

University of California, Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences
Spring 2019

EE 290O ★ FINAL REPORT

Curi et al. — Control of Low-Inertia Power Grids: A Model Reduction Approach

Professor : Duncan S. Callaway
Students : Victoria Cheng
Rodrigo Henriquez-Auba
Jonathan Lee
Keith Moffat
Date : 05/14/2019

1 Introduction

This report presents our findings on the paper “Control of Low-Inertia Power Grids: A Model Reduction Approach” by Sebastian Curi, Dominic Groß and Florian Dörfler, presented on 2017 IEEE 56th Annual Conference on Decision and Control (CDC).

The paper introduces a dq modeling framework for both devices and network. The main results of the paper are:

- A model order reduction for electrical differential equations based on Tikhonov’s Theorem.
- A parallel between the voltage on the DC side of converters v_{dc} with the frequency of the synchronous generators ω .
- A decentralized nonlinear droop control to stabilize frequency. A key component is that the proposed controller is not dependent on measurements of frequency for the converters.

The paper introduces the modelling for each device in an absolute reference frame that is assumed to be known for each device. It uses a 2 states generator model, plus an electromotive force behind an impedance. The entire model uses currents in the dq framework as the states of the AC system. Converter models are based in an average switching model, while loads are modeled as constant RL impedances. No control is considered directly in the models, that are general enough to consider classic or novel techniques if it’s required. Though not a strict novelty of the paper, they use a precise and formal notation for time-domain modelling of multi-source power networks with linear network elements.

1.1 Project objectives

In power systems one of the key elements of any model is for it to be actually possible to simulate study cases using that model. Curi’s paper is framed mostly as a theoretical paper, so a first step for us was to actually simulate in a tool the proposed model.

A second objective was to analyze the model order reduction proposed in the paper. That is, for simulation purposes, does it make a difference considering the full order model against the reduced model? What about the simulation times?

Finally, a third objective was to implement standard controllers in the proposed model. That is, setting a simplified AVR and droop control for generators and droop controllers for DC/AC converters.

1.2 Organization

The report is organized as follows. Section 2 introduces the model proposed by the authors, summarizing their key components, devices and equations. Section 3 introduces the framework used to simulate the proposed model in Matlab. Section 4 discusses some results that the authors’ proposed and describes our extensions. Section 5 presents the simulations performed in

Matlab to analyze the behaviour of the system under disturbances. Finally, Section 6 concludes the report.

2 Modelling framework

The model proposed works in a dq framework, since it is assumed that the electrical components of each device have identical values for each phase and that all three-phase signals are balanced. That is a three-phase voltage (or current), $v_{abc} \in \mathbb{R}^3$, can be written in (α, β) coordinates via the Clarke transformation:

$$v_{\alpha\beta} = T_{abc \rightarrow \alpha\beta} v_{abc}, \quad T_{abc \rightarrow \alpha\beta} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$$

And hence a voltage can be written in a rotating frame at nominal frequency $\theta_r = \omega_0 t$, where $\omega_0 = 1$ pu ($2\pi \cdot 60$ rad/s), using the Park transformation:

$$v_{dq} = R(\theta_r) v_{\alpha\beta} = R(\theta_r) T_{abc \rightarrow \alpha\beta} v_{abc}, \quad R(\theta_r) = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix}$$

All parameters in the following sections are written in a system per unit basis.

2.1 Notation

We define \mathbb{R} as the set of real numbers, $\mathbb{R}_{\geq 0}$ as the set of non-negative real numbers and \mathbb{S}^1 the set of the unitary circle. The parameter $j \in \mathbb{R}^{2 \times 2}$ plays the same role as in the imaginary number, defined as:

$$j = R(\pi/2) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$\mathbb{1}_{2 \times 2}$ is the 2×2 identity matrix and $r(\theta)$ denote the position in the unit circle with angle θ :

$$r(\theta) := \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

2.2 Sets, states and parameters

The power system considered has the following elements:

- n_v number of AC buses
- n_t number of transmission lines or transformers (series element that connect two nodes)
- n_g number of synchronous generators
- n_c number of DC/AC converters
- n_l number of loads.

2.2.1 Buses

Each bus k has the following components:

Symbol	Type	Description	Info.
v_k	State	AC voltage in pu in the dq framework	$v_k \in \mathbb{R}^2$
$G_{v,k}$	Parameter	Shunt conductance in pu of bus k	$G_k = g_k \mathbb{1}_{2 \times 2}$
$C_{v,k}$	Parameter	Shunt capacitance in pu of bus k	$C_k = c_k \mathbb{1}_{2 \times 2}$
$Y_{v,k}$	Parameter	Shunt admittance in pu of bus k	$Y_k = G_k + j\omega_0 C_k$

Table 1: States and parameters of each bus k .

Then, for each bus the model requires 2 states, the AC voltage in the d component and the AC voltage in the q component. Thus, buses contribute with $2n_v$ states to the model.

2.2.2 Lines

Each line k has the following components:

Symbol	Type	Description	Info.
$i_{t,k}$	State	AC current across the line in pu in the dq framework	$i_{t,k} \in \mathbb{R}^2$
$R_{t,k}$	Parameter	Series resistance in pu of line k	$R_{t,k} = r_{t,k} \mathbb{1}_{2 \times 2}$
$L_{t,k}$	Parameter	Series inductance in pu of line k	$L_{t,k} = \ell_{t,k} \mathbb{1}_{2 \times 2}$
$Z_{t,k}$	Parameter	Series impedance in pu of line k	$Z_{t,k} = R_{t,k} + j\omega_0 L_{t,k}$

Table 2: States and parameters of each line k .

Then, for each line the model requires 2 states, the AC current across the line in the d component and the AC current in the q component. Thus, lines contribute with $2n_t$ states to the model.

In addition, between buses and lines is defined the topology of the network, described by the oriented incidence matrix \mathcal{E} . This matrix has a size of $n_v \times n_t$. Each row represent a particular bus, while each column represent a particular line. The matrix is then defined by:

$$\mathcal{E} = [\mathcal{E}_{t,1} \ \dots \ \mathcal{E}_{t,m} \ \dots \ \mathcal{E}_{t,n_t}]$$

where each column m : $\mathcal{E}_{t,m}$, defines which buses are connected to line m . The sending end is completed with -1 , while the receiving end is completed with a 1. All the other elements are zero.

2.2.3 Synchronous Machines

Each synchronous machine k has the following components:

Symbol	Type	Description	Info.
$\theta_{g,k}$	State	Rotor angle relative to an absolute reference frame θ_r	$\theta_{g,k} \in \mathbb{S}^1$
$\omega_{g,k}$	State	Absolute angular rotor velocity in pu	$\omega_{g,k} \in \mathbb{R}$
$i_{g,k}$	State	Stator AC current flowing in, in pu in dq framework	$i_{g,k} \in \mathbb{R}^2$
$i_{f,k}$	Input	Rotor DC field current in pu	$i_{f,k} \in \mathbb{R}$
$\tau_{m,k}$	Input	Mechanical torque applied to the rotor, in pu	$\tau_{m,k} \in \mathbb{R}$
$R_{g,k}$	Parameter	Stator series resistance in pu	$R_{g,k} = r_{g,k} \mathbb{1}_{2 \times 2}$
$L_{g,k}$	Parameter	Stator series inductance in pu	$L_{g,k} = \ell_{g,k} \mathbb{1}_{2 \times 2}$
$Z_{g,k}$	Parameter	Stator series impedance in pu	$Z_{g,k} = R_{g,k} + j\omega_0 L_{g,k}$
$\ell_{m,k}$	Parameter	Mutual inductance between rotor and stator in pu	$\ell_{m,k} \in \mathbb{R}_{\geq 0}$
M_k	Parameter	Inertia constant of generator k in MWs/MVA	$M_k \in \mathbb{R}_{\geq 0}$

Table 3: States, inputs and parameters of each generator k .

Each generator k is also described with an indicator vector $I_{g,k} \in \{1, 0\}^{n_v}$ that has a 1 on the bus that is located and 0 otherwise.

For each generator, the model requires 4 states, the rotor angle, the angular rotor velocity, the stator AC current, both in the d and q components. Thus, generators contribute with $4n_g$ states to the model. In addition, each generator adds two inputs, the mechanical torque and the field current.

2.2.4 3-Phase DC/AC Converter

Each DC/AC converter k , without matching control, has the following components:

Symbol	Type	Description	Info.
$v_{dc,k}$	State	Voltage on the DC side in pu	$v_{dc,k} \in \mathbb{R}$
$i_{c,k}$	State	Converter AC current flowing in, in pu in dq framework	$i_{c,k} \in \mathbb{R}^2$
m_k	Input	Averaged modulation signal for each dq framework	$m_k \in \mathbb{R}_{[-1,1]}^2$
$i_{dc,k}$	Input	Current source from buck/boost that supplies the DC side	$i_{dc,k} \in \mathbb{R}$
$R_{c,k}$	Parameter	Series resistance of the output RL filter, in pu	$R_{c,k} = r_{c,k} \mathbb{1}_{2 \times 2}$
$L_{c,k}$	Parameter	Series inductance of the output RL filter, in pu	$L_{c,k} = \ell_{c,k} \mathbb{1}_{2 \times 2}$
$Z_{c,k}$	Parameter	Series impedance of the output RL filter, in pu	$Z_{c,k} = R_{c,k} + j\omega_0 L_{c,k}$
$G_{dc,k}$	Parameter	Shunt conductance on the DC side	$G_{dc,k} \in \mathbb{R}_{\geq 0}$
$C_{dc,k}$	Parameter	Shunt capacitance on the DC side	$C_{dc,k} \in \mathbb{R}_{\geq 0}$

Table 4: States, inputs and parameters of each converter k .

Each converter k is also described with an indicator vector $I_{c,k} \in \{1, 0\}^{n_v}$ that has a 1 on the bus that is located and 0 otherwise.

For each converter, the model requires two states, the voltage on the DC side and the output AC current in d and q components. Thus, converters contribute $3n_c$ states to the model. In addition, each converter adds two inputs, the DC current coming from the energy source and the modulation signals that construct the AC voltage.

2.2.5 Loads

Each balanced load k is defined by:

Symbol	Type	Description	Info.
$i_{l,k}$	State	AC current flowing into load, in pu in dq framework	$i_{l,k} \in \mathbb{R}^2$
$R_{l,k}$	Parameter	Resistance of the load, in pu	$R_{l,k} = r_{l,k} \mathbb{1}_{2 \times 2}$
$L_{l,k}$	Parameter	Inductance of the load, in pu	$L_{l,k} = \ell_{l,k} \mathbb{1}_{2 \times 2}$
$Z_{l,k}$	Parameter	Impedance of the load, in pu	$Z_{l,k} = R_{l,k} + j\omega_0 L_{l,k}$

Table 5: States and parameters of each load k .

Each load k is also described with an indicator vector $I_{l,k} \in \{1, 0\}^{n_v}$ that has a 1 on the bus that is located and 0 otherwise.

For each load, the model requires 2 states, the AC current flowing into the load, both in d and q components. Thus, loads contribute with $2n_l$ states to the model.

2.3 Dynamical system

The authors organize the model dividing its states in two categories:

- The so called “DC” states x , that are the ones that represents DC states, that should converge to a fixed value, given by:

$$x = (\theta_g, \omega_g, v_{dc})$$

- The so called “AC” states z , that are the ones that represents AC states in their original frame. Those are:

$$z = (i_g, i_c, i_l, i_t, v)$$

In addition, the inputs are defined by $u = (\tau_m, i_f, i_{dc}, m)$. With this, the system can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} f_{dc}(x, z, u) \\ f_{ac}(x, z, u) \end{bmatrix}$$

2.3.1 DC dynamical system

The DC equations are given by $f_{dc}(x, z, u)$:

$$\dot{\theta}_{g,k} = \omega_0(\omega_g - 1) \quad \forall k \in \{1, \dots, n_g\} \quad (\text{DC-1})$$

$$\dot{\omega}_{g,k} = \frac{M_k^{-1}}{2} \left\{ -D_k(\omega_g - 1) + \tau_{m,k} - \tau_e(\theta_{g,k}, i_{g,k}, i_{f,k}) \right\} \quad \forall k \in \{1, \dots, n_g\} \quad (\text{DC-2})$$

$$C_{dc,k} \dot{v}_{dc,k} = -G_{dc,k} v_{dc,k} + i_{dc,k} - i_{sw,k}(i_{c,k}, m_k) \quad \forall k \in \{1, \dots, n_c\} \quad (\text{DC-3})$$

with $\omega_0 = 2\pi \cdot 60$ rad/s and:

$$\tau_e(\theta_{g,k}, i_{g,k}, i_{f,k}) = -\ell_{m,k} i_{f,k} i_{g,k}^\top j r(\theta_{g,k}) \quad \forall k \in \{1, \dots, n_g\} \quad (\text{Aux-1})$$

$$i_{\text{sw},k}(i_{c,k}, m_k) = \frac{1}{2} i_{c,k}^\top m_k \quad \forall k \in \{1, \dots, n_c\} \quad (\text{Aux-2})$$

Equations (DC-1) and (DC-2) are the standard equations for a 2-states model of a synchronous machine. The electrical torque produced by the synchronous machine is given by equation (Aux-1). Note that the induced voltage for a generator is given by:

$$v_{\text{ind},k} = \ell_{m,k} i_{f,k} \omega_{g,k} j r(\theta_{g,k}) \quad (\text{Aux-3})$$

that implies that the electrical torque of the machine k is simply given by:

$$\tau_e = \frac{i_{g,k}^\top v_{\text{ind},k}}{\omega_0 \omega_{g,t}} = -\ell_{m,k} i_{f,k} i_{g,k}^\top j r(\theta_{g,k}) \approx v_{\text{ind},k}^d i_{g,k}^d + v_{\text{ind},k}^q i_{g,k}^q = p_e$$

as expected for a classical model of synchronous machine modeled by 2 states and an EMF behind a reactance.

Finally, equation (DC-3) is the current Kirchhoff law on the DC side of the converter. The averaged DC current flowing out through the switches is given by $i_{\text{sw},k}$ defined in equation (Aux-2).

2.3.2 AC dynamical system

The AC equations are given by $f_{\text{ac}}(x, z, u)$:

$$L_{t,k} \dot{i}_{t,k} = -Z_{t,k} i_{t,k} + \mathcal{E}_{t,k}^\top v \quad \forall k \in \{1, \dots, n_t\} \quad (\text{AC-1})$$

$$C_k \dot{v}_k = -Y_{v,k} v_k + \mathcal{E}_{t,k} i_t + i_{\text{in},k} \quad \forall k \in \{1, \dots, n_v\} \quad (\text{AC-2})$$

$$L_{g,k} \dot{i}_{g,k} = -Z_{g,k} i_{g,k} + \mathcal{I}_{g,k}^\top v - v_{\text{ind},k} \quad \forall k \in \{1, \dots, n_g\} \quad (\text{AC-3})$$

$$L_{c,k} \dot{i}_{c,k} = -Z_{c,k} i_{c,k} + \mathcal{I}_{c,k}^\top v - v_{\text{sw},k} \quad \forall k \in \{1, \dots, n_c\} \quad (\text{AC-4})$$

$$L_{l,k} \dot{i}_{l,k} = -Z_{l,k} i_{l,k} + \mathcal{I}_{l,k}^\top v \quad \forall k \in \{1, \dots, n_g\} \quad (\text{AC-5})$$

on where $\mathcal{I} = I \otimes \mathbb{1}_{2 \times 2}$ is the indicator matrix on which each device is located, but expanded to consider both d and q frames. In addition, $v_{\text{ind},k}$ is given by (Aux-3) and:

$$i_{\text{in},k}(i_g, i_c, i_l) = [\mathcal{I}_g i_g + \mathcal{I}_c i_c + \mathcal{I}_l i_l]_k \quad \forall k \in \{1, \dots, n_v\} \quad (\text{Aux-4})$$

$$v_{\text{sw},k}(v_{\text{dc},k}, m_k) = \frac{1}{2} v_{\text{dc},k} m_k \quad \forall k \in \{1, \dots, n_c\} \quad (\text{Aux-5})$$

Equation (AC-1) is the Kirchhoff voltage law, that states that the evolution of the current through a line k is driven by the difference of voltages between the buses the line is connected and the line impedance.

Equation (AC-2) is the evolution of the bus voltage k , that depends on the shunt admittance and the current flowing in/out to connected lines and the current flowing in/out of the devices

that are connected on that bus. These currents are given by (Aux-4), that port current from devices that are connected to bus k .

Equation (AC-3) is the evolution of AC stator current of each generator k , that depends on the series impedance, the voltage that is connected and the induced EMF behind that impedance. That induced voltage is given by equation (Aux-3).

Equation (AC-4) is the evolution of AC output current of each DC/AC converter k , that depends on the output RL filter impedance, the bus voltage that the converter is connected and the averaged switched voltage given by equation (Aux-5).

Finally, equation (AC-5) is the evolution of AC current flowing into 3-phase balanced loads, that depends on the bus voltage that the load is connected and its impedance.

3 Software implementation

The entire model was coded in Matlab and it is publicly available in the EMAC repo at <https://github.com/Energy-MAC/CuriDynamicModel>.

3.1 Data

Data is structured using csv files. Buses, lines, generators, converters and loads has their own csv file on which the number of devices and parameters are specified.

	A	B	C	D	E	F	G
1	GenName	BusNumber	Resistance [pu]	Inductance [pu]	MutualInductance [pu]	Inertia [MWs/MVA]	Damping [pu]
2	Gen1	1	0.130434783	0.0000457	0.0000430	5.148	2
3	Gen2	2	0.130434783	0.0000457	0.0000430	6.54	2

Figure 1: Generator csv data. Each row is a generator with parameters defined at each column.

Depending on how many devices are considered in the data files the model will define the necessary matrices and parameters to construct the system of differential equations.

We also wrote a utility to import network topology and line impedances from MATPOWER cases into our model (see ‘load_network.m’).

3.2 Dynamical equations

Each equation is defined in a particular Matlab function script that receives the parameters and return the right hand side of the aforementioned equations.

Functions are defined in a matrix form when is possible to reduce computational burden when constructing the model.


```

1 function domegag_dt = domega_gens_dt(theta_g, omega_g, i_g, i_f, tau_m, invM, D, l_m)
2 %This function represents generator dynamics based off Equation (4b) in Curi Paper
3
4 num_gens = length(omega_g);
5 tau_e = zeros(num_gens,1);
6
7 for i=1:num_gens
8     tau_e(i) = electricalTorque(theta_g(i), i_g(2*i-1:2*i), i_f(i), l_m(i));
9 end
10
11 domegag_dt = invM*(-D*(omega_g-1) + tau_m - tau_e)/2;
12 end

```

Figure 2: Equation (DC-2) defined in the Matlab script. For this equation the left hand side term M is moved to the right hand side as M^{-1} .

With all the equations properly defined in the script, a differential equation function script is used. This script has the main purpose of stacking the entire dynamical system in a single vector. This script is then called by the method `ode15s` in Matlab. The mass function of the method is also defined based on the parameters on the left hand side of the equations.

Finally, our main script has two primary parts. First, it reads all the input files and construct the parameter and matrices to be used in the functions. Second, it calls the method `ode15s` to solve the entire dynamical system.

4 Discussion of model and extensions

4.1 Stiff modelling and model order reduction

One of the key results mentioned in [1] is that the AC equations can be simplified by the model order reduction method proposed to algebraic equation by setting the derivative to zero. By comparing the DC and AC equations, it can be seen that that most of DC equations have time constant between 1 ~ 10s, except the DC voltage of the DC side of the converters. On the other hand, AC equations have time constants that depend on the inductance or capacitance of the devices. Those values are on the order of 10^{-5} seconds. This a extremely stiff model.

By trying to solve this model directly as an ODE, for example using methods like `ode45` or `ode23` by passing the left hand side time constant to the right hand side (i.e. dividing for that inductance/capacitance), the model doesn't finish the simulation in a reasonable timeframe. This is expected, since those methods are suited for nonstiff differential equations.

As an alternative, we use a method for stiff differential equations, like `ode15s` to solve it. However, given the typical values, there is no difference in setting the left hand side (that is the Mass Matrix) to the values of 10^{-5} or directly setting those to zero. The `ode15s` method realize that those equations are essentially algebraic and don't use a smaller time step to properly solve those ODEs. Thus, in practice, the model order reduction proposed in [1] is always enforced in a simulation because it is not practical (because it is overwhelmingly computationally expensive) to implement those equations as ODEs.

4.2 Finding DC and AC equilibrium points

As the model presented in the paper includes no feedback control, the choice of inputs is left to the users. During our debugging process, we wanted to be able test whether the system had stable equilibria. This introduced the need to find inputs that would admit an equilibrium. For example, with a single generator and a single load, arbitrary inputs of mechanical torque and excitation current do not necessarily result in an equilibrium. The task g of finding an equilibrium x^* of a function f with parameters y , and initial guess at the equilibrium x can be written as:

$$g : (x, y, f) \mapsto x^* | f(x^*, y) = 0$$

In our case, we have state-update equations f_{dc} and f_{ac} for our DC and AC states, and each are a functions of inputs u , DC states x , and AC states z :

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} f_{dc}(x, z, u) \\ f_{ac}(x, z, u) \end{bmatrix}$$

For the same reasons that a stiff solver is necessary to simulate the differential equation, a straightforward numerical algorithm (e.g. Newton-Raphson) to finding the equilibrium does not converge. We were unable to find a version of MATLAB's `fsolve` that could handle this ill-conditioned problem, so we built on the approach in the paper to find an equilibrium by separating the DC and AC states and using nested `fsolve` calls:

$$g_{ac} : (z_{init}, (x, u), f_{ac}) \mapsto z^* | f_{ac}(x, z^*, u) = 0$$

$$g_{dc} : (x_{init}, u, f_{dc}) \mapsto x^* | f_{dc}(x^*, g_{ac}(z_{init}, (x^*, u)), u) = 0$$

The first expression amounts to finding an equilibrium of the AC states given the DC states and control, and the second amounts to finding an equilibrium of DC states given control and the condition that the AC states are at equilibrium. g_{ac} can be solved using MATLAB's `fsolve` and is implemented in our project as 'find_ac.m'. g_{dc} can be found with `fsolve` as well, but amounts to finding the zeros of the reduced model introduced in the paper. That is, Curi defines the function:

$$f_{red}(x, u) := f_{dc}(x, h(x, u), u) = f_{dc}(x, g_{ac}(z_{init}, (x, u)), u)$$

It turns out that the way Curi defines $h(x, u)$ as the steady state of the AC variables is equivalent to our definition of g_{ac} above, and it is equivalent for any initial guess z_{init} because f_{ac} is globally exponentially stable with the existence of a unique equilibrium (shown in the paper). We implement this in 'find_eq.m' where we effectively use `fsolve` to find the equilibrium of f_{red} given inputs u , and we use our implementation of g_{ac} to find equilibrium values of the AC states for any DC states and inputs.

As mentioned above, g_{dc} does not have a solution for arbitrary u (if it does not, `fsolve` will not converge, and simulations with this arbitrary input result in sustained oscillations). To address this, we implemented another layer that allows free inputs, meaning that only a subset of inputs would be determined. We implemented this in 'find_eq_u.m', which effectively uses the same process as find_eq.m but would stack the free inputs along with the DC states

and include them in the search for an equilibrium point. Using this approach, we were able to find equilibrium points for the system and associated open-loop control to simulate the system evolution under initial conditions close to the equilibrium and observe the behavior. Results are presented below.

4.3 Closed-loop control

As the model presented in the paper is open-loop, we experimented with adding closed loop controllers to regulate the system to a desirable equilibrium.

4.3.1 Automatic Voltage Regulation (AVR)

We implemented a simple AVR with an integral controller, where for the k 'th generator, the field current $i_{f,k}$ is adjusted based on the difference between the voltage magnitude at its terminals and reference magnitude with gain $k_{\text{avr},k}$:

$$\dot{i}_{f,k} = k_{\text{avr},k}(v_{\text{ref},k} - \|v_k\|_2)$$

4.3.2 Torque control

We implemented a PI controller for mechanical torque. Note that the proportional part reflects classical droop, and we added an integral term to return the frequency to nominal. This distributed, local implementation of the integral controller is different than the centralized, system-wide AGC used in practice; however, this suffices for our simulations with single generators.

$$\begin{aligned}\dot{\phi}_k &= k_{\text{Droop,I},k}(\omega_0 - \omega_g) \\ \tau_{m,k} &= k_{\text{Droop,P},k}(\omega_0 - \omega_g) + \phi_k\end{aligned}$$

4.3.3 Converter control

The two entries of the converter control input m_k allow the converter controller to determine the d and q components of the internal converter switching voltage and current separately. However this additional degree of control freedom is not necessarily desirable, as the other loads and generators in the network are defined in terms of their output power, rather than the d and q components of the current and voltage.

To reduce the converter control input dimension, and have the control inputs more closely reflect the control inputs of rotating machines, [1] introduces virtual oscillator control, based on the virtual oscillator in [2]. The virtual oscillator control introduces a virtual rotating frequency $\omega_{c,k}$ and a virtual internal angle state $\theta_{c,k}$, which, like the generator angle $\theta_{g,k}$, is defined relative to the rotating angle reference θ_0 shared throughout the network. The state equation for $\theta_{c,k}$ is:

$$\dot{\theta}_{c,k} = \omega_0(\omega_{c,k} - 1), \quad \forall k \in \{1, \dots, n_c\} \quad (\text{DC-4})$$

The creation of the virtual internal angle state $\theta_{c,k}$ allows the converter voltage to be controlled similarly to the generator voltage, with the new control input $\mu_k \in \mathbb{R}_{[0,1]}$ playing an analogous role to the field current $i_{f,k}$ in the rotating generators. Similarly, $i_{dc,k}$ plays an analogous role to the mechanical torque $\tau_{m,k}$ in a rotating generator. m_k is defined in terms of the new input μ_k by the following equation:

$$m_k = \mu_k j r(\theta_{c,k}), \quad \forall k \in \{1, \dots, n_c\} \quad (\text{Aux-6})$$

A key difference between the converter and generator is that the frequency $\omega_{c,k}$ is a controllable input for the user. This feature allows the converter to be controlled to be grid forming, grid supporting, or grid following. For example, strictly controlling a converter with a virtual oscillator rotating at $\omega_{\text{gridforming}}$ creates a grid-forming device. The grid forming device will allow $\theta_{c,k}$, and therefore $P_{c,k}$ and $Q_{c,k}$, to adjust as necessary to maintain $\omega_{\text{gridforming}}$ on the network.

On the other end of the spectrum, controlling $\omega_{c,k}$ to match the grid frequency exactly (using a PLL) creates a grid-following device. For the grid following device, the output power is determined by μ_k and $\theta_{c,k}$, which should be constant because $\omega_{c,k} = \omega_{\text{grid}}$.

In addition to these two modes, the converter can also be controlled to be grid supporting. That is, to adjust its internal frequency based on the instantaneous power imbalance between the power supplied to the converter (in the form of dc current) from the energy source (or sink), and the power delivered to the grid. This power balance is represented in equation (DC-3).

The Curi paper proposes a control for $\omega_{c,k}$, based on the control in [2], titled ‘‘Matching Control.’’ The principle behind Matching Control is to pin the converter frequency to the dc bus voltage. The intuition behind this control is that the DC capacitor is the energy storage element for converters, and v_{dc} plays an analogous role to the rotational velocity ω of a rotating machine. The stored energy in the capacitor is $\frac{1}{2}C_{dc}v_{dc}^2$, and the stored energy in a rotating machine is $\frac{1}{2}M_{dc}\omega^2$ (M = the moment of inertia for the machine). This analogy is also evident in the power balance equations (DC-3) and (DC-2).

The equation that defines the matching control is:

$$\omega_{c,k} = k_{c,k}(v_{dc,k} - v_{dc,k}^*) + \omega_0, \quad \forall k \in \{1, \dots, n_c\} \quad (\text{Aux-7})$$

$v_{dc,k}^*$ is the DC voltage reference, analogous to the rotating frequency reference. $k_{c,k}$ is the Matching Control parameter, which maps a DC voltage deviation to an internal virtual frequency deviation. Note that centering the DC voltage deviation at $v_{dc,k}^*$, rather than at 0, provides restoring (stabilizing) behavior. Furthermore, by tying $\omega_{c,k}$ to $v_{dc,k}$, the converter can provide true inertia to the system.

With the virtual oscillator angle $\theta_{c,k}$, the matching control input $\mu_{c,k}$, and the matching control parameters $k_{mc,k}$ and $v_{dc,\text{ref},k}$ included, the list of states inputs and parameters for the converters is adjusted to:

Symbol	Type	Description	Info.
$\theta_{c,k}$	State	Rotor angle relative to an absolute reference frame θ_r	$\theta_{c,k} \in \mathbb{S}^1$
$v_{dc,k}$	State	Voltage on the DC side in pu	$v_{dc,k} \in \mathbb{R}$
$i_{c,k}$	State	Converter AC current flowing in, in pu in dq framework	$i_{c,k} \in \mathbb{R}^2$
μ_k	Input	Averaged modulation signal for each dq framework	$m_k \in \mathbb{R}_{[0,1]}$
$i_{dc,k}$	Input	Current source from buck/boost that supplies the DC side	$i_{dc,k} \in \mathbb{R}$
$R_{c,k}$	Parameter	Series resistance of the output RL filter, in pu	$R_{c,k} = r_{c,k} \mathbb{1}_{2 \times 2}$
$L_{c,k}$	Parameter	Series inductance of the output RL filter, in pu	$L_{c,k} = \ell_{c,k} \mathbb{1}_{2 \times 2}$
$Z_{c,k}$	Parameter	Series impedance of the output RL filter, in pu	$Z_{c,k} = R_{c,k} + j\omega_0 L_{c,k}$
$G_{dc,k}$	Parameter	Shunt conductance on the DC side	$G_{dc,k} \in \mathbb{R}_{\geq 0}$
$C_{dc,k}$	Parameter	Shunt capacitance on the DC side	$C_{dc,k} \in \mathbb{R}_{\geq 0}$
$k_{mc,k}$	Parameter	Matching control parameter	$k_{mc,k} \in \mathbb{R}_{\geq 0}$
$v_{dc,ref,k}$	Parameter	DC voltage reference	$v_{dc,ref,k} \in \mathbb{R}_{\geq v_k}$

Table 6: States, inputs and parameters of each converter k .

With these definitions, the controllers for the converters now closely reflect the controllers for the synchronous generators. Thus, μ_k can be used for AVR to control the AC voltage at the bus, and $i_{dc,k}$ can be controlled using a PI droop controller to properly match the required real power and restore the DC voltage to the set point.

5 Simulations

5.1 2 Buses: 1 Generator - 1 Load

We used this simulation for debugging and to verify expected behavior for a simple system. We placed a generator at bus 1 and a load at bus 2. The load was modelled as constant impedance to give $P = 1$ and $Q = 0.05$ at a voltage magnitude of 1. Please see the case data under ‘cases\debug2bus\’.

5.1.1 Finding equilibrium point

We first tested the open-loop system without any infinite bus by finding an equilibrium point using the tool described above and perturbing the system slightly. We held the excitation current as a fixed input at 1 p.u. and calculated that a mechanical torque of 1.0185 admitted a set of equilibrium AC and DC states. We verified that the system stayed at this equilibrium with a frequency of 1 p.u. We then applied initial conditions with all states at their equilibrium value except we set generator frequency at $\omega_g = 1.01$ and observed the system converge to the equilibrium as shown below (please see the file ‘main_2bus_example.m’ for code). Note that dq components are specified in the global reference frame with frequency 1, so we observe that though their magnitudes do not change much, each component oscillates while the generator rotor frequency is different than the global reference frame frequency (Figure 3). As there is no control, this convergence occurs because of damping in the generator swing equation.

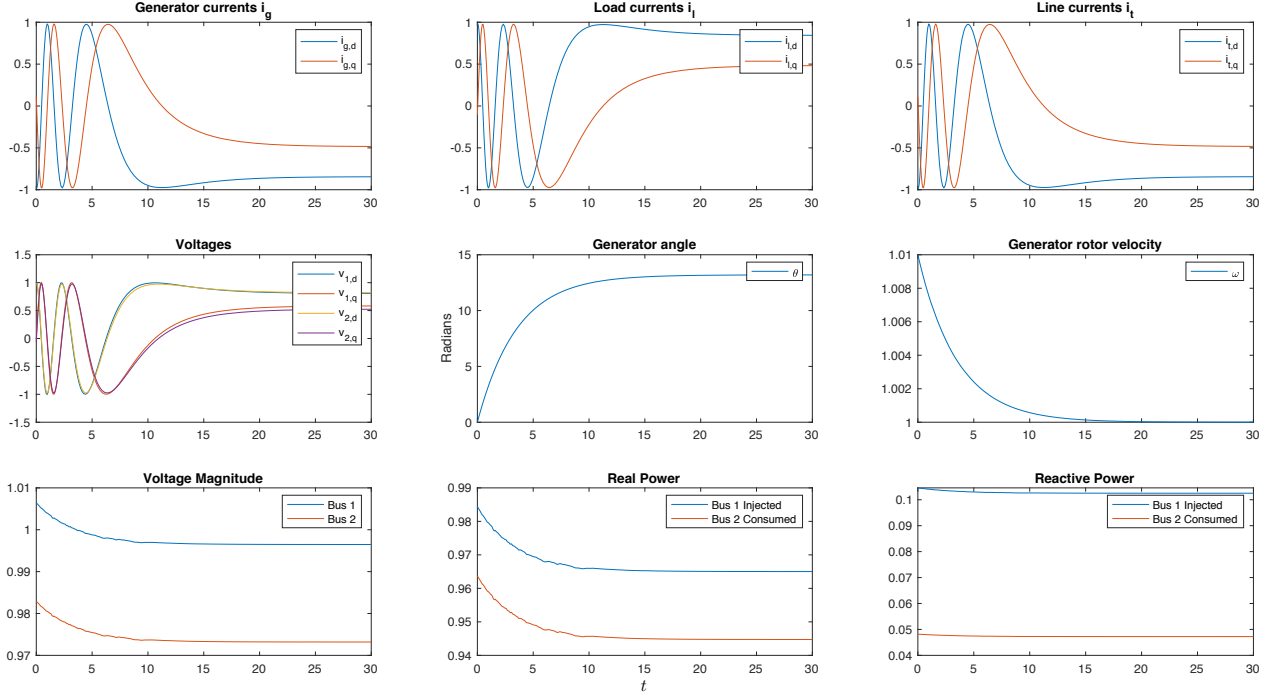


Figure 3: Results for 2 bus example with generator and load. As can be seen the frequency converges back to their equilibrium at $\omega_g = 1$ due to the generator damping.

5.1.2 Using AVR

We implemented the AVR with gain $k_{\text{AVR}} = 1$ without any turbine control to regulate the generator bus voltage (bus 1) to 1 p.u. The results in Figure 4 show that the voltage is regulated to 1 and the DC states of rotor velocity and field current approach asymptotes. 2 phenomena are important to observe. First, because the rotor velocity is different from our global reference frame $\omega_o = 1$ the generator angle relative to the global reference frame is changing. This is what causes the sustained oscillations in the dq components of the generator current and voltages even as the voltage magnitude is constant. Second, there is some noise in the the voltage magnitude and real power. We think this is a numerical issue resulting as those quantities are non-linear and the simulation uses a variable step-size. We implemented this simulation by setting the PI gains on the turbine controller to zero.

5.1.3 PI controller for mechanical torque

The simulation including the PI turbine controller is shown in Figure 5. We used gains $k_{\text{Droop,I}} = 2$ and $k_{\text{Droop,P}} = 20$. In contrast to the simulation without turbine control, the generator frequency returns to 1 and we see that the dq components of voltage and current reach steady state values. The noise in voltage magnitude and power also dies out as the generator angle converges to a steady value. It is worth noting that the d and q components of each voltage and current effectively switch after the transient; e.g. initially the d components of voltage are

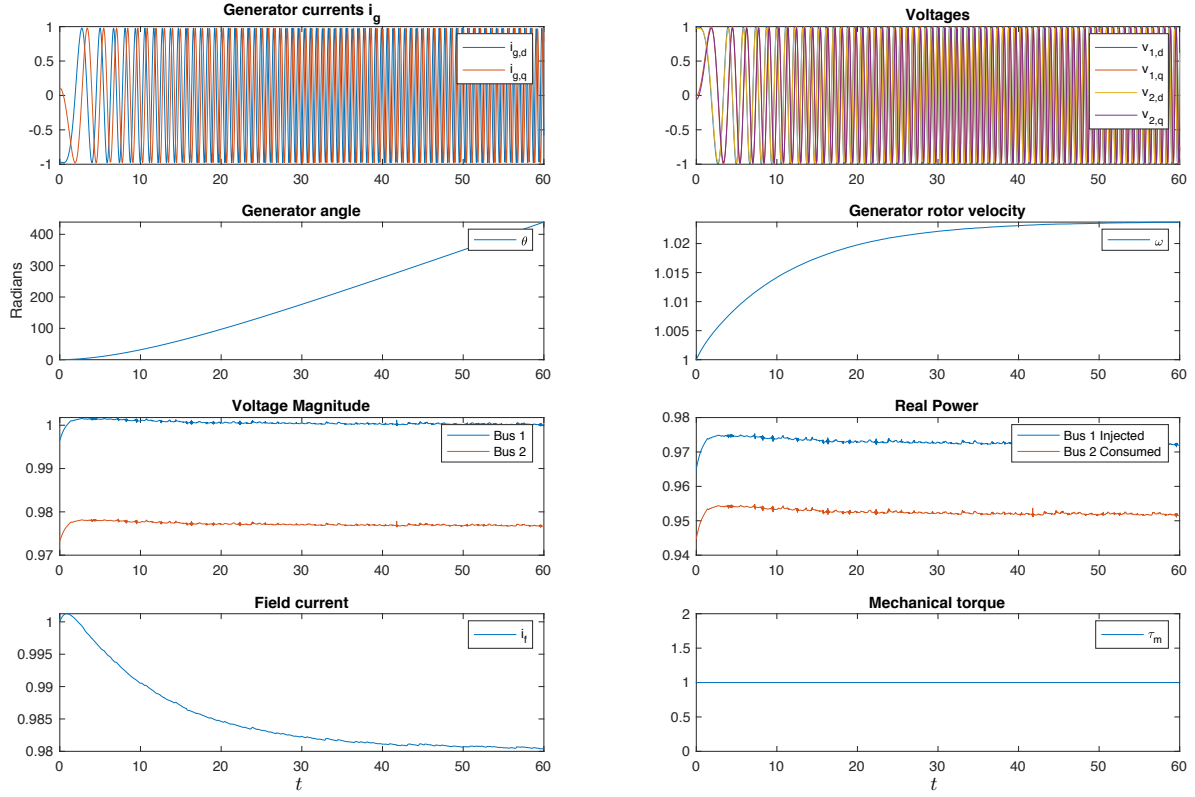


Figure 4: 2 bus example with AVR on generator. There is an excess of mechanical torque, that increases the speed of the rotor.

1 and they settle to close to zero. It's not clear whether or not this represents any problem or potential bug; we think this just represents the angle of the generator's reference frame being offset from the global reference frame.

5.2 2 Buses: 1 Converter - 1 Load

The purpose of this simulation was to debug and verify the behavior of a system with a converter running model matching virtual oscillator control. The simulation includes a converter on bus 1 and a constant impedance load on bus 2. To evaluate the performance of the model matching control, we compared the behavior of a controller running model matching control with the performance of generator running grid forming control (also with an internal virtual oscillator).

5.2.1 Grid forming simulation

In this simulation, we set the model matching constant $k_{c,k}$ to zero. This set our internal frequency at the constant value $\omega_{c,k} = 1$, regardless of the power draw and DC voltage. The code for this test case can be found in 'cases\debug2bus_converters\' , and the relevant script is `main_2bus_converters.m`.

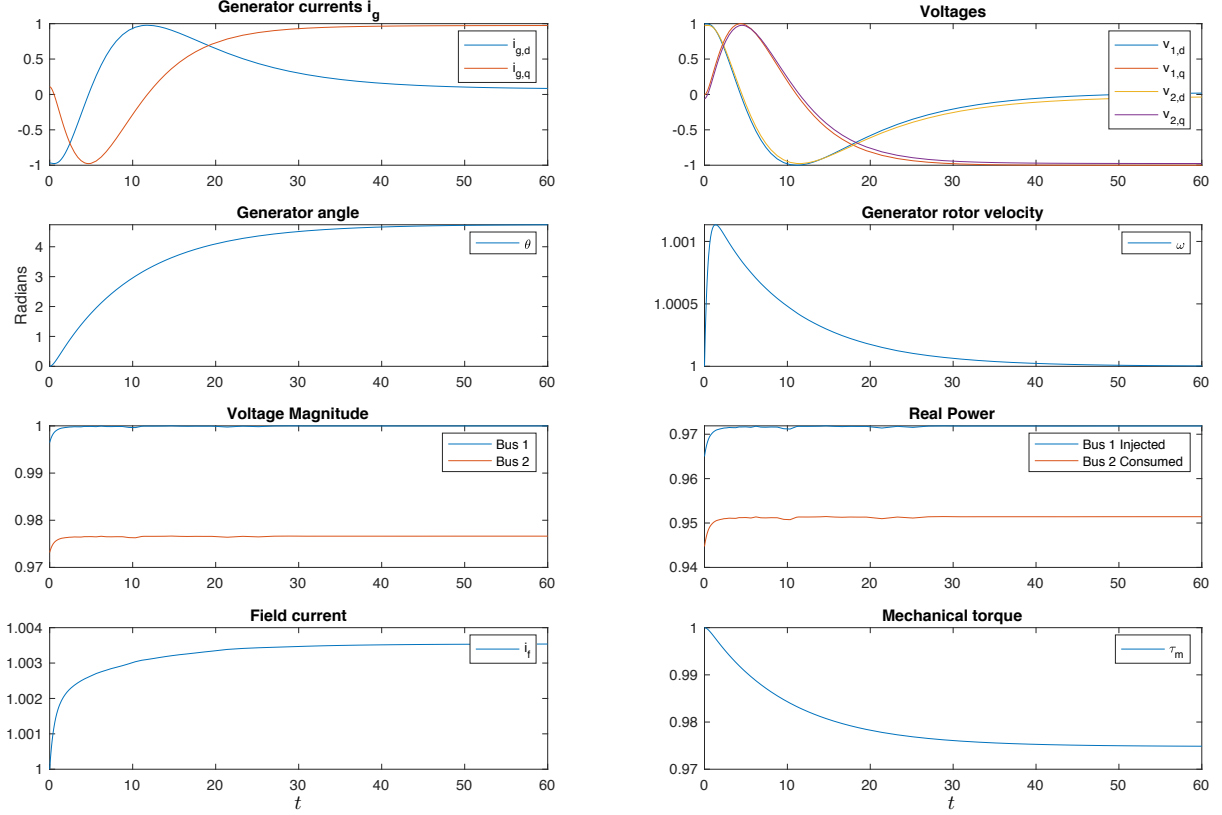


Figure 5: Results for 2 bus with AVR and turbine controller on the generator.

The equilibrium point for a constant impedance load that draws $P = 1$ and $Q = 0.05$, is obtained at $v_{dc}^* = 2$ with $i_{dc} = 0.3507$ and $\mu = 0.8497$. We compared behaviours by increasing the load by 20% (by decreasing the real part of the impedance by 20%). The results for the simulation are shown in figure 6. As can be seen, the converter's virtual angle remains constant despite the change in the load, but the DC voltage and AC voltages drop.

5.2.2 Matching controller

For this simulation we repeat the same study (decreasing the real part of the impedance by 20%) for a converter running matching control. The matching constant is set to $k_c = 0.1$. With matching control, we expect the internal frequency of the virtual oscillator to be proportional to the difference between the DC voltage and the reference DC voltage set-point. We set $v_{dc,ref} = 2$.

The results of the matching control simulation are shown in figure 7. The DC voltage converges to $v_{dc} = 1.6192$, which corresponds to an internal frequency of:

$$\omega_c = 0.1(1.6192 - 2) + 1 = 0.96192$$

Thus, the virtual rotating frequency of the converter deviates from the reference frame frequency ω_0 . As showed in Figure 7, this results in oscillating d and q voltage and current

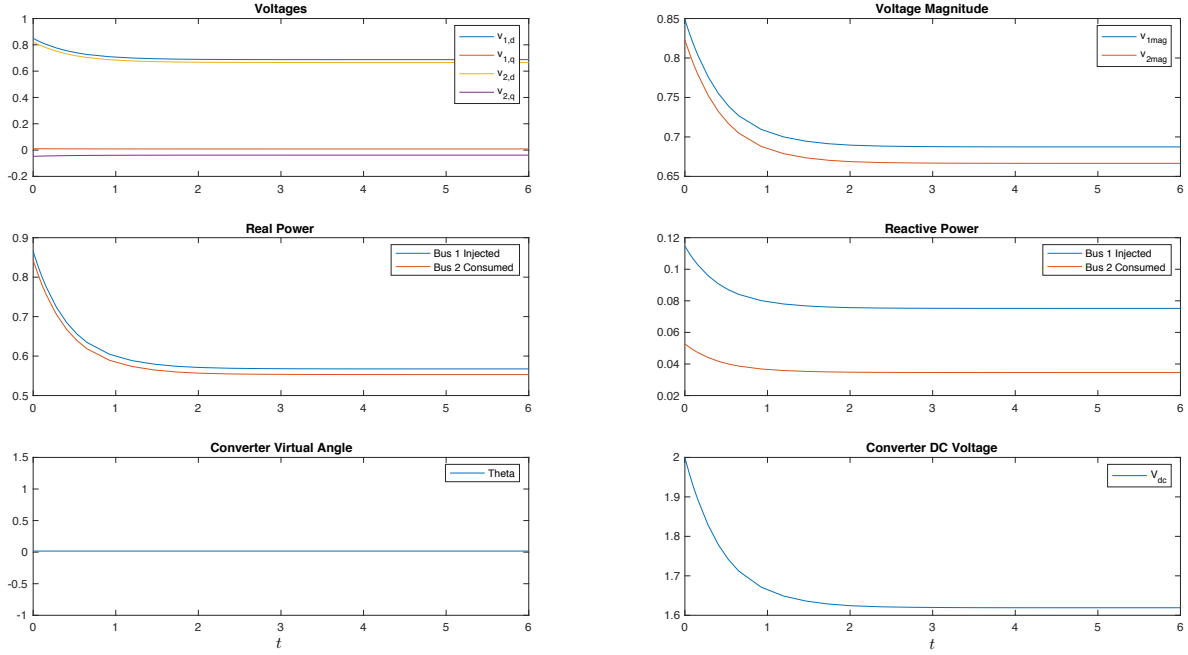


Figure 6: Results for 2 bus case with 1 converter and 1 load, as a grid-forming device.

components. However the AC voltage magnitudes and output power decay in the predictable decaying-exponential form. This demonstrates the grid is oscillating at $\omega_{c,1}$, not ω_0 . Thus, the matching controller is functioning properly.

In future work, we intent to study the effect of traditional droop controllers, as well as other types of grid-supporting controllers, with the matching controller.

6 Conclusions

In this report we discuss the model presented in [1]. The proposed model is coded and simulated in Matlab to assess its behaviour and performance. We simulated two simpler study cases, considering a two-bus system with 1 generator and 1 load, and a two-bus system with 1 converter and 1 load.

We discussed the importance of algebraic equations, as a computational issue, instead of fast differential equations in stiff models. In addition, we address the importance of finding initial input setpoints and equilibrium points to properly set-up the simulation. We also discussed and demonstrated the merits of Matching Control for inverters.

Simulations showed that the model proposed can be effectively used to simulate systems, if the assumption considered in [1] are valid.

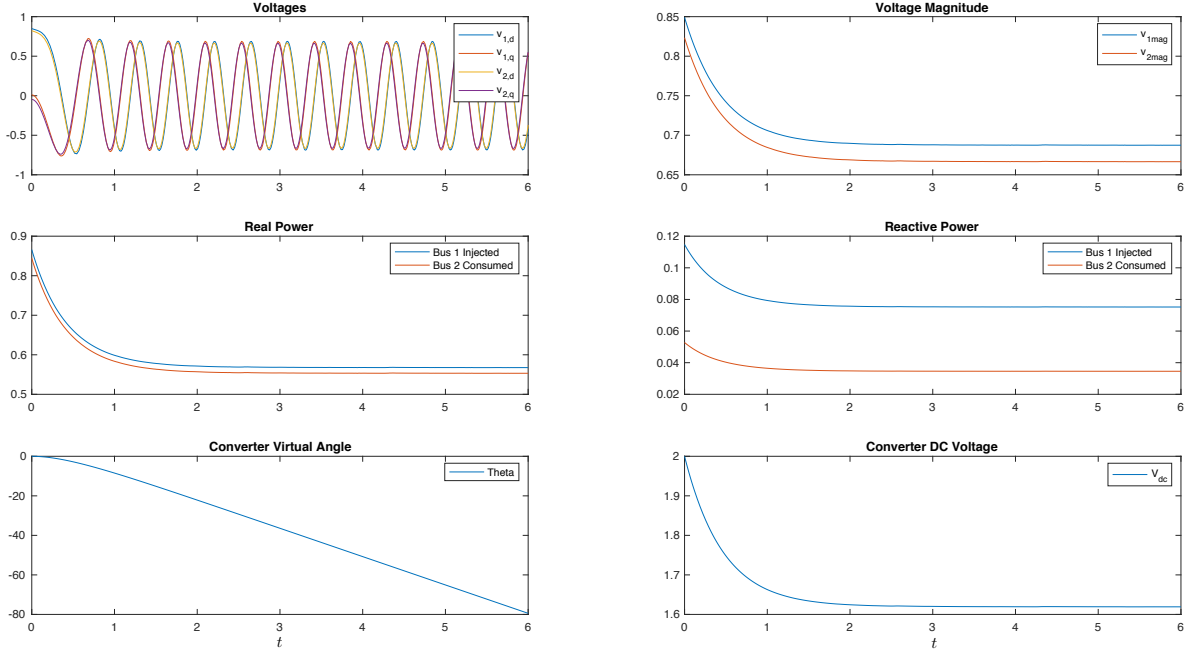


Figure 7: Results for 2 bus case with 1 converter and 1 load, with matching constant $k_c = 0.1$.

6.1 Lessons learned

Several lessons were learned in doing this project among we can highlight:

6.1.1 Divide and conquer

When implementing models that has many aspects and complexities, it is important to divide the problem in smaller ones that can be easily understood and simulated properly. At the beginning we attempt to run the entire model with many features, but the model didn't converge to anything reasonable. Due to this, we then decided to start up running simpler problems, that eventually work out.

6.1.2 Theory and computational simulations

One of the key features of the studied paper is that it proposes a technique to simplify the model based on a model order reduction theorem. We observed that in practice, to simulate the model, the proposed reduction is necessary to actually run the model. Essentially, it is computationally expensive to run stiff models, where time constants between ODEs differ on the order of 10^6 s, and simplify to algebraic equations is a necessary process.

6.1.3 Initial conditions and equilibrium points

Working with the model we realize that is not simple to initialize the models. We realize that by arbitrarily choosing inputs may yield to non-existent equilibrium points, or unstable equilibria. For this purpose, it is critical to implement scripts that properly handle the set points of the inputs that can be used to then run effective simulations.

6.2 Team member contributions

- Rodrigo set-up the repository in Github and the final report in Overleaf. He wrote the basis of the main code, on where it reads csv files and transform it to the required matrix form, that are then added to a parameter struct. He also set-up the main ODE script where stack all the vectors and wrote the functions for differential equations of buses, lines and port variables. For the report, he wrote the introduction, the modelling framework, the software implementation and the conclusion.
- Jonathan set-up and ran the simulations for the study case of generator and 1 load. He wrote the functions for differential equations of loads and code to import load data from csv files and the utility for importing MATPOWER network/line data, though this utility was not used in our ultimate simulations. He wrote the code and scripts for both AVR field current controller and PI torque controller adding a wrapper to the simulation for a general feedback controller. He also wrote and set-up the scripts to find initial conditions and equilibrium points for both AC and DC system. For the report, he wrote the discussion of model and extensions and the simulations for the 1 generator and 1 load study case.
- Keith helped with the main code and wrote the differential equation functions for DC/AC converters. This included implementing the Model-Matching Virtual Oscillator Control in the converters. He set-up and ran the simulations for the converter test cases, with the help of Rodrigo. In addition, he reformulated the main code so that the code can work with any combination of converters, generators, and infinite buses, without adjusting the code. For the report, he wrote the converter control section, and the converter simulations section.
- Victoria wrote the differential equation function for the synchronous generators.

References

- [1] S. Curi, D. Groß, and F. Dörfler, “Control of low-inertia power grids: A model reduction approach,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 5708–5713.
- [2] C. Arghir, T. Jouini, and F. Dörfler, “Grid-forming control for power converters based on matching of synchronous machines,” *Automatica*, vol. 95, pp. 273–282, 2018.