A toolbox for the modelling and simulation of islanded microgrids

Shivam Saxena*, Hany E. Farag and Amir Asif

Department of Electrical Engineering and Computer Science, York University, Toronto, Ontario, M3J 1P3, Canada
E-mail: shivam@cse.yorku.ca
E-mail: hefarag@cse.yorku.ca
E-mail: asif@cse.yorku.ca
*Corresponding Author

Abstract: This paper proposes a simplified toolbox for the modelling and simulation of large microgrids operating in islanded mode. The toolbox integrates an easy to use user interface built in LabVIEW with MATLAB's stiff Ordinary Differential Equations (ODEs) solver to allow a user to model custom microgrid models for test purposes. For smaller models, the user is able to drag and drop network components onto a custom drawer and visualize the network as it is being built. For larger models, a configuration file can be used which contains all the necessary component data for the simulation. This flexibility and ease of use allows for rapid prototyping and testing of microgrid dynamics and steady-state behaviours at different operating conditions.

Keywords: Microgrids; Islanded microgrids; distributed generation; power electronics; inverters; power generation; nonlinear systems; Smartgrid; power sharing; LabVIEW; MATLAB

1 Introduction

The demand for power in North America continues to grow at a steady rate and is projected to increase 28% by the year 2040 (Campillo, 2012). This ever increasing load on the electric power grid is placing a great deal of strain on an ageing and outdated infrastructure. Due to the centralized and hierarchical nature of the electric power grid, a major component fault can cause an outward spiralling effect, which can leave many consumers without power. This was in clear evidence during the Northeast power blackout of 2003 in United States and Canada, in which a combination of multiple transmission lines became faulted and a race condition in the control software of the EPG left upwards of 50 million consumers without power for over 24 hours (Andersson and Donalek, 2005).

As such, the promise of an electrical power network which is decentralized, automated, and distributed has come to realization with the concept of the Smart Grid. The Smart Grid seeks to decentralize the massive power grid into a network of smaller, more manageable subsystems, each of which service a smaller demand for power. These smaller grids, true to their name, are known as Microgrids. Microgrids generate power locally through distributed generation and distribute it among local loads. The term microgrid has been created to be the building block of smart grids (IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems With Electric Power Systems, 2011). A microgrid should be able to operate in two modes of operation: grid-connected or islanded. The successful implementation of the microgrid concept demands a proper definition of the regulations governing its integration in distribution systems. In order to define such regulations, an accurate evaluation of the benefits that microgrids will bring to customers and utilities is needed. Therefore, there is a need for careful consideration of microgrids in the assessment, operation, planning and design aspects of smart grids (Abdelaziz et al., 2013).

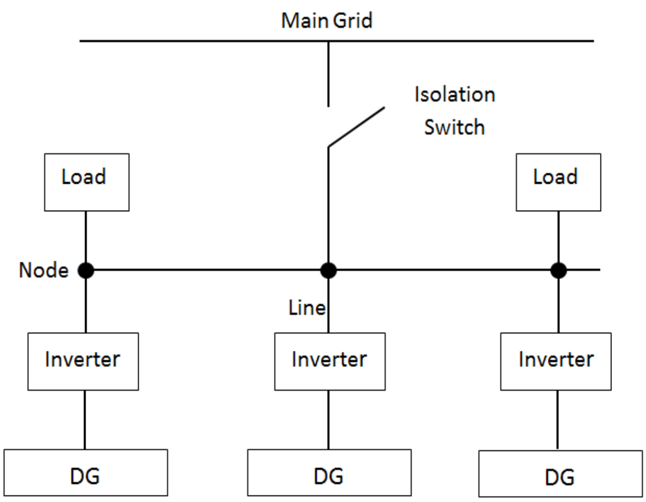
The motivation behind this research is to build a toolbox for the modelling and simulation of inverter based microgrids. Such toolboxes have been well established for conventional power networks (Milano, 2005; Larsson, 2004), however, there exists a lack of a realistic simulation platform used to model large, custom microgrids. Simulink and PSCAD are the most popular tools in which to model microgrids (Panigrahi et al., 2006), however, creating large networks using these tools can be a complex and time consuming process. The modelling of the distributed generator (DG) in particular is difficult as all the internal components and controllers must be modelled. This toolbox uses a simplified model for the DG as a set of five differential equations in which the constant parameters are enabled to be customized by the user. The accuracy of this model has been verified in previous research (Farag et al., 2013). Using an intuitive user interface built in LabVIEW, a user can drag and drop a customized DG and model a microgrid with transmission lines and loads which are similarly customizable. The LabVIEW User Interface is integrated with a MATLAB ODE solver which is then used to solve the custom network modelled as a system of nonlinear differential equations. The traditional approach in modelling power networks and Microgrids is to linearize the nonlinear differential equations, which define the system (Pogaku et al., 2007; Bottrell, 2013). However, electrical power networks are highly nonlinear in nature, and thus this is not an optimal strategy for capturing system dynamics. This toolbox preserves the non-linearity of the ODEs and allows a user to define a custom microgrid for the purposes of simulation.

2 Inverter Based Microgrids

Traditionally, power is generated in large, centralized power plants which are located close to natural resources (coal, natural gas). However, as touched on before, the equipment used to generate electricity is ageing and becoming increasingly more prone to faults. Secondly, and most importantly, emissions from these power plants are extremely harmful to the environment. As such, there is a great deal of research and implementation geared towards harnessing renewable energy sources which are efficient to use and also environmentally friendly. These energy sources could be solar panels, photovoltaic panels, and fuel cells etc. These sources are low cost, low voltage, high reliability, and most importantly, have minimum impact on the environment.

Distributed generation is a concept of aggregating a variety of small energy sources (less than 100 kW) and using the combined power to satisfy larger power demands (Lasseter, 2002). However, the voltages produced by these microsources are DC, and are converted to AC using a voltage source inverter. Thus, an inverter based microgrid is an electrical power network comprised of microsources which produce power locally. The generation units which produce the power are called Distributed Generators (DGs), and they are connected to a bus (node) on the network through transmission lines. The system is finalized by attaching local loads to the nodes (Figure 1).

Figure 1 An example of a test microgrid network



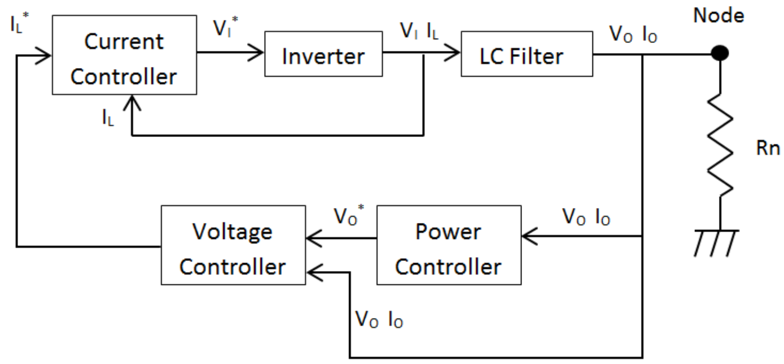
As mentioned before, the microgrid can operate in two modes – Grid Connected and Islanded. When the microgrid is operating in grid connected mode, the overall system dynamics are dominated by the main grid itself (Farag et al., 2013). Traditionally, EPGs try to handle an increase in power demand by turning on smaller, more expensive generators to balance the load. In this case, the microgrid's power supply can be used instead to save cost. In islanded mode, the microgrid services only its local loads and is an individual, self-sufficient network. The system dynamics are governed by all three components: DGs, Lines and Loads. The focus of this paper will be on microgrids operating in islanded mode, and their modelling and state space representation is covered in the next section.

3 Islanded Microgrid Modelling

3.1 Distributed Generators (DG)

The DG is responsible for producing power for the network to use and uses a voltage source inverter to provide AC power. As such, the DG is modelled by a set of non-linear ordinary differential equations that include the DG and inverter (Figure 2).

Figure 2 Block Diagram of the Distributed Generator and Inverter

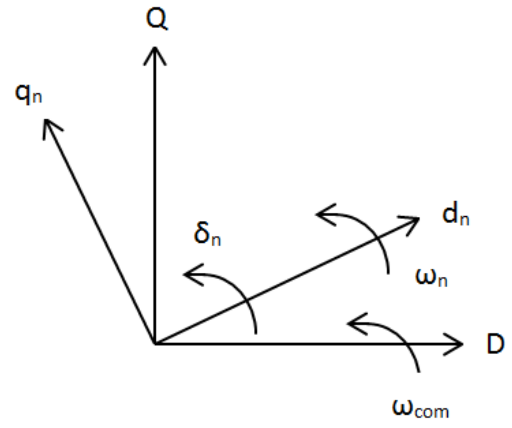


The model for the combined DG and inverter is split into several control loops as indicated in Figure 2. The most important controller is the Power Controller, whose primary objective is to efficiently share the load of the network among the DGs. This is achieved by introducing a droop gain as a system parameter and allows the DGs to share power appropriately in the network. It receives the output voltage and current of the LC Filter (V_o and I_o) of the LC Filter and sets the output magnitude and phase of the voltage (V_o*). The Voltage and Current Controllers, which are designed to reject high frequency disturbances and provide adequate damping for the output LC Filter (Pogaku et al., 2007), are then used to compute the final DC Voltage (V_I*). The Inverter then converts the DC Voltage to AC. The LC Filter, which provides stability to the system and is used to attenuate the frequency ripple of the inverter (Micallef, 2012) then produces the output voltage and the current which feeds into the node. The addition of R_N at the node is to ensure that the numerical solution to the system is well defined. Since the Power Controller and Output LC Filter dominate system dynamics for the DG, the voltage and current controllers can be omitted to simplify the DG model (Farag et al., 2013).

The state variables of interest for the modelling of the DG are: δ - the angle of the DG, P_G - the active power generation of the DG, Q_G – the reactive power generation of the DG, Io_d – Output current in the d^{th} dimension, and Io_q – Output current in the q^{th} dimension. It is important to mention that the state space modelling done in this paper is in the DQ frame (Figure 3). Each DG (say n) is rotating at its own angular frequency, ω_n, on the axis (d_n, q_n). This can be calculated by taking the difference δ_n between the angles associated with the individual DG and the original DQ reference frame on axis (D, Q). As such, an arbitrary DG is appointed as the 'common' DG and is aligned to the DQ axis. The angle for this DG is set to zero, and its rotational frequency is a parameter named ω_{com} which is then used to calculate the angle of all the other DG's. When modelled together in a single network, all DG's (and by extension all other network state variables), must be translated to the DQ frame by using the following transformation:

$$\begin{bmatrix} fD \\ fQ \end{bmatrix} = \begin{bmatrix} \cos(\delta n) & -\sin(\delta n) \\ \sin(\delta n) & \cos(\delta n) \end{bmatrix} \tag{1}$$

Figure 3 Block Diagram of the Distributed Generator and Inverter



The set of nonlinear differential equations describing the modelling of the common DG unit (in this case, DG 1 is assigned as common) is given hereunder in equations (2)-(5).

$$\delta_1{}' = 0 \tag{2}$$

$$P_{G1}{}' = \tfrac{3}{2}\,\omega_{c1}[Vo_1{}^{*}Io_{d1} - Nq_1Q_{G1}Io_{d1}] - \omega_{c1}P_{G1} \tag{3}$$

$$Q_{G1}{}' = -\tfrac{3}{2}\,\omega_{c1}[Vo_1Io_{q1} - Nq_1Q_{G1}Io_{q1}] - \omega_{c1}Q_{G1} \tag{4}$$

$$Io_{d1}{}' = (-Rc_1/Lc_1)\,Io_{d1} + \omega_{com}Io_{q1} + (V_{od}/Lc_1) - (V_{bd}/Lc_1) \tag{5}$$

$$Io_{q1}{}' = (-Rc_1/Lc_1)Io_{q1} - \omega_{com}Io_{d1} + (V_{oq}/Lc_1) - (V_{bq}/Lc_1) \tag{6}$$

where $V_{od} = V_n{}^{*}$ and $V_{oq} = 0$

Similarly, the nonlinear differential equations represent the rest of DG units connected to the microgrid network are given below in terms of unit *n* identified by inserting an additional subscript *n* in each variable.

$$\delta_n{}' = \omega_n{}^{*} - Mp_nP_{Gn} - \omega_{com} \tag{7}$$

$$P_{Gn}{}' = \tfrac{3}{2}\,\omega_{cn}\,[V_n{}^{*}(Io_{qn}\sin(\delta_n) + Io_{dn}\cos(\delta_n))$$

$$- Nq_nQ_{Gn}\,(Io_{qn}\sin(\delta_n) + Io_{dn}\cos(\delta_n))] - \omega_{cn}P_{Gn} \tag{8}$$

$$Q_{Gn}{}' = -\tfrac{3}{2}\,\omega_{cn}\,[V_n{}^{*}(Io_{qn}\cos(\delta_n) - Io_{dn}\sin(\delta_n))$$

$$- Nq_nQ_{Gn}\,(Io_{qn}\cos(\delta_n) - Io_{dn}\sin(\delta_n))] - \omega_{cn}\,Q_{Gn} \tag{9}$$

$$Io_{dn}{}' = (-Rc_n/Lc_n)*Io_{dn} + \omega_{com}*Io_{qn} + (V_{od}/Lc_n) - (V_{bd}/Lc_n) \tag{10}$$

$$Io_{qn}{}' = (-Rc_n/Lc_n)*Io_{qn} - \omega_{com}*Io_{dn} + (V_{oq}/Lc_n) - (V_{bq}/Lc_n) \tag{11}$$

3.2 Lines

Lines are physical transmission lines which connect one node to another. In this paper, transmission lines comprises of a resistor and inductor. The state variables of interest are the currents of the line in the DQ frame (I_{LineD}, I_{LineQ}), and differential equations describing the line connecting two buses are given as follow:

$$I_{LineD,n}' = -(R_{Line,n} * I_{LineD,n}) / L_{Line,n} + \omega_{com} * I_{LineQ,n} + (V_{bjD} / L_{Line,n}) - V_{bkD} / L_{line,n} \quad (12)$$

$$I_{LineQ,n}' = -(R_{Line,n} * I_{LineQ,n}) / L_{Line,n} - \omega_{com} * I_{LineD,n} + (V_{bjD} / L_{Line,n}) - V_{bkQ} / L_{Line,n} \quad (13)$$

where subscripts j,k represent the to and from node, respectively, and subscript n represents the n^{th} Line in the network.

3.3 Loads

In this work, loads are represented by their equivalent admittance. The state variables of interest are the currents of the load in the DQ frame (I_{LoadD}, I_{LoadQ}), and the differential equations are given as:

$$I_{LoadD,n}' = -(R_{Load,n} * I_{LoadD,n}) / L_{Load,n} + \omega_{com} * I_{LoadQ,n} + (V_{bjD} / L_{Load,n}) \quad (14)$$

$$I_{LoadQ,n}' = -(R_{Load,n} * I_{LoadQ,n}) / L_{Load,n} - \omega_{com} * I_{LoadD,n} + (V_{bjQ} / L_{Load,n}) \quad (15)$$

3.4 Nodes

A node, or a bus, is a connection point within the microgrid which connects together a DG, line, and/or load. The voltage at each node is calculated and is needed for both the line and load modelling, where R_n is a virtual resistor. The introduction of the virtual resistor (and its relatively high value), is to ensure that the numerical solution of the system is well grounded and that the node dynamics do not interfere with the overall system dynamics (Pogaku et al., 2007).

$$V_{bDi} = R_N * (\sum I_{oDi} - \sum I_{LoadDi} + \sum I_{LineDi,j}) \quad (16)$$

$$V_{bQi} = R_N * (\sum I_{oQi} - \sum I_{LoadQi} + \sum I_{LineQi,j}) \quad (17)$$

Where the subscripts i,j represent the to and from node, respectively.

3.5 Overall System Model

The network, or overall system model, can then be modelled by combining the set of differential equations of all DGs, Lines, and Loads into an overall state vector. The state variables for all the DGs are modelled first, followed by the Lines, followed by the Loads. Although the order can be defined in any such way for the MATLAB ODE solver, this order is preferred when it comes to creating an index map of the state vector. As can be seen in equation (17), there are 5 state variables for each DG unit, 2 state variables for each load, and 2 state variables for each load.

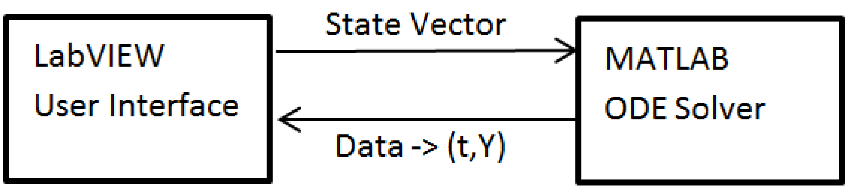
$$X = [\delta_n, P_{Gn}, Q_{Gn}, Io_{dn}, Io_{qn}, I_{LineD,n}, I_{LineQ,n}, I_{LoadD,n}, I_{LoadQ,n}] \tag{18}$$

4 Software Design of Toolbox

4.1 Framework

The objective of the toolbox is to provide an easy to use interface in which the user can solve an *n* node, *n* DG, and *n* Load microgrid network operating in islanded mode. As shown in Figure 4, the software consists of two platforms – LabVIEW and MATLAB. The general framework is written in LabVIEW which includes the user interface, main state machine, and main data storage objects. MATLAB is then called from within the LabVIEW framework to solve the set of differential equations generated by the custom network.

Figure 4 LabVIEW and MATLAB data communication



4.2 LabVIEW

LabVIEW is a graphical programming language and system design platform which is widely used in both academic and research institutions. LabVIEW is chosen for writing the general framework because it is naturally inclined towards a multithreaded style of programming. Multiple threads can be statically or dynamically created and interprocess communication is easily facilitated through built in queues and notifiers. LabVIEW also supports object oriented style programming. Since the toolbox is modelled as a system of objects, and in the future has the potential for the aggregation of more network components, object oriented programming is an important consideration when it comes to the maintenance and expansion of the code in the future. Most importantly, LabVIEW offers an excellent interface to facilitate communication with MATLAB.

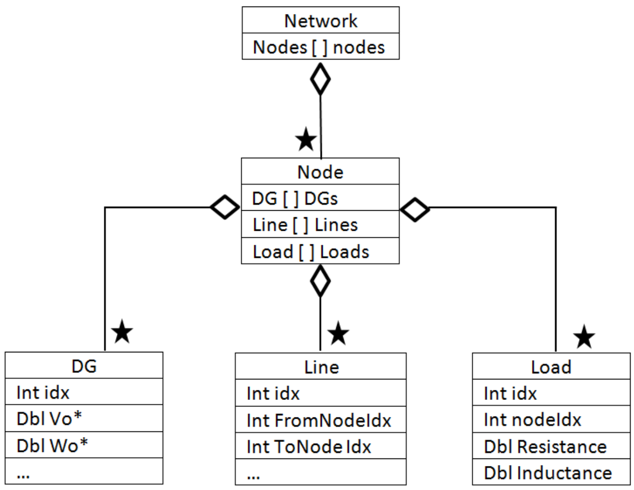
4.3 MATLAB ODE Solver

When the terms of a differential equation cause rapid variation in the solution, the differential equation is classified as a stiff differential equation (Schadenko, 2003). MATLAB specializes in solving stiff, nonlinear differential equations, and provides a set of special solvers for this purpose. The ODE solver *ode15s* is a variable order solver which uses two methods to integrate a system of differential equations: Numerical Differentiation Formulas (NDFs) or Backward Differentiation Formulas (BDFs) (Shampine and Reichelt, 1997). The *ode15s* solver is also a variable step solver, and will attempt to decrease the step size when rapid variations occur to capture dynamics as accurately as possible. The solver also provides parameters for defining relative and absolute tolerances, which allow the user to specify constraints on the eventual solution of the system of differential equations.

4.4 Object Model

Central to the idea of the toolbox is the idea of a network. A single network is custom created by the user and represents all the components that make up the network: DGs, Lines, Loads, and Nodes. An object diagram of the software model is shown in Figure 5. As can be seen in the figure, it follows that a network consists of many nodes, while the nodes themselves can have many DG's, Lines, and/or Loads.

Figure 5 Object Model of toolbox



4.5 System Architecture

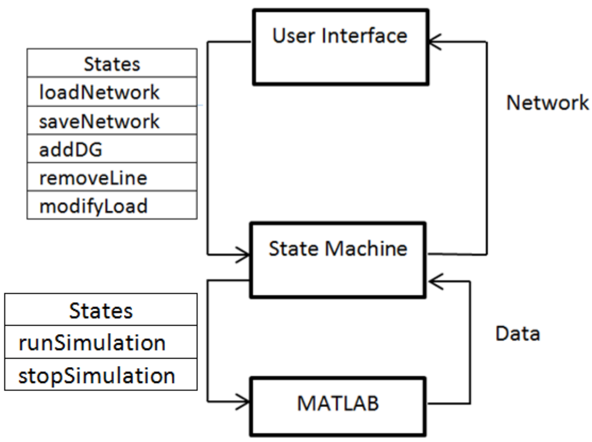
Figure 6 shows the system architecture of the proposed toolbox. As shown in the figure, the system architecture is divided into three separate threads – the User Interface (UI) Thread, the Main State Machine Thread, and the MATLAB ODE thread.

The User Interface Thread is responsible for capturing all user events and forwarding relevant data to the main state machine. The user uses the user interface to add/modify/remove nodes, DGs, Lines, and Loads. The User Interface thread captures the specific option, services the user request, and forwards the updated data to the main state machine where the network data is stored.

The Main State Machine Thread facilitates the program state and holds the latest copy of the Network data. It receives state change instructions from the User Interface Thread (state could go from idle, to running a simulation, to exiting) and also updates the network according to the user's changes. It also sends instructions and data to the MATLAB Loop when the simulation is ready to be run and receives the data back when the simulation is complete. It is implemented as a queued state machine whose default state is simply idle. The UI thread interrupts the main state machine whenever the user engages with the program to make a request.

The MATLAB ODE thread is used to execute the time domain simulation of the network and return the results back to the Main State Machine

Figure 6 System Architecture of the toolbox



Interprocess communication is managed through the use of single element queues. Once instantiated, a reference (pointer to the memory location) to the queue can be called from anywhere in the program to either enqueue or dequeue an element in the queue. Two queues are used for each communication link between thread (UI Thread to State Machine, State Machine to MATLAB) to facilitate a send/receive interface. Screenshots of the user interface can be seen in Figures 7 and 8.

Figure 7 Main UI of the toolbox, where user can add/remove/modify any network component

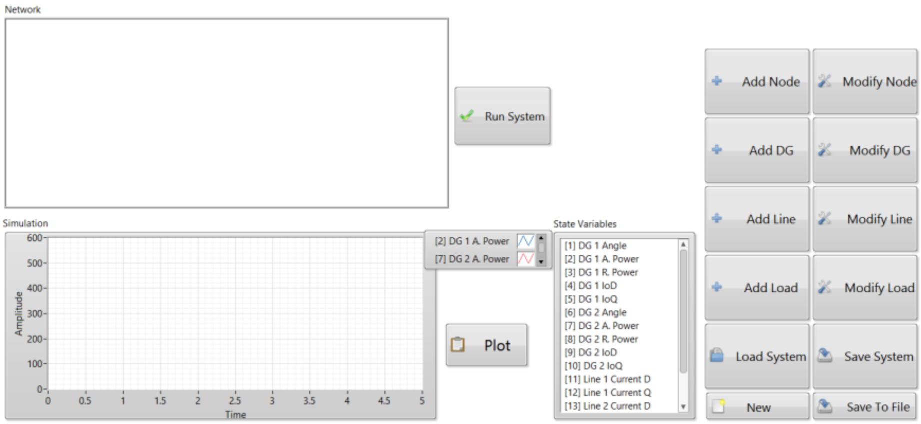
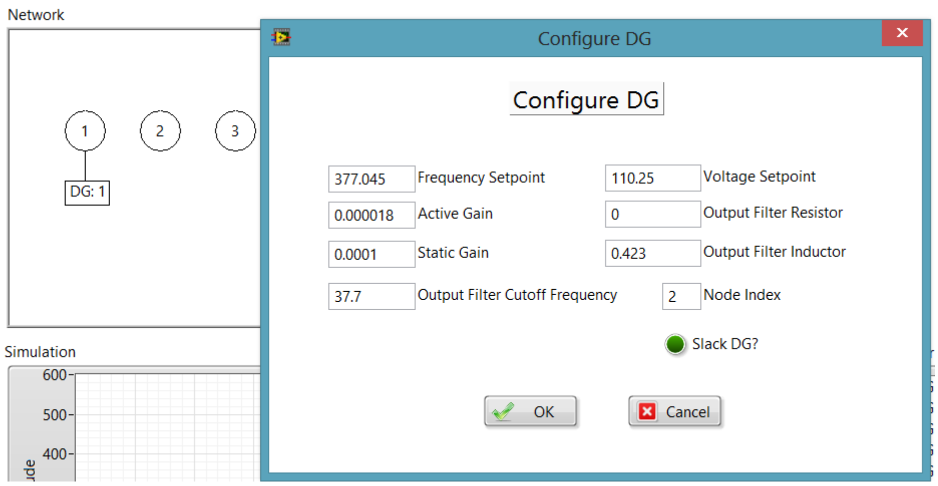


Figure 8 The user configures a custom DG and can visualize the addition to the network



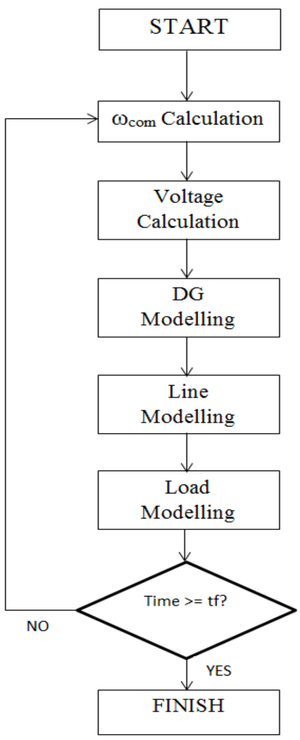
4.6 Index Map

Since the toolbox should be able to solve *any* network, the state vector fed to the MATLAB ODE solver must be dynamically created depending on the network the user has made. The ODE solver must also then be able to execute the correct equation depending on the state variables in the vector. This requires the use of an index map which can be used to hold information about each state variable. A simple object is made which contains the type of component (DG, Line, Load), it's component index (there could be multiple DG's, Lines, Loads), and the relevant state variable for that component (angle, power, current etc.). When an equation needs a particular state variable, it can then find the correct index of where it resides in the state vector by searching through the index map by component index and state variable.

4.7 Simulation Algorithm

Figure 9 shows a flowchart for the proposed simulation algorithm of the custom network generated by the LabVIEW framework. As can be seen in the figure, first: ω_{com} is calculated by using the angle equation for the common DG. Since the angle of this DG is always set to zero, ω_{com} can be solved for by using the state values from the previous time iteration. Second, voltage calculation is done for every node in the system by applying KCL at each node. The currents from the DG and To Node are considered positive, while the currents from the From Bus and Load are considered negative. Third, the equations for the DGs, Lines, and Loads are solved simultaneously depending on the custom network. For this, the index map is traversed one by one until all the state variables have been solved. Fourth, the ODE solver checks if the time step has surpassed the simulation time set by the user. If it has, the simulation is stopped and the data is returned to the user. If not, the simulation increments to the next variable time step.

Figure 9 Flowchart of the proposed simulation algorithm

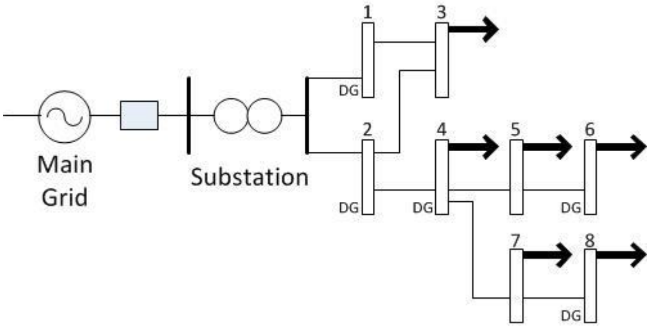


5 Simulation Results

5.1 Case Study – 8 Node/Bus Model

The toolbox was tested on a 8 bus, 5 DG microgrid operating at 110V and 60Hz frequency as shown in Figure 10.

Figure 10 8 Bus Microgrid



Simulation and network parameters are included in Table 1. The experiment involved setting the droop gains of each DG to the same value so that the real power can be shared equally. Due to line impedance mismatches, the voltage at the nodes of the bus are not equal, and as such equal reactive power sharing cannot be achieved since the bus voltages affected by the droop gains settle at different values (Micallef, 2012). This does not affect equal real power sharing since the frequency of the microgrid is set constant by the operator. The microgrid is also set configured to be tested from a blank start, which indicates that all state variables are set to zero.

As can be seen from Figure 11, the power generated by all the DGs converge to the same value, which indicates that the DGs are indeed sharing the responsibility of the 5 attached loads equally. The reactive power output of the DGs is also included for reference in Figure 12. In the second trial, the droop gains for the first two DGs are set to 25% more capacity than the second three DGs. This scenario ensures that the first two DGs, which are presumed closer to the substation, will take more of the load than will the other DGs. As can be seen in Figure 13, the power supplied by the first two DGs are equal, and are significantly higher than the power supplied by the last three DGs. To test the accuracy and stability of the system, active and reactive power losses were calculated at each node by using the following formulas:

$$\text{Active Power Loss} = I^2R \quad (18)$$
$$\text{Reactive Power Loss} = I^2X \quad (19)$$

The steady state value of all the currents flowing into the nodes (Io_d,Io_q,I_{LineD}, I_{LineQ}, I_{LoadD}, I_{LoadQ}) was summed together and multiplied by the resistance and inductance at that particular node to find the active and reactive power loss. It was found that the calculated value for active and reactive power loss was approximately zero.

Considering the negligible resistance of the lines and high virtual resistor value at the nodes, the power loss of the network is indeed expected to be close to zero.

Figure 11 Active Power Generation from the 5 DGs converging to the same value

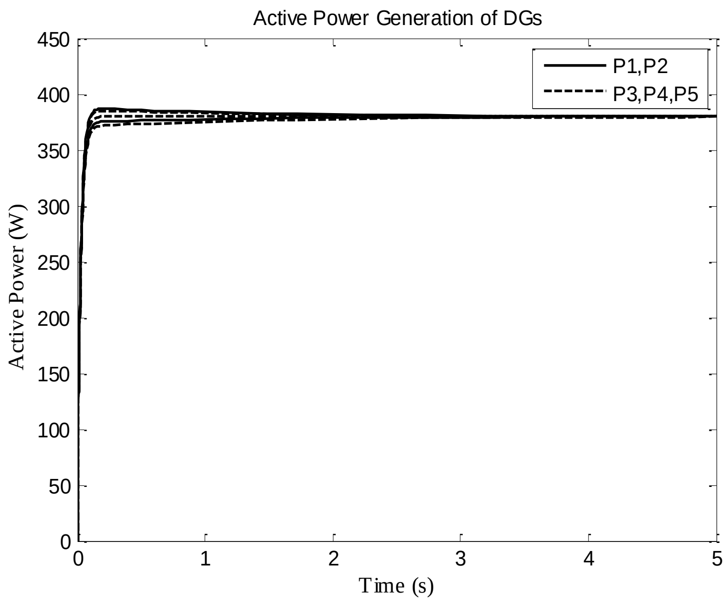


Figure 12 Reactive Power Generation from the 5 DGs

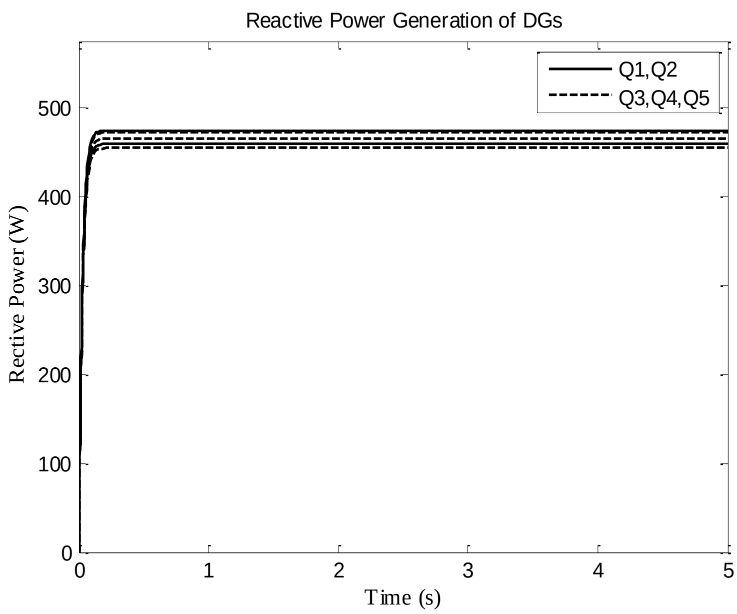
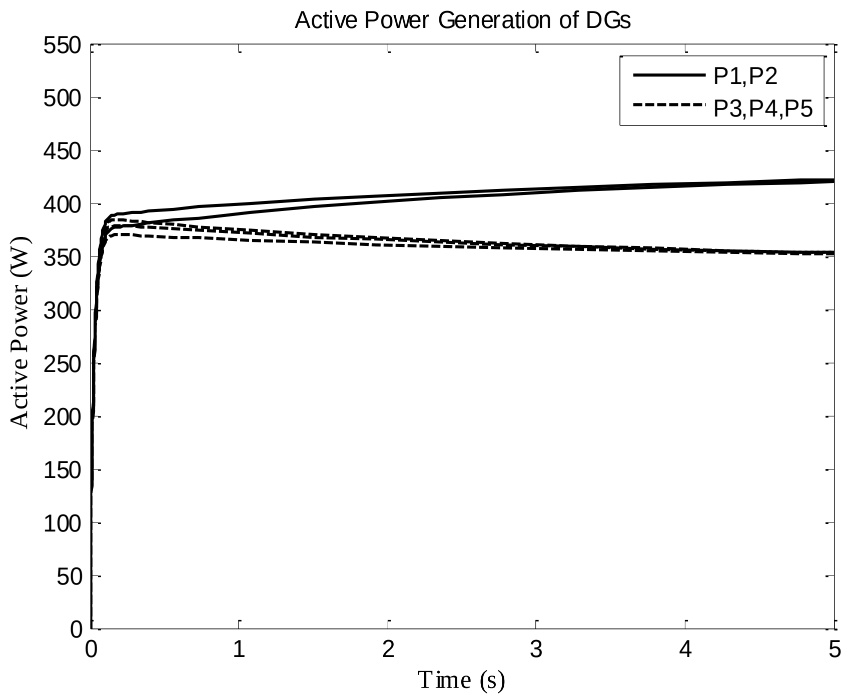


Figure 13 Active Power sharing with unequal droop settings



6 Future Work

Future steps to extend the simulations of the toolbox include: Modelling the omitted current and voltage controllers, modelling several varieties of loads instead of a simple RL load, and performing state estimation within the microgrid. State estimation in particular is an important research aspect within electrical power grids since sampling is only done approximately every 10 minutes (Blood, 2011).. Modelling larger networks, such as the well-known 69 bus microgrid system, is also a priority. It is noted that the drawing the 69 bus system using the toolbox can be a tedious task. As such, a spreadsheet file with relevant data regarding the DGs, Lines, Loads, and nodes is also accepted within the toolbox which will populate the Network object and run the simulation exactly as if the user had drawn the network.

6 Conclusion

The objective of this paper is to present a dynamic toolbox for the modelling and simulation of islanded microgrids. A user is able to create custom networks using a user friendly user interface as well as to input larger networks to the system by file. This toolbox therefore does not require the user to have any programming experience since the components are modelled internally and the state matrix is dynamically created based on the custom network. Furthermore, the intuitive user interface allows the user to quickly design, develop, and prototype a variety of custom microgrids, allowing each one to be saved to file and loaded back at a later time. Finally, the non-linearity of the system of differential equations is preserved, thus providing a more accurate reflection of the dynamics of an electrical power system. An eight bus test model is also developed and droop gain tests are done in order to ensure that the model is in working order. This toolbox has also been tested on a variety of network configurations, including three, five, and six bus models.

Appendix

Table 1 Constants for 8 Bus Model

Parameter	Value	Units
$\omega_1^*,\omega_2^*,\omega_3^*,\omega_4^*,\omega_5^*$	377.045	rad/s
Mp_1,Mp_2,Mp_3,Mp_4,Mp_5	1.8 e^{-5}	-
Nq_1,Nq_2,Nq_3,Nq_4,Nq_5	1.0 e^{-4}	-
$\omega_{c1},\omega_{c2},\omega_{c3},\omega_{c4},\omega_{c5}$	37.7	-
$V_{o1}^*,V_{o2}^*,V_{o3}^*,V_{o4}^*,V_{o5}^*$	110.25	V
$Rc_1^*,Rc_2^*,Rc_3^*,Rc_4^*,Rc_5^*$	0	ohms
$Lc_1^*,Lc_2^*,Lc_3^*,Lc_4^*,Lc_5^*$	0.401,0.423,0.423,0.423,0.423	ohms
$R_{Line,1},R_{Line,2},R_{Line,3}$ $R_{Line,4},R_{Line,5}$	0	ohms
$L_{Line,1},L_{Line,2},L_{Line,3}$ $L_{Line,4},L_{Line,5}$	0.0226,0.0339, 0.0226, 0.0226,0.0226	ohms
$R_{Load,1}$, $R_{Load,2}$, $R_{Load,3}$, $R_{Load,4}$, $R_{Load,5}$	13.104, 13.104, 26.208, 39.312, 39.312, 52.416	ohms
$L_{Load,1}$, $L_{Load,2}$, $L_{Load,3}$, $L_{Load,4}$, $L_{Load,5}$	15.76, 15.76, 31.52, 47.28, 47.28, 63.04	ohms

References

Abdelaziz, M et al., (2013) 'A Novel and Generalized Three-Phase Power Flow Algorithm for Islanded Microgrids Using a Newton Trust Region Method' *IEEE Transactions on Power Systems*, Vol.20, No. 4, pp. 1716-1727

Andersson, G and Donalek, P. (2005) 'Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance', *IEEE Transactions on Power Systems,* Vol. 20 No.4, pp. 1922-1928

Blood, E. (2011) *From Static to Dynamic Electric Power Network State Estimation: the Role of Bus Component Dynamics*. PhD thesis, Carmegie Melon University, Pittsburgh, USA.

Bottrell, N et al., (2013) 'Dynamic Stability of a Microgrid with an Active Load', *IEEE Transactions on Power Electronics,* Vol. 28 No.11, pp. 5107-5119

Campillo, J et al., (2012), 'Electricity Demand Impact from Increased use of Ground Sourced Heat Pumps' in *Innovative Smart Grid Technologies (ISGT Europe): 2012 3rd IEEE PES International Conference and Exhibition*, Berlin, Germany, pp. 14-17.

Farag, H et al., (2013) 'Voltage and Reactive Power Impacts on Successful Operation of Islanded Microgrids', *IEEE Transactions on Power Systems,* Vol. 28 No.2, pp. 1716-1727

Larsson, M (2004) 'ObjectStab – an educational tool for power system stability', *IEEE Transactions on Power Systems,* Vol. 19 No.1, pp. 56-63

IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems With Electric Power Systems (2011), IEEE Std. 1547.4

Lasseter, R. (2002) 'Microgrids', *Proc. Power Eng. Soc. Winter Meeting*, vol. 1, pp. 305-308

Micallef, A. (2012), 'Secondary control for reactive power sharing in droop-control islanded microgrid', in *IEEE International Symposium on Industrial Electronics*, Hangzhou, China, pp. 1627-1633.

Milano, F (2005) 'An Open Source Power System Analysis Toolbox', *IEEE Transactions on Power Systems,* Vol. 20 No.3, pp. 1199-1206

Pogaku, N et al., (2007) 'Modeling, Analysis, and Testing of Autonomous Operation of an Inverter-Based Microgrid', *IEEE Transactions on Power Electronics,* Vol. 22 No.2, pp. 613-625

Schadenko, S. (2003), 'Solution methods review of the stiff differential equations systems', in *Modern Techniques and Technologies, 2003. MTT 2003. Proceedings of the 9th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists*, Tomsk, Russia, pp. 66-68.

Shampine, L and Reichelt, M. (1997) 'The MATLAB ODE Suite', *SIAM Journal on Scientific Computing*, Vol. 18 pp. 1-22