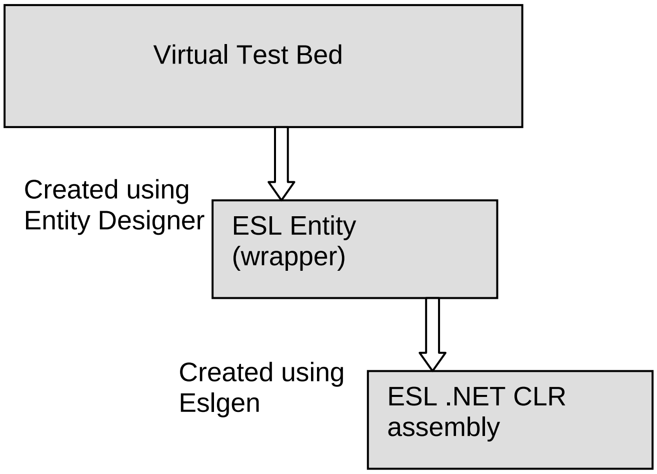

Figure 12 The ESL - VTB interface

The embedded segment for Network A is shown in Figure 13. The other two segments are similar. Communication with the embedded segment is through the *Esl_imp* and *Esl_state* package structures. Variables defined in these structures become visible in the generated .NET assemblies and are therefore accessible from VTB.

```
Embedded
    Package Esl_inp;
        Real:iba;
    End Esl_inp;
--
    Package Esl_state;
        Real:i1_out,vc1_out;
    End Esl_state;
--
Segment NetworkA;
    Constant Real:R1/2.0/, L1/0.5/, C1/0.0225/, R3/8.0/;
    Real: VS1, VS2/20.0/,i1,vc1,pi;
    Use Esl_inp;
    Use Esl_state;
--
    Initial
        Algo := 8;
        Nstep:=20;
        pi:=4.0*atan(1.0);
        VS1 := 10*sin(2*pi*3*t);
        i1:=1.0;
        vc1:=1.0;
    Dynamic
        vc1' := (i1+iba)/C1;
        i1' := (-(R1+R3)*i1-vc1+VS1-VS2-R3*iba)/L1;
    Step
        VS1 := 10*sin(2*pi*3*t);
    Communication
        i1_out:=i1;
        vc1_out:=vc1;
End NetworkA;
```

Figure 13 Embedded Network A code

In order to replicate the ESL segmented solution, it is necessary to arrange for the algebraic coupling equations to be executed between calls to the ESL Network entities. This could be achieved by implementing the equations directly using native VTB arithmetic elements, as shown in Figure 14. Clearly, this results in a cluttered schematic with much scope for error. A better solution is to encapsulate the algebraic calculations in fourth VTB entity. Figure 15 shows the code for such an entity and Figure 16 shows how the entity is combined with the network entities in a VTB schematic.

Although each of the three ESL Network entities will run with their own step-length (specified by setting the parameter *Nstep* – number of integration steps per communication interval) thus satisfying the multi-rate requirement, a current limitation of the VTB approach is that all entities are invoked at the same rate. This means that in this instance the communication rate for Network A/Network B must be the same as for Network B/Network C (which is the case for our choice of step-lengths).

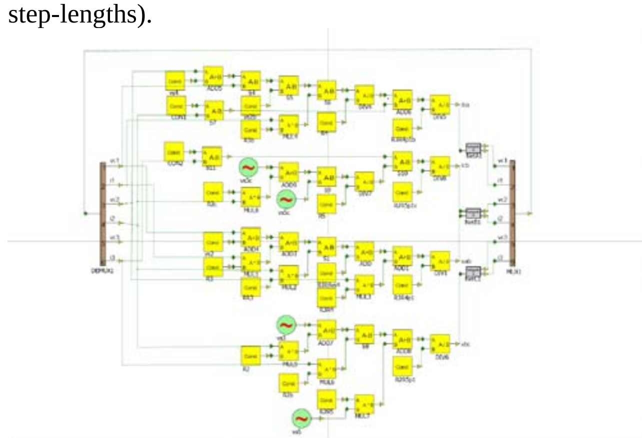

Figure 14 Coupling equations implemented in VTB

```
Embedded
        Package Esl_inp;
                Real:i1,vc1,i2,vc2,i3,vc3;
        End Esl_inp;
--
        Package Esl_out;
                Real:iba,vab,icb,vbc;
        End Esl_out;
--
Segment Top;
```

```
        Constant Real:
R1/2.0/,R2/4.0/,R3/8.0/,R4/1.0/,R5/2.0/,R6/3.0/;
        Constant Real:
L1/0.5/,L2/3.0/,L3/15.0/,C1/2.25e-2/,C2/1.5/,C3/7.5/;
        Real: VS2/20.0/,VS4/10.0/,VS3,VS5,pi;
        Use Esl_inp;
        Use Esl_out;
        Initial
                Algo := 8;
                Nstep:=1;
                pi:=4.0*atan(1.0);
                iba := -2.0;
                icb := -1.0/3.0;
        Dynamic
        Communication
                VS3:= 20*sin(2*pi*3*t);
                VS5:= 200*sin(2*pi*5*t);
                vab:= (vc1+VS2+R3*i1-
R3*i2+R3/R4*VS4+vc2*R3/R4)/(1+R3/R4);
                iba:= (-i2+(vc2+VS4-vc1-VS2-
R3*i1)/R4)/(1+R3/R4);
                vbc:= (VS3+i2*R2-
i3*R2+R2/R5*VS5)/(1+R2/R5);
                icb:= (-i3-(VS3+i2*R2-
VS5)/R5)/(1+R2/R5);
End Top;
```

Figure 15 Embedded algebraic coupling entity code

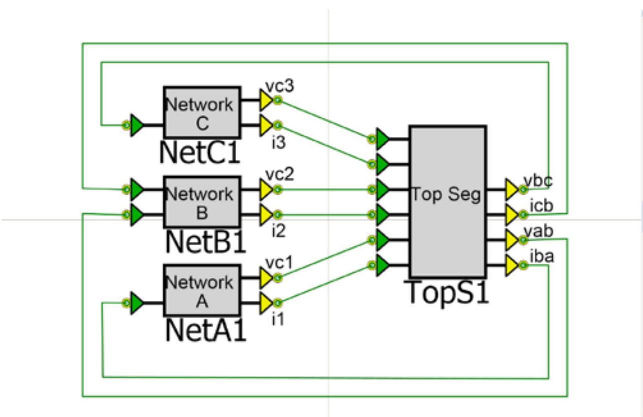

Figure 16 Coupling using ESL entity

An example of graphical output from VTB is shown in Figure 17.

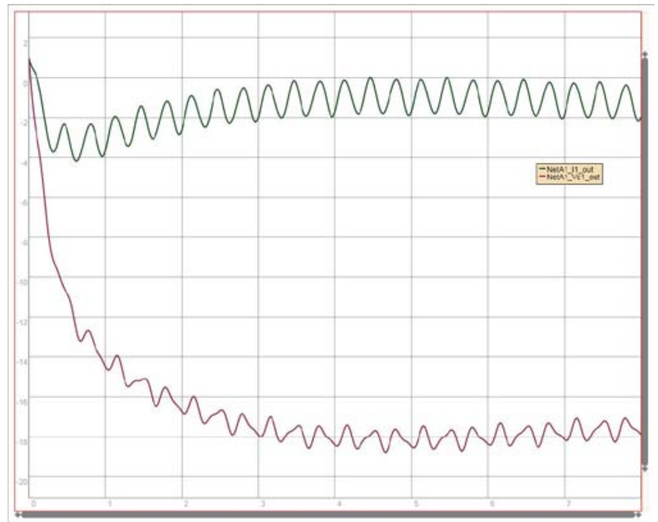

Figure 17 VTB States for Network A

6. DISCUSSION

A simple three-network circuit has been used to show how ESL's segment structures can be used to implement multi-rate simulation programs. A major problem with multi-rate simulations is how to implement the coupling between the networks. Errors are introduced because inputs from neighbouring networks are held constant during the communication period. Two approaches are presented: (1) using simple inductive coupling between the segments; and (2) using the state-variable outputs from each segment to calculate the segment inputs.

Initially, for the purpose of verification, segmented solutions were obtained in which all segments used the same step-lengths. These solutions compared favourably with a reference un-segmented solution. The segmented solution was repeated using appropriate step-lengths for each segment, providing the graphical results shown above. In all cases Euler integration was used. The multiple step-length solutions generally showed good agreement with the reference solution; however some small discrepancies were noted for the i_1 state variable from Network A (see Figure 9). Significantly better agreement was obtained using state-variable coupling approach, however further work is required to analyse the accuracy and stability of this approach.

When the problem was solved under the ESL-VTB environment for the state-variable coupling method, identical results were obtained (Figure 17). As pointed out, a limitation of the version of VTB that was used in these studies (an early release of VTB 2009) is that a common communication rate must be used between all ESL embedded segments. That is easily overcome in a purely ESL solution by calling segments at different rates and will be possible in an expected multi-rate version of VTB 2009.

A further point is that it is possible to invoke ESL *remote* segments from *embedded* segments. The ESL remote segments could be executed on different computers to each other and the VTB host thus implementing a parallel processing environment, as shown in Figure 18. This would be advantageous for very large scale systems. Also, this arrangement works equally well if the remote computers are running Windows or Linux.

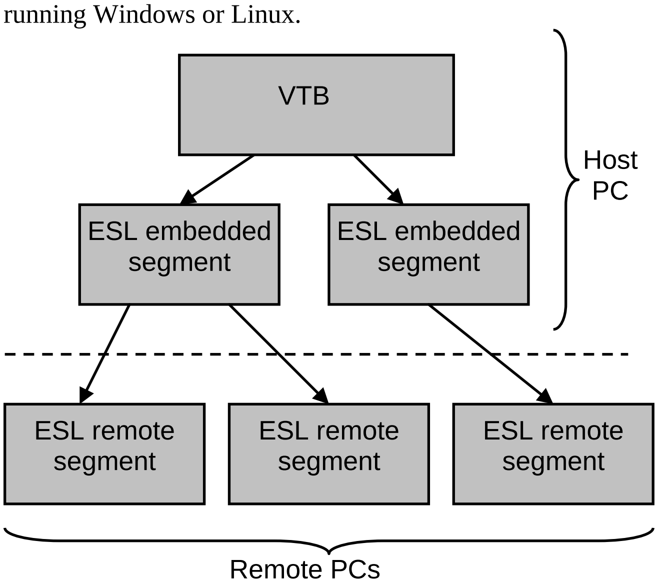

Figure 18 Remote segment architecture

7. CONCLUSION

An approach to implementing multi-rate simulations using the segment feature of ESL has been described and illustrated using a simple three network electrical circuit. A purely ESL solution uses emulated segments to represent the three networks, each using a different step-length. The segments return the network state variables which are used in algebraic coupling equations solved at each communication point at the top model level. A second solution makes use of on-going developments to integrate ESL with VTB and utilises embedded segments to model the networks. Both solutions yield good results compared to an un-segmented solution.

References

[1] Haraldsdottir, A. and R. M. Howe, R.M. 1988. "Multiple Frame Rate Integration." In Proceedings of the AIAA Flight Simulation Technologies Conference, Sept. 7-9, 1988.

[2] Word, D., Zenor, J.J., Bednar, R., Crosbie, R.E. and Hingorani, N.G. 2007. “Multi-Rate Real-Time Simulation Techniques.” In Proceedings of Summer Computer Simulation Conference (SCSC’07), (San Diego, CA, June 15-18). SCS, San Diego, CA, 195-198.

[3] Pearce, J.G. and Crosbie, R.E. 2000. "ESL-ISE - A Simulation Tool Developed for the Space Industry." In Proceedings of the 2000 International Conference on Simulation and Multimedia in Engineering Education, (San Diego, CA, Jan 23-27). SCS, San Diego, CA, 115-120.

[4] Dougal, R.A. 2005. “Design Tools for Electric Ship Systems.” In Proceedings of IEEE Electric Ship Technologies Symposium, (Philadelphia PA, July 25-27). IEEE, 8-11.

[5] Bednar, R. and Crosbie, R.E. 2007. “Stability of Multi-Rate Simulation Algorithms.” In Proceedings of Summer Computer Simulation Conference (SCSC’07), (San Diego, CA, June 15-18). SCS, San Diego, CA, 189-194.

[6] Schutt-Aine, J.E. "Latency insertion method (LIM) for the fast transient simulation of large networks." In IEEE Transactions on Circuits and Systems-I: Fundamental theory and applications, vol. 48, pp. 81-88, January 2001.

[7] Pearce, J.G. 2007. “Simulation advances using the ESL Simulation Language and the Virtual Test Bed.”, In Proceedings of Grand Challenges in Modeling and Simulation Conference (GCMS’08), (Edinburgh, Scotland, UK, June 16-18). SCS, San Diego, CA, 285-290.

Bibliography

John Pearce received his BSc in Electrical Engineering (1970) and PhD in Computer Simulation (1973) from the University of Salford, UK. He held a Research Fellowship for four years in the Department of Electrical Engineering at the University of Salford where he worked on Simulation of Atomic Collision Processes (his PhD topic) and the development of Continuous System Simulation Languages (CSSLs). This was followed by a period of some twenty-six years as a full-time member of academic staff in the same department at Salford University. He continued to develop his interest in System Simulation and CSSLs and, together with the late John Hay and Roy Crosbie, developed the ISIS and ISIM simulation languages. From 1986 he has been associated with ISIM International Simulation (originally a division of Salford University Business Services Limited and from 1992 a limited company). He was appointed a director of ISIM in 1999. During this period he made major contributions to a series of successful research contracts with the European Space Agency (ESA) the culmination of which was the creation of the ESL simulation language. From 2001 he has concentrated on running ISIM and further development of ESL. From 2007 he has been employed as an Independent Contractor to CSU, Chico Research Foundation, California State University, Chico, CA. and has contributed to a series of real-time simulation projects funded by the US Office of Naval Research. This has included providing support for multi-rate simulation, the development of ESL and integration of ESL with the University of South Carolina Virtual Test Bed (VTB) software.