

CHALLENGES IN TEACHING THE DIFFERENCE BETWEEN PHYSICAL AND NUMERICAL INSTABILITY

Federico Milano

University College Dublin (IRELAND)

Abstract

This paper describes a subtle but crucial problem that often arises in the dynamic analysis of complex nonlinear systems, namely the difference between physical and numerical instability. The former is the consequence of the intrinsic behaviour of the system that is simulated, while the latter is an issue of the numerical technique, typical an algorithm, that is used to simulate the system. The paper discusses the didactical challenges to teach the difference between the two kind of instabilities and provides a variety of examples based on the lectures and the lab activities of the module "Power System Dynamics and Control" taught by the author at University College Dublin in the last five years.

Keywords: Electrical energy systems, electrical engineering education, stability analysis, numerical methods, computer-based laboratory.

1 INTRODUCTION

The author has about fifteen-year experience in teaching, in different institutions and countries, power systems modules, including power system modelling, control and stability analysis and in developing open-source software tools for research and educational purposes [1-8]. A relevant issue that has been observed is the difficulty that students encounter when they have to discuss simulation results of large systems that depends on a large number of parameters and/or of variables. A typical question that often arises is whether the "instability" that a system shows is due to the fact that the system is actually unstable or it is due to some numerical issues, e.g. the numerical scheme shows an erratic behaviour but the system is actually stable.

The first didactic issue to solve is to make the students understand the difference between these two kinds of instability. Explaining the distinction between mathematical models and computer-based implementation is anything but trivial. The second challenge is to find methods, which depend on the analysis to be carried out, to distinguish between the two. Very often, the distinction can be done introducing advanced mathematical concepts which are not suitable for the module level. Hence, the paper discusses alternative "intuitive" techniques that allow solving the problem, if not rigorously, at least qualitatively. The proposed didactic approach has been tested by the author in his modules "Power System Dynamics and Control" and "Stability Analysis of Nonlinear Systems" taught since 2013 at the UCD School of Electric and Electronic Engineering.

The proposed approach, while necessarily approximated, is based on the idea that "physical" instabilities depends only on the parameters of the devices that form the system, whereas "numerical" instabilities depends on the parameters of the numerical methods utilised to solve the simulation. While this is not always fully true, a sensitivity analysis that systematically perturbs either physical parameters or the parameters of the numerical methods can often lead to distinguish the kind of instability that is observed in the simulations.

The didactical challenge, however, is not just to learn a set of techniques to identify numerical and physical instabilities but, rather, to understand that deep conceptual difference between the two. On one hand, physical instability is a property of the mathematical model that is used to describe the reality. Such a model can be approximated and, in some cases, one of which is also discussed in the paper, the instability can be removed by modifying the model. Numerical instability, on the other hand, is an "additional layer" between the mathematical model and its solution. Since we generally do not have a closed-form analytical solution of a set of nonlinear differential equations, we need to rely on numerical methods, typically handled by a computer, to find such solutions. Unfortunately, such methods introduce their own issues, typically due to numerical approximations and/or discretizations of continuous equations, which, in many cases, may be indistinguishable from the physical instability. Understanding how numerical approximations impact on students' ability to interpret the behaviour of a

complex nonlinear dynamical system is a beautiful and tough didactical challenge, a description of which is provided in the paper.

The paper provides the following contributions.

- A discussion on the didactic challenges of teaching computer-based modelling of nonlinear dynamic systems to students of engineering modules. This discussion is particularly focused to electric power systems, but main concepts can be applied to any engineering area involving sets of nonlinear differential equations.
- A variety of examples that illustrate the occurrence of numerical instabilities that can be confused with an actual instability of the simulated system and qualitative approaches that allows discriminating between physical and numerical instabilities. Students' feedback is duly discussed whenever relevant.

The remainder of the paper is organized as follows. Section 2 describes the contents and the computer-based lab activities of the module “Power System Dynamics and Control” and discusses the main challenges that the students face when simulating power systems. Section 3 discusses at a high level the steps required to modelling and simulating physical models and briefly discusses the challenges related to power system modelling and dynamic analysis. Section 4 presents a variety of examples, all based on the lab activities of the module “Power System Dynamics and Control” and illustrates qualitative methods to distinguish between physical and numerical instabilities. Finally, Section 5 draws conclusions and outlines future work.

2 OUTLINES OF POWER SYSTEM DYNAMICS AND CONTROL

This section provides a brief description of the programme of the module “Power Systems Dynamics and Control” (PSDC) as thought at the 4th stage of the BE and ME programmes of Electrical Energy Systems of University College Dublin by the authors. Reference books of PSDC are [9-12].

2.1 Module Contents

PSDC introduces the main control requirements of the most important devices that compose a HV transmission system. The focus of the module is on the dynamic behaviour of power systems. This includes frequency control, voltage control and auxiliary controllers aimed to improve the stability of the network. All topics are explained both theoretically and with simulation-based examples.

The module is divided into three parts.

- Part I: Modelling and control of the synchronous machine. Dynamic model of the machine, Park transformation and per-unit equations. Primary and secondary controls including automatic voltage regulators, turbine and turbine governors, under and over-excitation limiters, and power system stabilizers. Synchronous machine secondary controls including automatic generator controllers and secondary voltage regulators.
- Part II: Transformer and FACTS device controllers, including under-load tap changers and phase shifters. VSC model and controls. Control of shunt and series FACTS devices and HVDC-links.
- Part III: Distributed energy sources control with particular emphasis on models and controllers of wind turbines and energy storage devices (MPPT, voltage control, frequency control, etc.).

The learning outcomes of the module are: basic concepts of power system frequency and voltage control; knowledge of control systems of all principal devices for high voltage transmission systems; and practical examples based on numerical simulations.

2.2 Laboratory Activities

PSDC includes 1 introductory lecture of up to 2 hours and 4 lab activities, 2 hours each. The activities are intended to develop the following skills: critical analysis of the results; ability to solve problems; ability to identify the key aspects of a certain phenomenon; and the ability to understand the response of a power system based on simulation results. The latter is a crucial skill that the students are expected to achieve. Based on this skill, they are able to decide whether the results that they have obtained are reasonable or not and, if not, to find out why. After each laboratory, the students have to

submit a report which is evaluated based on the consistency of the organization of the matter, clarity of exposition, completeness of results and correctness of conclusions.

The lab activities of PSDC are: (i) inertial response, primary and secondary frequency regulation of synchronous machines; (ii) automatic voltage regulation of synchronous machines and power system stabilizers; (iii) voltage regulation of under-load tap changers and FACTS devices (namely SVC and TCSC); and (iv) frequency control through non-synchronous devices.

2.3 Main Challenges

The PSDC module is challenging for the students for at least two reasons, as follows.

- PSDC is the first module that discusses the dynamics of complex dynamic nonlinear systems. The students have attended, in the third stage, a module on “modelling,” but this is oriented to model and simulate either single devices or systems that can be described by a relatively small number of equations. The understanding of the interaction of several nonlinear devices, machines and controllers is thus new.
- PSDC is the first module where the students encounter the concept of numerical instability. The very concept of numerical instability is not directly discussed in the module, mostly because of lack of time, but can impact on the simulation results. The teaching assistants of the module do provide during the labs support to the students and help them with tricky simulations.

The interested reader can find further details on the didactical challenges of the PSDC module and its lab activities in [1, 2, 7, 8, 13].

3 POWER SYSTEM MODELLING

The model utilized to simulate electro-mechanical transients of power systems is a set of nonlinear differential algebraic equations, as follows:

$$\dot{x} = f(x, y) \quad (1)$$

$$0 = g(x, y) \quad (2)$$

where x are the state variables, y are the algebraic variables, f are the differential equations and g are the algebraic equations.

Equations (1) model the dynamic behavior of synchronous machines and system controllers, such as primary frequency and voltage regulators. Equations (2) model the transmission system, namely lines and transformers, as well as any other algebraic constraints of the devices included in the grid.

A typical study consists of three steps: (i) find an equilibrium point of (1)-(2); (ii) determine whether such an equilibrium point is stable; and (iii) solve a time domain simulation that includes a “large disturbance”, i.e. a fault or a line outage. The third step can be repeated for as many contingencies as required.

The procedure above is repeated several times by the students for each lab activities. The differences among labs rely basically only in the kind of devices and controllers considered in the simulations as well as in the network topology. There is, however, a common thread that unifies all labs, namely the methodological approach to the modelling and simulation of power systems.

In first place, the fact that equations (1) and (2) are a reliable representation of the actual power system is an assumption that it is accepted by the students without much questioning. The fact that reality can be represented as a set of equations, however, is not a given. In the past (early 60s), power systems were studied using analogic circuits (called Transient Network Analyzers, TNAs, [12]) that were able to reproduce the same dynamic behavior of the real-world system. Clearly, the TNAs were not flexible as modern computer-based simulations are and when computers were powerful enough to simulate realistic power systems, TNAs quickly disappeared from the scene.

Another important aspect that is not immediate to assimilate is the fact that (1)-(2) is the best approach to model the system is not written in stone. A set of differential-algebraic equations is adequate only if the time scale and the dynamics of interest are the electro-mechanical oscillations of synchronous machines (period between 0.5 and 2 s) and the primary regulation (between 100 ms and

10 s). If electro-magnetic transients were to be studied, the model would change considerably and would consist almost exclusively of differential equations. This difference is so important that, in power system analysis there are two major “families” of simulators: the ones that consider electro-mechanical models and rms values of network voltages and currents; and the ones that consider electro-magnetic transients (EMT). This paper discusses exclusively the first class of models and simulators. The issues to distinguish between physical and numerical instabilities, however, are similar for all modelling approaches.

Regardless the model approach, the simulator requires a further step before one is actually able to get a solution. This step typically requires discretize the original equations and make them suitable for some iterative numerical method. The discretization transforms the set of continuous equations (1)-(2) into a *map*, as follows:

$$x_{i+1}^k = F(f(x_i^k, y_i^k), g(x_i^k, y_i^k), x_i^k, y_i^k, k) \quad (3)$$

$$y_{i+1}^k = G(f(x_i^k, y_i^k), g(x_i^k, y_i^k), x_i^k, y_i^k, k) \quad (4)$$

where k is the index of the simulation step (e.g. in time domain simulations it corresponds to a simulation time, and for steady-state analysis $k=0$, i.e. there is only one step) and i is the index of the iteration of the numerical method. The map iteration stops whenever the difference between the variables at the i -th and at the $i+1$ -th iterations is below a certain threshold or if the number of iterations exceeds a maximum value (fixed a priori). F and G are functions that depends on f and g and on the numerical method. Examples of numerical methods for power flow analysis and time domain integration are given in [12].

It is important to notice that one never really solves (1)-(2), unless an explicit analytical solution is known. Actually, one has to solve (by means of a computer, generally) the map (3)-(4) which is substantially different from the original set of equations (1)-(2). The fact that the solution of (1)-(2) might be different from (3)-(4) and that the map can introduce its own stability issues is often overlooked not only by the students but also by the lecturers. The layers of modelling and simulation of physical systems is shown in Figure 1.

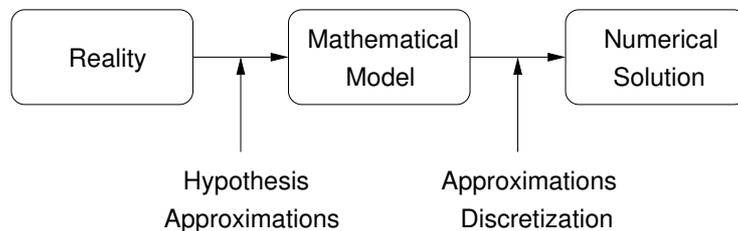


Figure 1. Layers of the modelling and simulation of physical systems.

4 EXAMPLES

This section discusses a variety of examples where the difference between (1)-(2) and (3)-(4) may lead to misunderstanding the results or incorrect conclusions. All examples considered in this section refer to the analysis of power systems and were encountered in the undergrad modules taught by the author, in particular the computer-based labs of module PSDC.

4.1 Power Flow Analysis and Maximum Loading Condition

The power flow analysis is one of the most important problems in power systems. Its solution allows initializing the set of differential-algebraic equations (1)-(2) as well as the solution of many other crucial problems (state estimation, optimal power flow and market analysis, fault analysis, etc.). It can be thus considered a fundamental step of any analysis involving power systems. From the mathematical point of view, it consists in the solution of a set of nonlinear algebraic equations, namely the vector y that solves:

$$0 = g(y) \quad (5)$$

where g express the mismatches of the power injections at the network buses and y are the voltage magnitude and phase angles at the network buses.

The literature on the solution of the power flow problem is vast and the interested reader can find a thorough discussion in [12]. Since (5) are nonlinear, it is not guaranteed that they have a solution, nor that, if a solution exists, one is able to find it. (5) might also have more than one solution and, *a priori*, one cannot know whether the solution that will be found by the numerical method will be the one that is sought. In most cases, in fact, of all possible solutions of (5), only one has a physical meaning and can be used in practice. Other solutions are called “spurious” or “false”, meaning that are not within the acceptable range of values that can be taken by bus voltages (e.g., a spurious solution can show negative voltage magnitudes).

There are several methods to solve (5) but the most relevant one is the Newton-Raphson method that transforms (5) into the following map:

$$y_{i+1} = y_i - g_y^{-1}(y_i)g(y_i) \quad (6)$$

where g_y is the Jacobian matrix of g . The main issue of the Newton-Raphson method is to choose a good *initial guess* y_0 to start the iterations. If (6) does not converge, there are two possibilities:

- The problem is unsolvable, e.g. (6) does not have a solution.
- Equations (6) have a solution but the initial guess y_0 does not allow reaching it.

In their reports, the students generally conclude that (5) does not have a solution and do not further investigate the problem. This is interesting because the conclusion above is misleading twice: first because the equations that are solved by the computer are actually not (5) but (6) and second because the possibility that the issue is the numerical method (in this case, its initialization) is not taken into account. It is also interesting to note that the choice of the initial guess is basically heuristic. In most cases the best choice is the so-called *flats start* (voltage magnitudes equal to 1 pu and phase angles equal to 0). While this choice works in the vast majority of the cases, one should never forget that it is just one of the infinite possible choices for the initial guess.

While there is no actual definitive method to distinguish between an unsolvable case and a bad initial guess, there are nevertheless several techniques that can at least provide some insight on the problem. Changing the numerical method (e.g. using a *robust* Newton method) can help. Robust algorithms to solve (6) are generally slower but, in some case, they tend to converge if not to the solution at least to the closest value on the feasibility region of the solutions. This does not guarantee that a solution of (6) does not actually exist but is a good clue that the problem is hard to solve. This information is still useful as it means that, even if not unsolvable, the system is not working at a feasible operating point. Another option is to vary the load consumption and observe if, for some loadin g condition the Newton-Raphson algorithm converges. If it does, it could be a clue that the problem with the original loading level is unsolvable. On the other hand, the idea of changing (e.g. randomly) the initial guess and try to obtain a solution is much less effective and is not recommended.

4.2 Limit Cycles vs Numerical Oscillations

A PSDC lab proposed to the students consists in determining the effect of automatic voltage regulation on the dynamic of a power system. The test case is the well-known IEEE 14-bus system adequately modified so that after a line outage the system becomes unstable and shows stationary undamped oscillations (stable limit cycle). This phenomenon is shown in Figure 2 and described in details in [12] and further discussed in [1].

The students almost never question the conclusion that it is the system to be unstable and that the oscillations are due to its behavior. This is again a consequence of the misplaced assumption that the computer solves (1)-(2). Since the computer is actually solving (3)-(4), another interpretation of the oscillations shown in Figure 2 is that the numerical method utilized to solve the time domain integration of the system is showing a numerical (spurious) oscillation.

In this case, it is actually quite simple to distinguish between the two issues. Numerical oscillations, in fact, will depend on the time step (parameter k in (3)-(4)). A smaller time step will lead to a higher frequency oscillation or, depending on the integration method, may also lead to eliminate completely the issue. A physical stable limit cycle, on the other hand, will not disappear by reducing the integration time step. On the contrary, a smaller time step will generally increase the resolution of the limit cycle and thus make the simulation more detailed.

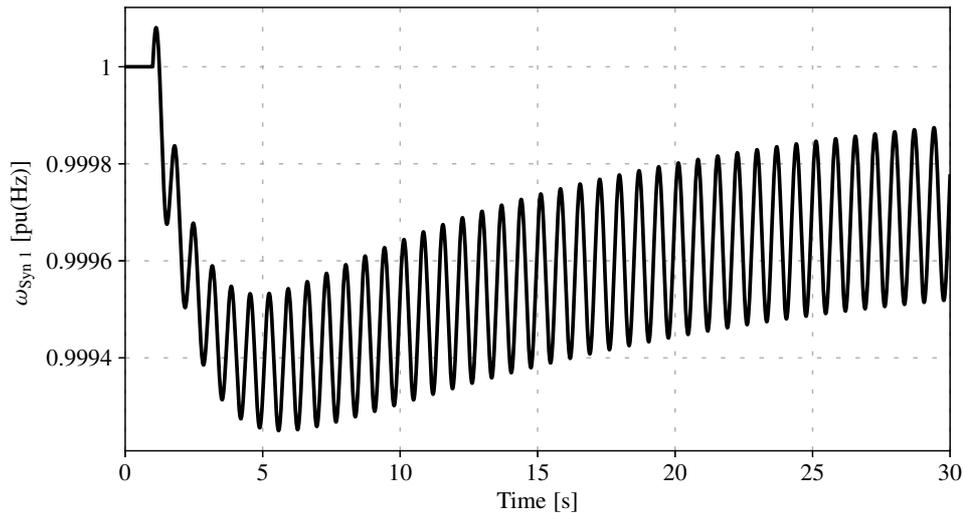


Figure 2. Undamped oscillations in a power system.

4.3 Physical vs Numerical Damping

A dual case with respect to the previous one, is the effect of some solvers to introduce a numerical damping to the trajectories of the system. The simulation results shown in Figure 2 are obtained using an implicit trapezoidal method with a time step of 0.02 s. Figure 3 shows the same case study but solved using an implicit Euler method with time step 0.02 s. The implicit Euler method is known to L-stable and, in some cases, also *hyper-stable*, meaning that it can lead stable trajectories even if the physical system is actually unstable. The implicit trapezoidal method is not prone to this issue and is thus to be preferred, but since it requires the knowledge of the previous step, it is not *self-starting* and, for this reason, the implicit Euler method is also commonly used.

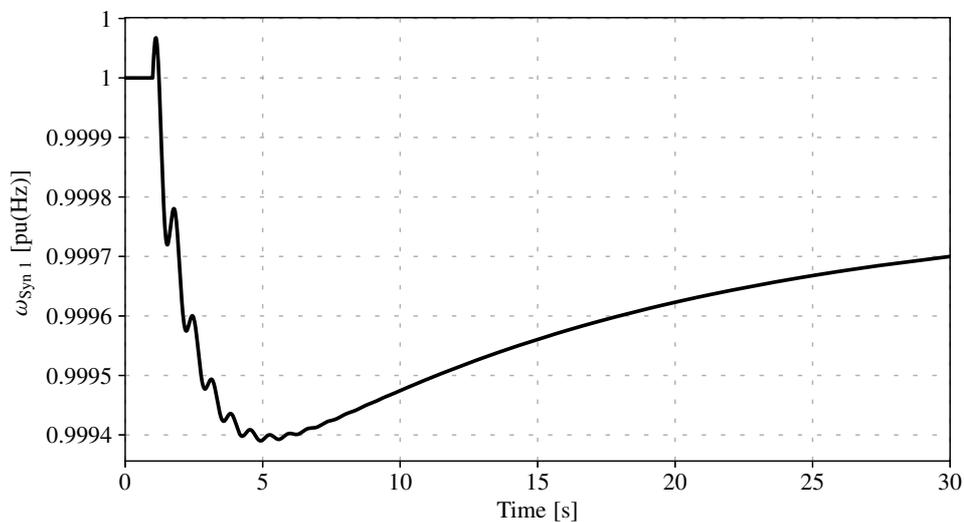


Figure 3. Numerically damped oscillations in a power system (Euler method with time step 0.02 s).

Also in this case, reducing the time step can solve the problem, provided that the students do not trust the solution and doublecheck it by using different time steps. Using a time step of 0.005 s, the implicit Euler method returns the trajectory shown in Figure 4. The oscillations are still damped but with a much lower damping than that shown in Figure 3. This can be possible only if the damping is due to the numerical integration scheme. As a matter of fact, the implicit Euler method with a time step of 0.001 s gives the same trajectory as shown in Figure 2.

The main conclusion of this analysis is that it is always a good idea to solve the same simulation at least twice using different integration schemes or the same scheme with different time steps. Numerical issues, either spurious oscillations or spurious damping tend to disappear or, at least, to change their period or ratio by changing the integration method or some of its key parameters. Again, the conclusion is that one has to focus on (3)-(4) rather than on (1)-(2) and, ultimately, never trust the first results obtained with the simulations. Students, however, tend to trust unconditionally the results provided by a computer and have to be carefully educated to think otherwise.

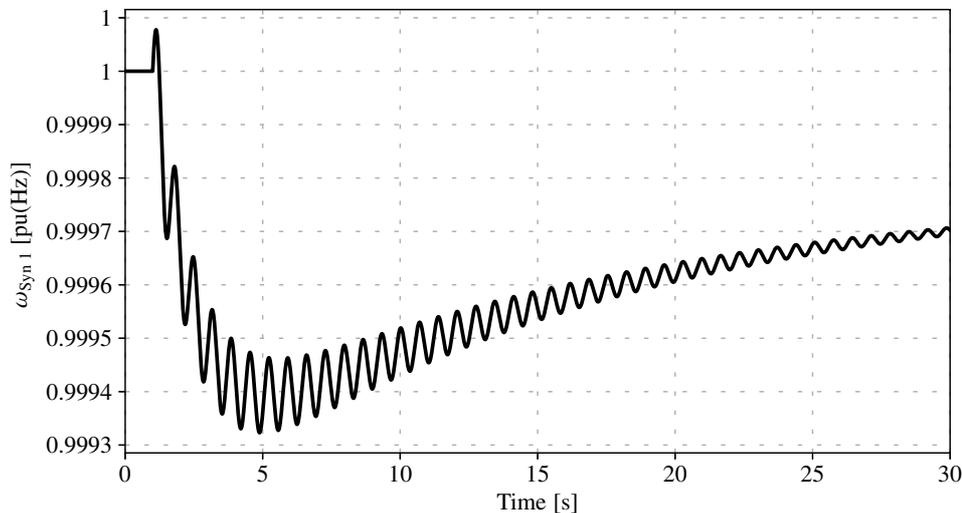


Figure 4. Numerically damped oscillations in a power system (Euler method with time step 0.005 s).

4.4 Trajectory Deviations due to System Model

A further source of confusion for the students is the fact that simulation results can change, even drastically, depending on the model of the system itself. Let us consider again the case study discussed in Sections 4.2 and 4.3. We have already concluded that the system does not have a stable equilibrium point and enters into a stable limit cycle, that leads to stationary undamped oscillations.

This conclusion is based on the several tests that we have solved using different solvers and different time steps. We can thus reasonably exclude that such oscillations are a property of (1)-(2) and not a spurious issue introduced by (3)-(4). However, while this conclusion appears reasonable and is also mathematically correct, we have not questioned at any time the model itself of the system. In other words, since any model (1)-(2) is the result of some assumptions and of an approximation of the reality, one may wonder whether the oscillations are due to the specific model used for the synchronous machines and/or their controllers or are actually independent from the model.

This question, in mathematical terms, can be reformulated as follows: is the limit cycle *robust*? i.e. will the limit cycle appear even if the equations of the system are perturbed? The study of the robustness of bifurcations (such as the unstable operating point that originates the limit cycle) is generally well beyond undergraduate module and is certainly beyond the learning outcomes of the module PSDC. The answer must thus be “qualitative.”

An easy way to test the robustness of the oscillations is to change the model of the synchronous machines. In the simulation results shown in Figure 2, the machines are modelled using a set of 6 differential equations that include mechanical variables and transient and sub-transient rotor fluxes. If we model the machines using a set of 4 differential equations, thus neglecting the dynamics of the sub-transient rotor fluxes, the resulting trajectory is shown in Figure 5.

When the students are shown the results of Figure 5, they tend to be quite confused. Typically, much more than when the lecturer discusses the issues of numerical integration and the effect of the time step. The concept that a numerical scheme can introduce issues, in fact, while unexpected is soon assimilated. On the other hand, the fact that the model itself, i.e. (1)-(2), is not written in stone and its choice can actually change completely the conclusion of a study cause deep bewilderment and concern. The students, in fact, start asking themselves what they should trust, the 6th or the 4th order model, or even whether they can trust any simulation result at all.

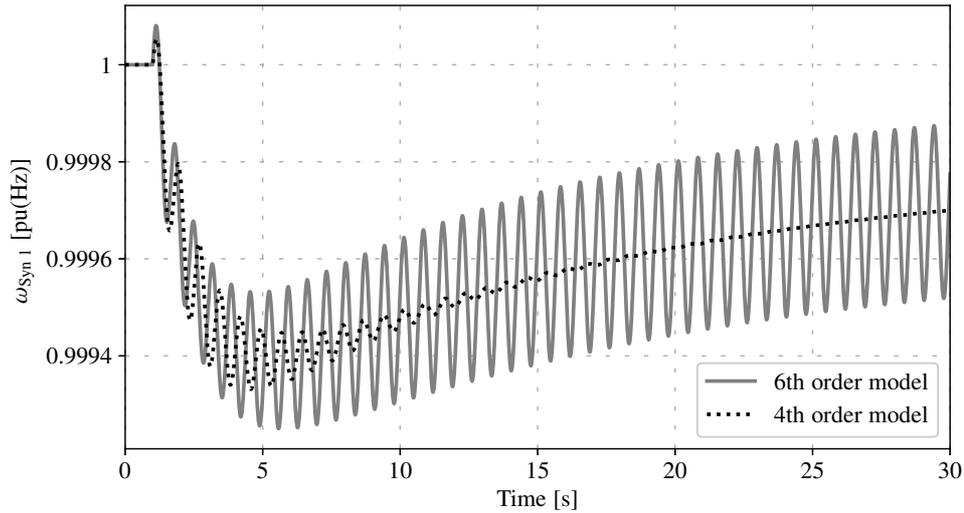


Figure 5. Comparison of the dynamic behaviour of different models of the same power system.

There is no easy answer to that question. The decision on which is the right model to use depends on many aspects, not least, the experience of the engineer that decides what is the right balance between fidelity and simplicity. Ultimately, *a model should be as simple as possible but not simpler than that.*¹ While, unfortunately, there is no fixed rule to decide what is *simple* and what is *too simple*, it is important that the students understand that the definition of the model is crucial and do not take any model as intrinsically correct or unquestionable.

4.5 Spurious Eigenvalues

This section completes the examples on numerical issues encountered by the students of the module PSDC with a discussion on eigenvalue analysis. This is a technique utilized to determine whether a given equilibrium point is stable or not. After the solution of the power flow analysis, the next step is to initialize all dynamic models, e.g. synchronous machines and their controllers. This initialization, however, does not guarantee that the system is stable. If it is not, there is no point to solve the time domain simulation because an unstable operating point cannot be reached by any trajectory and, thus, no physical system can ever remain at an unstable point.

The standard technique to determine whether an equilibrium point is stable or not is based on the first Lyapunov criterion, which requires to linearize the system at the equilibrium point and then calculate the eigenvalues of the state matrix of the system. Considering (1)-(2), an equilibrium point is given by the vectors x_0 and y_0 that satisfy the condition:

$$0 = f(x_0, y_0) \quad (7)$$

$$0 = g(x_0, y_0) \quad (8)$$

The linearized system is given by:

$$\Delta \dot{x} = f_x \Delta x + f_y \Delta y \quad (9)$$

$$0 = g_x \Delta x + g_y \Delta y \quad (10)$$

And, finally, the state matrix is given by:

$$A = f_x - f_y g_y^{-1} g_x \quad (11)$$

As for the time domain simulations, there are several methods to compute the eigenvalues of the state matrix. Moreover, the definition of the Jacobian matrices in (9)-(10) that lead to the definition of the

¹ This sentence is attributed to Albert Einstein.

state matrix can be affected by several numerical issues. In commercial software tools, the Jacobian matrices are calculated numerically, i.e. by computing finite differences of the differential-algebraic equations obtained perturbing one by one the state and algebraic variables around the equilibrium point. This procedure inevitably introduces numerical errors and can thus affect the calculation of the eigenvalues.

Even if the elements of the Jacobian matrices are calculated analytically, there are still several sources of numerical issues. A common one is how “zero” elements are treated in the implementation. There is, in fact, a difference, between elements that are structurally zero (for example, because a given function does not depend on a given variable) or are zero because the coefficient that multiplies a variable in a given equation is set to zero for that specific device. In software tools that preserve the structure of a matrix and exploit sparsity, e.g. [14], it is not uncommon to use a very small number, i.e. 10^{-24} , rather than zero, for elements that can be non-null depending on the data of the system.

Another reason why a zero element can be actually assigned a small value is for better conditioning the Jacobian matrix. Sometime, in fact, adding a small number to the diagonal elements of the state matrix improves the stability of the numerical method that it is used to calculate the eigenvalues.

Finally, different implementations of the same solver can lead to different results. For example, the implementation of LAPACK, which provides a robust solver for unsymmetrical dense eigenvalue problems, can sometime provide different results if one uses the Intel implementation or the GNU public domain implementation.

All the issues above can lead to spurious eigenvalues. It is not uncommon that, if the Jacobian matrices are not properly conditioned (this happens for example, if the network is modelled with lossless branches and synchronous machines have null damping), that the eigenvalue analysis can return extremely big or extremely small eigenvalues. Now, if the big eigenvalues are negative, generally, they are not noticed (as the system is stable anyway). But huge positive eigenvalues create always concern in the students who know that, according to the Lyapunov first stability criterion, an operating point is unstable if at least one eigenvalue is positive.

Nevertheless, this conclusion is incorrect. Eigenvalues have the units of frequencies (i.e., the inverse of seconds). A very big positive eigenvalue indicates that the system has some form of energy that varies very fast and is unstable. If no dynamic of the system is as fast as the inverse of an eigenvalue, that eigenvalue must be spurious. For example, in an electro-mechanical power system model, the fastest dynamic that is taken into account is of the order of 0.001 s, which leads to 1000 Hz. Any eigenvalue whose absolute value is much bigger than 1000 (positive or negative) is physically impossible. Since numerical spurious eigenvalues tend to be of the order of 10^6 , but generally much bigger than that, there is no doubt that they are originated by numerical issues.

Students, not only undergraduate but also PhD ones, have great difficulty not to trust the results of the eigenvalue analysis. Again, this behaviour is mainly due to fail to appreciate the difference between a mathematical model and numerical solvers and the fact that numerical solvers utilize (3)-(4) rather than (1)-(2). In this case, at least, discriminating between physical and numerical instability is relatively simple. Using a different condition number for the diagonal elements of the state matrix eliminates the problem. An even easier solution is to solve a time domain simulation with no disturbances. If the system is actually unstable, any small numerical deviations with respect to the equilibrium point will lead the trajectories to diverge, the faster the higher are the positive eigenvalues.

5 CONCLUSIONS

The paper discusses the difference between physical and numerical instability. The confusion between the two originates by the methodological approach of the study of physical systems, which consists of two steps: first the definition of a mathematical model of reality and then the solution of a discrete map, generally by means of iterative algorithms implemented in a computer. The “trust” that students put on computers often makes difficult for them to question the results of a simulation and doublecheck such results using different algorithms or modifying the parameters of the simulation. Even more concerning for the students is the fact that the model *per se* can be questioned and that different models can lead to significantly different conclusions. While modelling is more an art than a well-defined procedure, it is still important that the students develop a critical approach and do not passively accept either the models proposed in the books or the solutions provided by computers. Future work will focus on developing an ad hoc learning process to teach the students the art of modelling and gently introducing them to the idiosyncrasies of numerical analysis.

ACKNOWLEDGMENTS

This work is funded by UCD OBRSS scheme and by Science Foundation Ireland, under the Investigator Programme, Grant No. SFI/15/IA/3074.

REFERENCES

- [1] F. Milano, *Thinking Nonlinearly: Experience in Teaching Power System Stability Analysis in Engineering Programmes*, EDULEARN 2016, Barcelona, Spain, 4-6 July 2016.
- [2] F. Milano, *Problem Solving through Limit-case Analysis: Experience in Electrical Engineering Programmes*, EDULEARN 2016, Barcelona, Spain, 4-6 July 2016.
- [3] F. Milano, *A Systematic Method for Failure Diagnosis in Electrical Machine Lab Activities*, ICERI 2011, Madrid, Spain, 14-16 November 2011.
- [4] L. Vanfretti, and F. Milano, *Application of the PSAT, an Open Source Software, for Educational and Research Purposes*, IEEE PES General Meeting, Tampa, USA, 24-28 June 2007.
- [5] F. Milano, L. Vanfretti, J. C. Morataya, *An Open Source Power System Virtual Laboratory: The PSAT Case and Experience*, IEEE Transactions on Education, Vol. 51, No. 1, pp. 17-23, February 2008.
- [6] F. Milano, *Experience of Unix Terminal-based Labs for Undergraduate Modules on Power System Analysis*, EDULEARN14, Barcelona, Spain, 7-9 July 2014.
- [7] F. Milano, *Experience Teaching Electrical Energy Systems to Non-electrical Engineering Students*, EDULEARN 2017, Barcelona, Spain, 3-5 July 2017.
- [8] F. Milano, *Methodological and Didactic Challenges of Teaching Electrical Energy Systems to Second Year Students*, EDULEARN 2017, Barcelona, Spain, 3-5 July 2017.
- [9] P. Kundur, *Power System Stability and Control*. New York: McGraw- Hill, 1994.
- [10] P. M. Anderson, A. A. Fouad, *Power System Control and Stability*, 2nd Edition, IEEE Press, John Wiley Interscience, Hoboken, 2003.
- [11] P. W. Pai, M. A. Sauer, *Power System Dynamics and Stability*, Prentice Hall, Upper Saddle River, 1998.
- [12] F. Milano, *Power System Modelling and Scripting*. London: Springer-Verlag, 2010.
- [13] F. Milano, G. M. Jónsdóttir, *Jupyter Notebooks for Computer-based Laboratories on Power System Dynamics and Control*, EDULEARN 2018, Palma de Mallorca, Spain, 2-4 July 2018.
- [14] F. Milano, *A Python-based Software Tool for Power System Analysis*, IEEE PES General Meeting, Vancouver, Canada, 21-25 July 2013.