

# A LINEAR-IMPLICIT QUANTIZED DEVS METHOD FOR VERY STIFF ELECTRICAL NETWORKS USING A LATENCY INSERTION METHOD

Joseph M. Hood  
Roger A. Dougal

College of Engineering and Computing  
University of South Carolina  
301 Main Street  
Columbia, SC, USA  
{hood,dougal}@cec.sc.edu

## ABSTRACT

A Linear-Implicit Quantized State System (LIQSS) method is combined with a Latency Insertion Method (LIM) to create a practical and efficient approach for simulating the dynamics of very stiff electrical systems. The LIQSS method is used to asynchronously update each state in the system, where the LIM modeling method is used to decouple the electrical system model at the individual node level making the asynchronous solution possible. The DEVS specification for the QDL method is presented along with the detailed simulation procedure. The combined method is validated by simulating a 40-state network model with a stiffness ratio of  $10^9$ .

**Keywords:** Stiff Systems, QDEVS, QSS, LIM, Circuits

## 1 INTRODUCTION

In order to simulate the full range of dynamic behavior in a typical electrical system using a single dynamical system model, a very high stiffness ratio ( $\lambda_{max}/\lambda_{min}$ ) must be supported by the simulation method. The uniform time-slicing of implicit solutions for modified nodal analysis or state space equations can only simulate such stiff system models by using a very small time step, making simulation at the required time scales impractical or impossible. To overcome the limitations of time-slicing methods for the simulation of very stiff systems, a method is proposed that uses the asynchronous state update features of Quantized Discrete Event Specification and the modeling advantages of Latency Insertion Method. This method is called Quantized DEVS-LIM, or simply *QDL*.

This work builds upon other earlier attempts at using a Quantized DEVS formulation with a LIM modeling approach (Benigni, Brown, Leonard, and Dougal 2015). The key improvements with the approach proposed here are the use of a linear-implicit quantization function allowing the simulation of very stiff systems, and the use of a LIM model that includes dependent sources. Additionally, this work contains the details of the DEVS functions and simulation procedure needed to replicate the method. The source code for this work written in MATLAB Script can be found at the URL <https://github.com/joehood/qdevs>.

In the following sections, the background on the QDEVS-based Linear-Implicit Quantized State System (LIQSS) and the LIM modeling approach will be provided. The DEVS specification for the proposed QDL method will then be given in the form of detailed descriptions of the relevant DEVS functions, along with a description of the full simulation procedure. Finally, the results are presented from two example simulations.

## 2 QUANTIZED STATE SYSTEMS

The Quantized Discrete Event Specification (QDEVS) as described in (Zeigler 1999), provides a formal DEVS specification of quantized, continuous systems. The Quantized State System formulation, based on the QDEVS specification, and described in (Kofman and Junco 2001) provides a framework for applying QDEVS to systems of ODEs. QSS begins with the assumption that a generic state equation system (SES) described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (1)$$

where  $\mathbf{x}$  is the state vector and  $\mathbf{u}$  is the input vector, can be approximated by a *Quantized State System* (QSS) in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)), \quad (2)$$

where  $\mathbf{q}$  is the quantized value *in the neighborhood* of the state vector  $\mathbf{x}$ , whose quantized resolution is defined by the quantum size  $\Delta Q$ . The states are quantized based on their trajectory using various hysteresis quantization functions depending on the specific QSS algorithm used. Importantly, QSS simulations have guaranteed bounded error (Kofman and Junco 2001), and analytically stable systems cannot become numerically unstable when using a fully-coupled QSS algorithm (Cellier and Elmqvist 1993).

Early QSS algorithms, such as those proposed in (Kofman and Junco 2001) and (Kofman 2002), suffer from a severe inefficiency when applied to very stiff problems. Stiff ODEs can cause very high-frequency numerical oscillations due to the interactions between very fast and very slow state transitions (Migoni, Ernesto, and Cellier 2010). To address this, a Linear-Implicit QSS (LIQSS) algorithm is proposed in (Migoni and Kofman 2007), which uses a semi-implicit solution in the quantization function to force steady-state derivatives to zero and prevent troublesome oscillations for stiff systems. We will use the LIQSS algorithm for the development of the QDL method.

## 3 LATENCY INSERTION METHOD

LIQSS requires a system described as a state equation system (SES) of the form  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$  (Cellier, Kofman, Migoni, and Bortolotto 2008). A electrical system in the form of an equivalent circuit model can be converted into a state equation system using the Latency Insertion Method (LIM), while still enforcing energy-conservation laws.

The Latency Inversion Method, as described in (Schutt-Aine 2001), allows an electrical system model to be de-coupled at the node level by exploiting existing latency, and inserting small, fictitious latency at nodes and branches where no significant physical latency exists. The method enforces Kirchoff's Voltage Law (KVL) and Kirchoff's Current Law (KCL) on the network. We are interested in an enhanced LIM formulation that includes full mutual-coupling between states via controlled sources, as well as support for ideal voltage and current sources. These enhancements are critical for the modeling of real-world systems with energy conversion.

DEVS requires an atomic model for each state variable. We define below two atomic model types: the LIM *node* and the LIM *branch* that will be mapped to the standard DEVS specification.

### 3.1 Atomic LIM Components

The LIM node defines a KCL-based ODE for the state of a node voltage, and the LIM branch defines a KVL-based ODE for the current flow between two arbitrary nodes. The generic LIM branch model with dependent sources is shown in Figure 1.

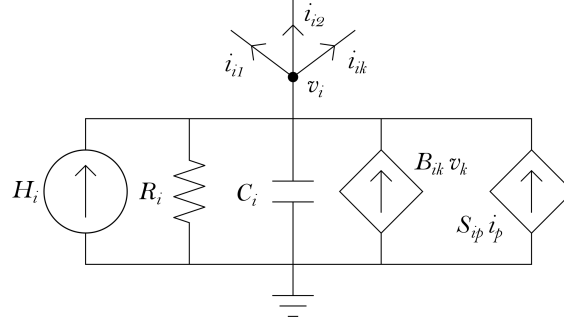


Figure 1: Generic LIM node with dependent sources.

The KCL equation for the the  $i^{th}$  node is

$$C_i \frac{d}{dt} v_i(t) + G_i v_i(t) - H_i(t) - B_{ik} v_k(t) - S_{ip} i_p(t) = \sum_{k=1}^{M_i} i_{ik}(t). \quad (3)$$

The generic LIM branch model with dependent sources is shown in Figure 2.

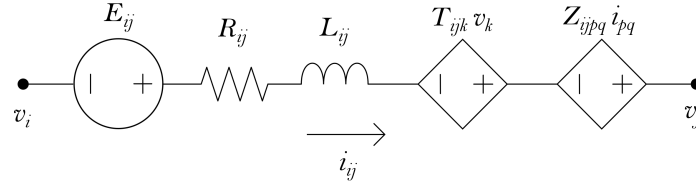


Figure 2: Generic LIM branch with dependent sources.

The KVL equation for a branch from  $i^{th}$  to the  $j^{th}$  node is

$$v_i(t) - v_j(t) = L_{ij} \frac{d}{dt} i_{ij}(t) + R_{ij} i_{ij}(t) - e_{ij}(t) - T_{ijk} v_k(t) - Z_{ijpq} i_{pq}(t). \quad (4)$$

The LIM formulation also allows for ideal, externally controlled voltage node components and current branch components. The method of inclusion of these components is trivial but worth noting. Because the voltages at the branch ports are effectively dc quantities as far as the branch model is concerned during the span of a DEVS time advance period, an ideal source does not require an special consideration by the branch model, and it is treated the same as a latent LIM node voltage value. The same situation exists for ideal current source branches in relation to the branch's terminal nodes.

### 3.2 LIM State Space Model

A LIM system model can be represented by a state space model, with partitions as shown in Equation 5. We are interested in the LIM state equation because various components of the LIM state space model will be used directly in the formulation of the QDL atomic model equations in the following section. Also, simulating the LIM state space system using an implicit numerical ODE solver provides a trusted benchmark solution with which to compare QDL simulation accuracy and computational performance.

The LIM system state space model is

$$\frac{d}{dt} \begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} C^{-1}(B-G) & C^{-1}(S-A) \\ L^{-1}(T-A^T) & L^{-1}(Z-R) \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} + \begin{bmatrix} C^{-1} & 0 \\ 0 & L^{-1} \end{bmatrix} \begin{bmatrix} H \\ E \end{bmatrix}, \quad (5)$$

where  $A$  is the connection incidence matrix that encodes the topology of the network. The elements of  $A$  are determined by

$$A_{i,k} = \begin{cases} 1, & \text{if the } i^{th} \text{ node is connected to the } i^{th} \text{ terminal of the } k^{th} \text{ branch} \\ -1, & \text{if the } i^{th} \text{ node is connected to the } j^{th} \text{ terminal of the } k^{th} \text{ branch} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

## 4 QDL DEVS SPECIFICATION

The DEVS specification of QDL consists of several function definitions corresponding to the formal DEVS-based QSS functions outlined in (Kofman 2004). These are the Derivative Update function ( $f$ ), the Internal State Transition Function ( $\delta_{int}$ ), the External State Transition Function (or  $\delta_{ext}$ ), the Time Advance Function ( $ta$ ), and the Quantization Function ( $Q$ ). These are all fully described below in terms of the Linear-Implicit QSS method and LIM models.

### 4.1 QDL Derivative Calculation

The derivative estimate is then determined from the LIM model for the nodes with

$$d_i = \frac{H_{ii}}{C_{ii}} - \frac{G_{ii}}{C_{ii}} \cdot q_i + \frac{1}{C_{ii}} B_i \cdot q^{node} + \frac{1}{C_{ii}} (S_i - A_i) \cdot q^{branch}, \quad (7)$$

and for the branches with

$$d_k = \frac{E_{kk}}{L_{kk}} - \frac{R_{kk}}{L_{kk}} \cdot q_k + \frac{1}{L_{kk}} Z_k \cdot q^{node} + \frac{1}{L_{kk}} (T_k - A_k^T) \cdot q^{branch}, \quad (8)$$

where  $C_{ii}$ ,  $G_{ii}$ ,  $H_{ii}$ ,  $R_{kk}$ ,  $L_{kk}$ , and  $E_{kk}$ , are quantities from the LIM model at node  $i$  and branch  $k$ .  $B_i$ ,  $S_i$ , and  $A_i$  are quantities from the LIM model at row  $i$ .  $Z_k$ ,  $T_k$ , and  $A_k^T$  quantities from the LIM model at row  $k$ .  $q_i$  is the quantized state at node  $i$ ,  $q_k$  is the quantized state at branch  $k$ ,  $q^{node}$  is the vector of node quantized states, and  $q^{branch}$  is the vector of branch quantized states.

## 4.2 QDL Internal Transition Function

The internal state for each QDL atom is calculated by the internal transition function  $\delta_{int}$ . An internal transition occurs when the simulation time has advanced to the atom's  $t^{next}$  value determined by its time advance function  $ta$ .

In this function, the node voltage states are updated as

$$v_i = v_i^{last} + d_i^{last} \cdot (t - t_i^{last}), \quad (9)$$

and the branch currents are updated as

$$i_k = i_k^{last} + d_k^{last} \cdot (t - t_k^{last}), \quad (10)$$

where  $v_i$  and  $i_k$  are the voltage at node  $i$  and the current at branch  $k$  respectively.

After the states are updated, the  $t^{last}$  values are saved:

$$t_i^{last} = t, \quad t_k^{last} = t. \quad (11)$$

Note that in the case of QDL, the *behavior* of the internal  $\delta_{int}$  and the external  $\delta_{ext}$  transition functions are identical. The difference is in how each is invoked. The internal transition is triggered when the simulation time  $t$  has advanced to the atom's  $t^{next}$  value determined in the  $ta$  function, and the external transition is triggered when one or more connected atoms' quantized states change. Because the behavior of the internal and external transition functions are identical, the confluent transition function  $\delta_{con}$  is not required.

## 4.3 QDL Time Advance Function

The time until the next internal transition is determined from the time advance function  $ta$ .

The time advance calculation for node  $i$  is

$$t_i^{next} = \begin{cases} t_i + (\bar{q}_i - v_i)/d_i, & \text{if } d_i^{last} > 0 \\ t_i + (\underline{q}_i - v_i)/d_i, & \text{if } d_i^{last} < 0, \\ \infty, & \text{otherwise} \end{cases} \quad (12)$$

and for branch  $k$  is

$$t_k^{next} = \begin{cases} t_k + (\bar{q}_k - i_k)/d_k, & \text{if } d_k(t_k^{last}) > 0 \\ t_k + (\underline{q}_k - i_k)/d_k, & \text{if } d_k(t_k^{last}) < 0, \\ \infty, & \text{otherwise} \end{cases} \quad (13)$$

where  $\bar{q}$  and  $\underline{q}$  are the upper and lower quantization limits respectively, and are dynamically updated by the quantization function described in the following section.

#### 4.4 QDL Quantization Function

The quantization function quantizes the internal state after a transition has occurred. The LIQSS uses an advanced hysteresis function that tracks the sign of the derivative, determines when the state is oscillating between two quantized levels ( $\pm\Delta Q$ ), and sets the quantized value such that the derivative is zero, using an approximation of the Jacobian diagonal. Below is the description of the LIQSS atomic DEVS model quantization method which is described in (Migoni and Kofman 2007).

Note that the quantization function is the same for node voltages and branch currents, but only the equations for  $\mathbf{v}$  and  $v_i$  are included below for brevity. The same equations apply to the branch currents by substituting  $\mathbf{i}$  and  $i_k$  in place of  $\mathbf{v}$  and  $v_i$ .

The following definitions will be used in the description of quantization method:

- $\mathbf{v}$  is the set of node voltages,
- $\mathbf{u}$  is the set of external inputs,
- $\mathbf{q}$  is the set of quantized states of all nodes and branches,
- $\mathbf{J}$  is the Jacobian matrix,
- $v_i$  is the value for the  $i^{th}$  voltage at time  $t$ ,
- $\dot{v}_i$  is the time derivative of  $v_i$ ,
- $f_i$  is the time derivative function of  $v_i$ ,
- $\Delta Q_i$  is the quanta value for the  $i^{th}$  node,
- $q_i$  is the quantized value of  $v_i$ ,
- $\underline{q}_i$  is the lower quantum limit of  $v_i$ ,
- $\bar{q}_i$  is the upper quantum limit of  $v_i$ ,
- $\hat{q}_i$  is the quantized value for which  $\dot{v}_i = 0$ , and
- $\tilde{q}_i$  is the approximate quantized value for which  $\dot{v}_i = 0$ .

Note that the dependence of  $v$ ,  $u$ , and  $q$  on the current simulation time  $t$  is implicit, except where the superscript *last* is used to denote quantities at the simulation time  $t^-$ , or the time of the previous quantization update.

$$q_i = \begin{cases} \underline{q}_i, & \text{if } f_i(\mathbf{q}, \mathbf{u}) (\underline{q}_i - v_i) \geq 0 \\ \bar{q}_i, & \text{if } f_i(\mathbf{q}, \mathbf{u}) (\bar{q}_i - v_i) \geq 0 \wedge f_i(\mathbf{q}, \mathbf{u}) (\underline{q}_i - v_i) < 0 \\ \tilde{q}_i, & \text{otherwise} \end{cases} \quad (14)$$

with the upper and lower quantized limits defined as

$$\underline{q}_i = \begin{cases} \underline{q}_i^{last} - \Delta Q_i, & \text{if } v_i - \underline{q}_i^{last} \leq 0 \\ \underline{q}_i^{last} + \Delta Q_i, & \text{if } v_i - \underline{q}_i^{last} \geq 2\Delta Q_i \\ \underline{q}_i^{last}, & \text{otherwise} \end{cases}, \quad (15)$$

$$\bar{q}_i = \underline{q}_i^{last} + 2\Delta Q_i, \text{ and} \quad (16)$$

$$\tilde{q}_i = \begin{cases} \bar{q}_i - (1/A_{ii}) \cdot f_i(\bar{\mathbf{q}}^i), & \text{if } J_{ii} \neq 0 \\ \underline{q}_i^{last}, & \text{otherwise} \end{cases}, \quad (17)$$

where  $\bar{\mathbf{q}}^i$  is equal to  $\mathbf{q}^{last}$  except for the  $i^{th}$  component, where it is equal to  $\bar{q}_i$  and  $J_{ii}$  is the  $ii^{th}$  component of the Jacobian matrix evaluated at  $\bar{\mathbf{q}}^i$ , i.e.,

$$J_{ii} = \left. \frac{\partial f_i}{\partial v_i} \right|_{\bar{\mathbf{q}}^i, \underline{q}_i^{last}}, \quad (18)$$

and therefore, when  $J_{ii} \neq 0$ , setting  $q_i = \tilde{q}_i$  will force  $\dot{v}_i = 0$  in the linear case. The calculation of  $\tilde{q}_i$  is achieved by taking  $\hat{\mathbf{q}}^i$  equal to  $\bar{\mathbf{q}}^i$  except for the  $i^{th}$  component is  $\hat{q}_i$ .

We solve for  $\hat{q}_i$  (the point where  $f_i = 0$ ) as

$$\hat{q}_i = \bar{q}_i - \frac{f_i(\bar{\mathbf{q}}^i, \mathbf{u})}{J_{ii}} + \frac{g(\bar{\mathbf{q}}^i, \mathbf{u}) - q(\hat{\mathbf{q}}^i, \mathbf{u})}{J_{ii}}, \quad (19)$$

where  $g(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, \mathbf{u}) - J_i \cdot \mathbf{x}$ ,  $J_{ii}$  is approximated by

$$J_{ii} \approx \frac{f_i(\bar{\mathbf{q}}^i, \mathbf{u}) - f_i(\underline{\mathbf{q}}^i, \mathbf{u})}{\bar{q}_i - \underline{q}_i}. \quad (20)$$

#### 4.5 QDL Simulation Procedure

The atom models are coupled via the topological connections as encoded in the LIM system port connection incidence matrix  $A$ , as well as the controlled source matrices  $B$ ,  $S$ ,  $T$  and  $Z$ . These coupled models become the Coupled DEVS System as described in (Zeigler 1999) and (Kofman 2004). The DEVS simulation procedure that implements the Coupled DEVS simulation is described in the flow chart in Figure 3.

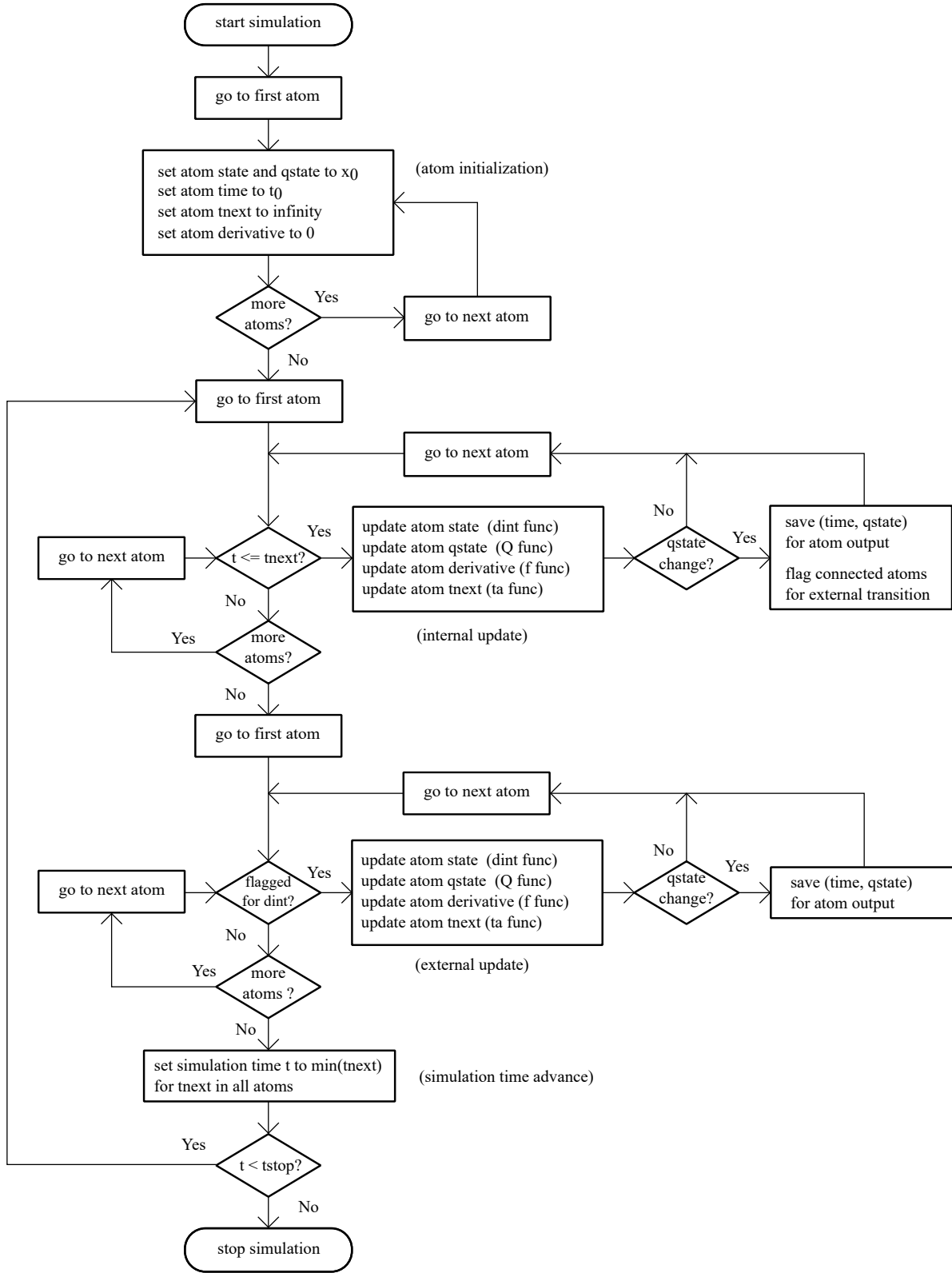


Figure 3: QDL simulation procedure.



## 5 SIMULATION EXAMPLES

We first illustrate the QDL method with a very simple example. A 2<sup>nd</sup> order system consisting of one node and one branch is presented in Figure 4. The simulation results are shown in Figure 5 compared against the implicit state space solution (dashed lines). The QDL results are piece-wise constant values. Note that the quantum size  $\Delta Q$  is different for the node voltage and the branch current. In general,  $\Delta Q$  can be specific to each atom.

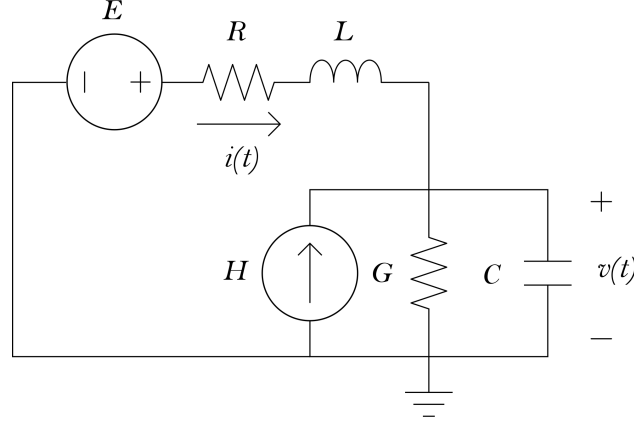


Figure 4: 2<sup>nd</sup> order QDL system.

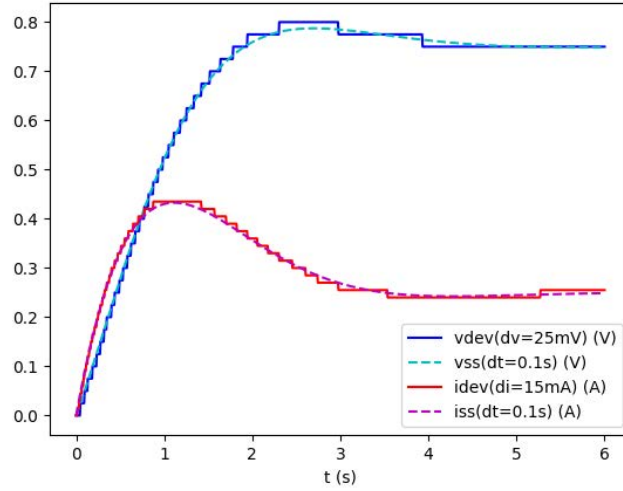


Figure 5: 2<sup>nd</sup> order QDL system simulation results.

### 5.1 Stiff LIM Grid Simulation

In order to test the capabilities of the QDL method with stiff systems, a dense grid was created with a very wide range of time constants. This grid has 16 LIM nodes and 24 LIM branches (see Figure 6). Each quadrant of the grid contains different levels of latency, controlled by setting the capacitance values for the nodes within in the quadrants. The eigenvalues of the system range approximately from  $10^{-3}$  seconds to  $10^6$  seconds, creating a stiffness ratio of approximately  $10^9$ . The voltage at the corner nodes are used as

representative states for analysis of the results of the simulation, and are conveniently named Node 1, Node 2, Node 3, and Node 4, corresponding to their quadrant.

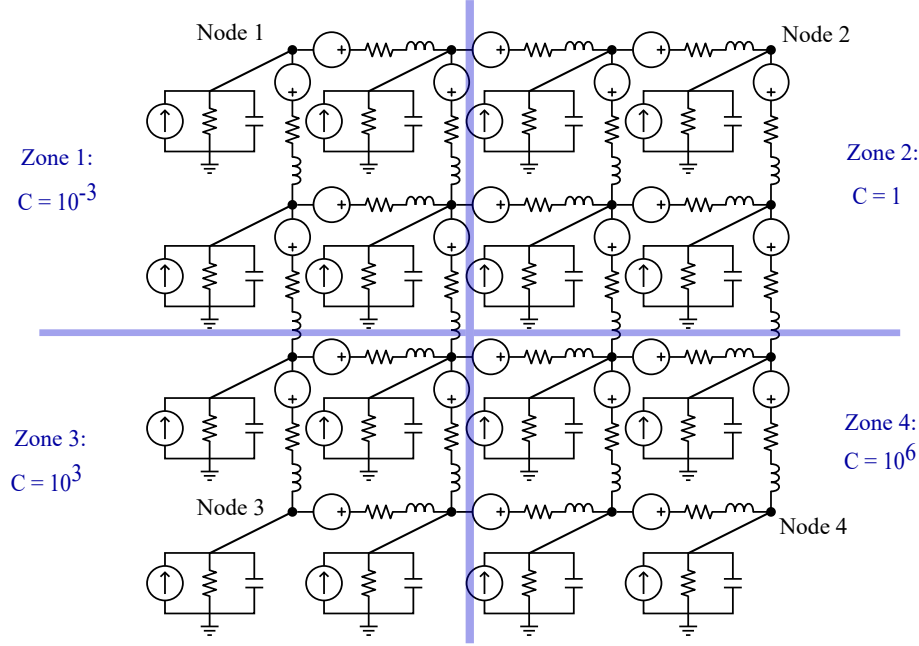


Figure 6: Stiff LIM grid with four latency zones.

The simulation was run for  $10^4$  simulation seconds. The current injection at Node 1 is stepped from 0 to 1A at  $t = 0$ , and from 1A to 10A at  $t = t_{sim}/2$  to provide a perturbation to create a dynamic response. The results are shown in Figure 7. Because the results from a simulation with such a large stiffness ratio are difficult to visualize on one time scale, zoomed plots of the faster transients are included for corner Node 1, Node 2 and Node 3 (see Figure 8). Note that the dynamic response of Node 4 is too slow to warrant a zoomed plot.

Included in each plot are two axes, a left axis for the voltage quantity, and a right axis for the update frequency. The update frequency is the rate at which the asynchronous state updates occur in the simulation for each atom. As expected and desired, the update frequencies are relatively high during the transients, and very low or non-existent during the steady-state portions. Each plot's legend shows the update frequency histogram bin size in seconds. Note that these bin sizes vary with each plot as they were chosen for readability.

An implicit state space solution as a benchmark was impractical to create for this example because of the extreme stiffness and long simulation time of this simulation. The run-time of such a simulation would take many hours or days to simulate on typical desktop hardware using an implicit state space solution with a time step small enough to capture the fast dynamics. For example, a time step of  $\lambda_{min}/10$  would require the solution of a 40-state system for  $10^8$  time steps. The QDL system, however, requires total updates on the order of  $10^5$  for all states combined, and runs in less than 5 mins on a laptop computer as a single-threaded application. Note that this is not an exhaustive performance or accuracy analysis. The quantification of computational efficiency and error will need to be part of future work. Also, because an implicit state space solution is impractical to perform on these types of the extremely stiff systems, a reasonable method of bench-marking the performance and results will have to be determined that does not require running the full simulation.

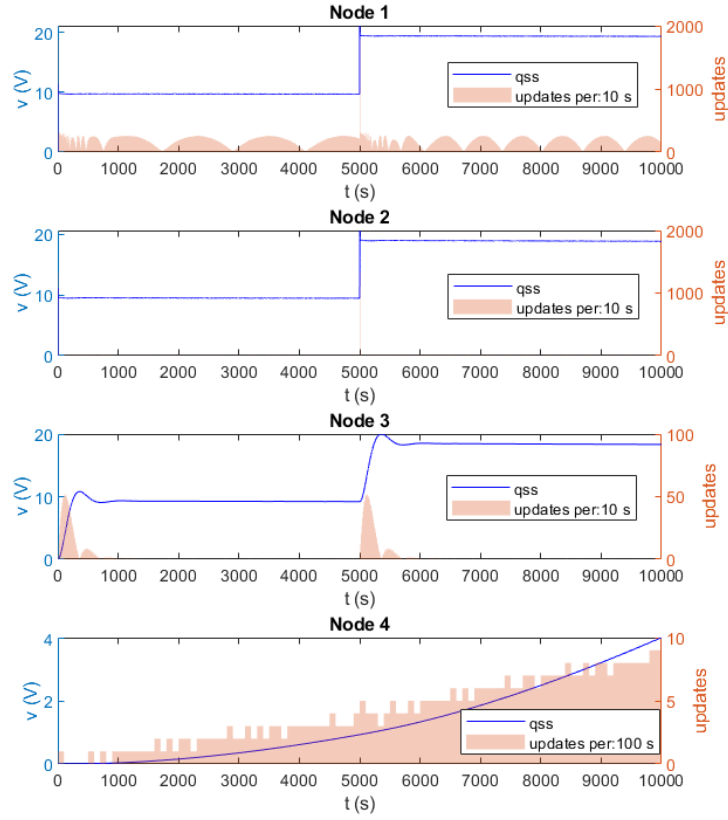


Figure 7: Stiff grid corner node voltage dynamic response.

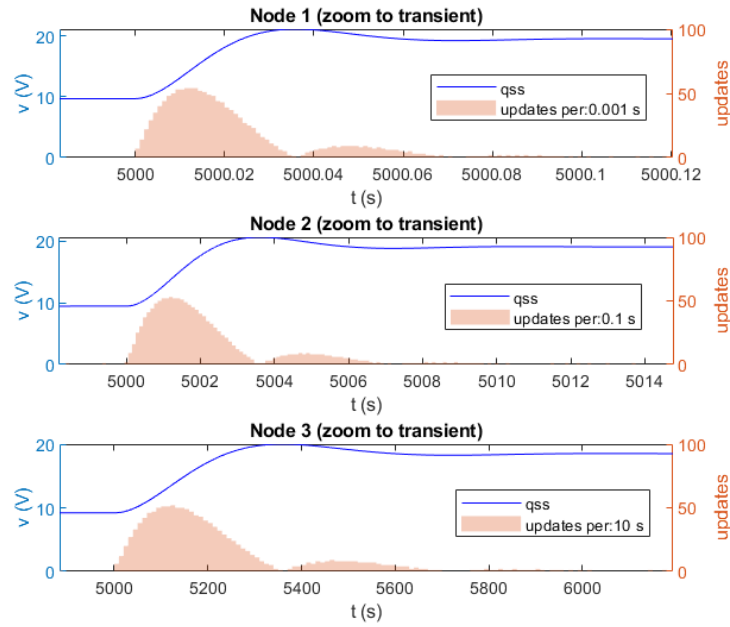


Figure 8: Stiff grid corner node voltage dynamic response (zoom to transient).

## 6 CONCLUSION

A method for simulating the dynamics of very stiff electrical system models was presented, using the QDEVS-based LIQSS formulation and the LIM modeling method. The combined method is called QDL, and is shown to be efficient in simulating the dynamics of very network with a stiffness ratio of  $10^9$  and a simulation time of  $10^4$  seconds. Future work should include the modeling realistic electrical networks that include non-linear electrical machines and power converters with vastly different time constants in a unified simulation model. Detailed comparisons between the QDL method and traditional implicit state space solution should be performed to quantify accuracy, stability and performance.

## REFERENCES

- Benigni, A., M. Brown, R. A. Leonard, and R. A. Dougal. 2015, Nov. "QDEVS natural coupled simulation of a microgrid". In *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 5014–5019.
- Cellier, F. E., and H. Elmqvist. 1993. "Automated formula manipulation supports object-oriented continuous-system modeling". *IEEE Control Systems* vol. 13, pp. 28–38.
- François E. Cellier and Ernesto Kofman and Gustavo Migoni and Mario Bortolotto 2008. "Quantized State System Simulation".
- Kofman, E. 2002. "A Second-Order Approximation for DEVS Simulation of Continuous Systems". *Simulation* vol. 78, pp. 76–89.
- Kofman, E. 2004. "Discrete Event Simulation of Hybrid Systems". *SIAM J. Scientific Computing* vol. 25, pp. 1771–1797.
- Kofman, E., and S. Junco. 2001. "Quantized-state Systems: A DEVS Approach for Continuous System Simulation". *Trans. Soc. Comput. Simul. Int.* vol. 18, pp. 123–132.
- Gustavo Migoni and Ernesto and François E. Cellier 2010. "Quantization–Based New Integration Methods for Stiff ODEs".
- Gustavo Migoni and Ernesto Kofman 2007. "Linearly Implicit Discrete Event Methods for Stiff Ode's".
- Schutt-Aine, J. E. 2001, Jan. "Latency insertion method (LIM) for the fast transient simulation of large networks". *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* vol. 48 (1), pp. 81–89.
- Bernard P. Zeigler 1999. "DEVS Theory of Quantized Systems".

## AUTHOR BIOGRAPHIES

**JOSEPH M. HOOD** is a Ph.D. candidate at the University of South Carolina. He has worked as an engineer for Siemens, developing simulation methods and models for the PSSE power system simulator. His research interests include simulation methods for power systems, advanced electric machine modeling, and DEVS methods applied to continuous systems. His email address is [hood@cec.sc.edu](mailto:hood@cec.sc.edu).

**ROGER A. DOUGAL** received his Ph.D. in electrical engineering from Texas Tech. University, Lubbock, in 1983. He is currently Carolina Distinguished Professor and Chair of Electrical Engineering at the University of South Carolina. His research interests include power electronics, power systems, hybrid power sources, simulation methods, and system design tools. He has organized related conferences such as the Electric Ship Technology Symposium (2013, 2015) and International Conference on DC Microgrids (2017). For more than two decades he has led development of software for analysis, design, and simulation of multidisciplinary power and energy systems. His email address is [dougal@cec.sc.edu](mailto:dougal@cec.sc.edu).