PSpice as a Simulation Tool for Control System Engineering Course

Mark Ho
School of Electrical and Information Engineering
University of South Australia, Mawson Lakes, Australia
Mark.ho@unisa.edu.au

***Abstract**: Control system engineering courses utilise simulation tools such as MATLAB to provide a range of functions for time and frequency response analysis. Students in their studies will use function such as step and rlocus to study the performance of a control system given its open-loop transfer function. However it is not obvious how theory can be extended to the implementation of the system and the controller. In this paper we proposed Orcad PSpice as an additional simulation tool to help student understand the link between MATLAB simulation results and its physical implementation. We describe how student can model a stable second order transfer function as a simple RLC circuit. How negative feedback with root locus analysis can be used to affect its time response for a step input and study the effect of system stability. Student will perform the modelling and analysis in MATLAB and verify the simulation results using PSpice circuit simulation.*

Introduction

Control systems engineering is a topic where all undergraduate electrical and electronic engineering degree student will encounter in their degree. A control system, as defined in Nise (2008), is "*subsystems and processes (or plants) assembled for the purpose of obtaining a desired output with the desired performance, given a specified input*". In the undergraduate course level students will be exposed to concept such as system transfer function, poles and zeros of a system, time response of the system, root locus techniques etc. Traditionally MATLAB has been used as the main tool for simulation and analysis of control system in the undergraduate level due to the excellent support provided by the control system toolbox. However often times MATLAB can only provide an abstract level of system simulation limit only to transfer function which some student may find hard to relate to their background as a electrical/electronic engineer.

Often there are analog model between physical system and electrical system therefore it is only natural that an electronic circuit simulation tool can be used to simulate a physical system (for example see Burke, T.G. & Schiller, D.R. 2003 and chapter 2 of Nise 2008). In this case PSpice is the perfect tools since most electrical and electronic students will have exposure to Orcad PSpice (by Cadence Design Systems, Inc.) simulation package or other software with identical functionality. The advantage of Orcad PSpice is that it has a nice graphical interface for circuit construction/simulations and a lite version of the software is often included as part of the text book (for example Orcad PSpice 9.2 Lite is included with Sedra, A. and Smith K. 2004) or available for download from the company's website.

Other authors have similar approach but were in the context of teaching power electronics for undergraduate level (for example Bellur, Dakshina M. & Kazimierczuk, Marian K., 2008 and Zhao Tao & Wang Qunjing, 2005). While others have use MATLAB and PSpice as a modelling tool for power systems and electronics (such as Hoang Le-Huy & Sybille, G. 2000) as well as a teaching tool for communication systems (for example McKinney, M. 2006).

In this paper we propose a way to use PSpice to supplement MATLAB as a teaching tool for control systems simulation. First we will briefly summaries the range of topics to be covered in a control system course. Then we will explain how a control systems transfer function can be modelled as a circuit in PSpice and how we can modify the circuit to reflect any changes in system response. Lastly we will look at how a PSpice circuit can be used to verify root locus results from MATLAB.

Syllabus of undergraduate control systems course

We will briefly go through some ideas student studying control systems will most likely encounter in their studies. This will form the basis on why MATLAB simulation by itself does not entirely show how control system theory can be related to electrical/electronic engineering. Two major topics that will be covered in control system are time response of a first/second order system and root locus technique (other topics being state-space variable, frequency response method, stability analysis and PID controller).

For time response simulation of first/second order system the *tf()* function and *step()* function of the control system toolbox in MATLAB can be very useful. The *tf()* function define the 's' domain transfer function of a system, while the *step()* function simulate the step response of the transfer function defined by *tf()*. If we apply negative feedback to a transfer function the *feedback()* function can be used to find the resultant transfer function if feedback is applied. The poles and zeros location of the transfer function can be determined by MATLAB with *pole()* and *zero()* function.

For root locus technique, which is a graphical techniques where by plotting the location of the opened-loop poles and zeros and applying a few simple rules, we can predict the location of the closed-loop poles as the opened-loop gain (*K*) is varied between 0 and +∞. The *rlocus()* function can be used to plot the root locus. Each individual point on the root locus and the exact *K* value used to achieve that pole location can be determined by MATLAB.

MATLAB as a simulation tool

As we can see MATLAB by itself is a very useful tool for control system simulation and analysis. The control tool box of MATLAB provides numerous features for student to easily experiment and learn from their experience.

However, by limiting all function to be simply transfer function plot students often cannot visualise how the results can be related to an actual system implementation. Since control system theory spans many different disciplines from mechanical to electrical system therefore there are numerous ways we can demonstrate the gap between theory and implementation. Given the background of electronic / electrical engineering students it is important that the demonstration is something they can either relate to or have prior experience with throughout their undergraduate studies. The challenge is how to link control system theory with basic circuit theory such that student with electrical and electronic engineering background are familiar with.

PSpice simulation of second order system

Here we propose a way to bridge the gap between MATLAB simulation and circuit simulation by using PSpice. To simulate the step response of the system traditional approach is to use MATLAB built in function such as *step* to simulate the system based on its s-domain transfer function. A similar response can also be obtained in a circuit simulation if we can find an equivalent second order transfer function. A typical second order transfer function can be expressed in the following form:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Where ω_n is defined as the natural un-damped frequency and ζ is the damping ratio. Such a transfer function can be constructed using an RLC circuit voltage gain response in the Laplace transform domain. The derivation of the transfer function is very straight forward can be found in page 46-46 of Nise (2008).

$$\frac{Vc(s)}{V(s)} = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Equating the two transfer function we can see that the natural un-damped frequency ω_n of the RLC circuit can be determined by $\sqrt{\frac{1}{LC}}$ and the damping ratio ζ can be determined by $\left(R\sqrt{\frac{C}{L}}\right)/2$. Thereby varying the value or *R*, *L* and *C* the dynamic time response of the system can be easily changed.

A comparison of PSpice circuit simulation and MATLAB step function results for *R*=*L*=*C*=1 is given in **Figure 1**. As can be seen from the results PSpice simulation obtain the same transient and steady state response as compared to MATLAB *step()* function results. Given that the student can freely change the value of R, L and C in the circuit they experiment with different ω_n and ζ value, test their understanding using MATLAB *step()* function and verify their idea using PSpice simulation. The step response is implemented by a DC voltage source with magnitude of 1V. Note that PSpice simulation profile must be setup with "*skip the initial transient bias point calculation*" disabled to successfully simulate the transient response.

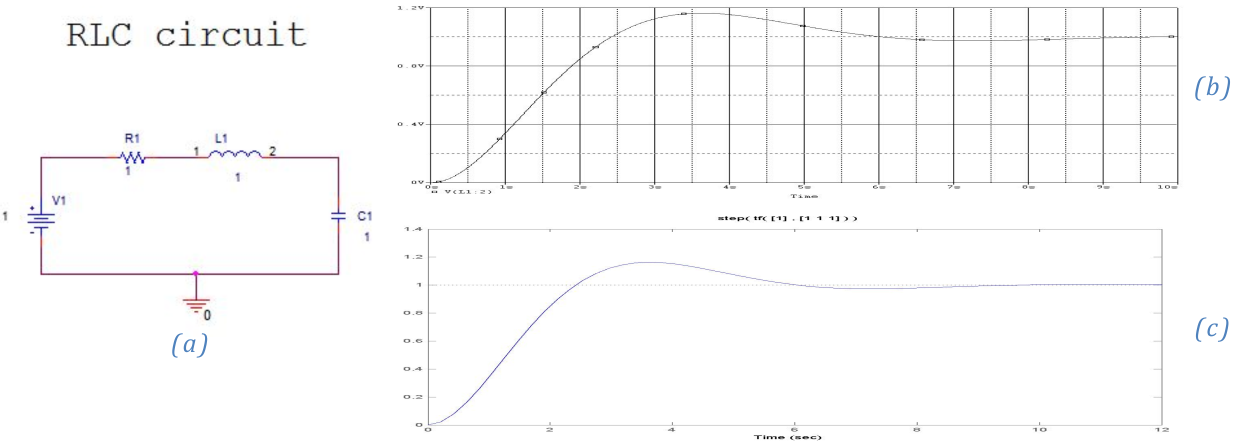

Figure 1: RLC circuit and its step response in PSpice simulation (a) PSpice RLC circuit (b) PSpice circuit simulation, (c) MATLAB step function plot.

Root locus analysis and simulation

Once students are familiar with transfer function and its relevant time response the next major topic is root locus technique for negative feedback systems. To simplify the analysis it is often assume that the feedback transfer function *H*=1 which reduces the closed loop transfer function to:

$$\frac{C(s)}{R(s)} = \frac{KG(s)P(s)}{1 + KG(s)P(s)}$$

where *G(s)* is the open looped transfer function, *K* is the open loop gain and *P(s)* is the transfer function of the single pole system. If we apply feedback to our second order transfer function with the additional single pole system the resultant transfer function is:

$$\frac{C(s)}{R(s)} = \frac{KP(s)\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{2}{LC}}$$

Figure 2 show the negative feedback control system for root locus analysis.

Figure 3 shows the PSpice circuit model that implements the system given in **Figure 2**. Negative feedback is implemented with a difference amplifier between two input, the 1V input dc voltage and feedback signal. For unity feedback a non-inverting unity gain amplifier is used to buffer the output of the second order system and the input of the difference amplifier. The voltage transfer function of the single pole system is implemented using an op-amp with transfer function:

$$\frac{V_O}{V_I} = -\frac{1}{s + \frac{1}{R_2 C_1}} \cdot \frac{1}{R_1 C_1}$$

Where R_2 and C_1 is the feedback resistance and capacitance respectively. The pole location can be changed by choosing the value of R_2 that correspond to the pole location by using the relationship: $R_2 = \frac{1}{pole \cdot C_1}$. The open-loop gain value K is determined by the ratio between the feedback resistance and input resistance ($-R_f/R_{in}$) of the '*K*' amplifier.

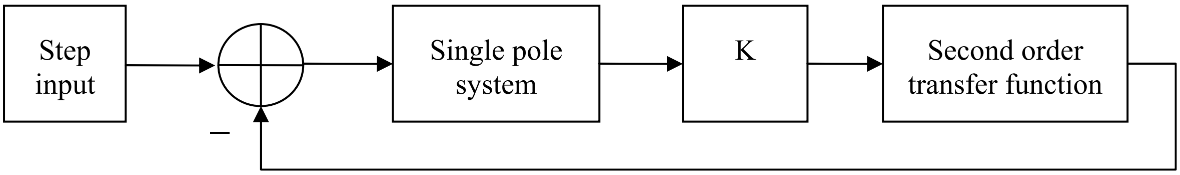

Figure 2: A third order negative feedback control system.

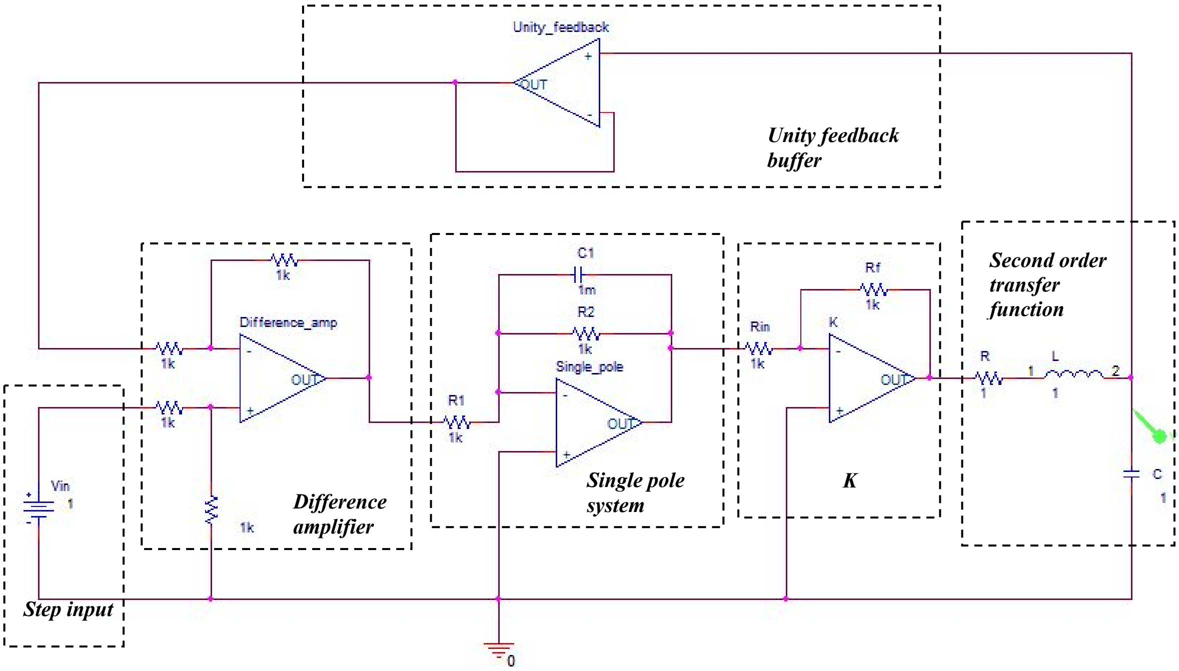

Figure 3: PSpice model of three poles negative feedback control system given in Figure 2.

Since after feedback is applied the system transfer function is changed therefore the overall system step response with feedback will be affected. Root locus technique provide us a way to predict the step response by plotting the poles and zeros of *G(s)P(s)* for various values of *K*. For root locus analysis we want students to understand how the open loop gain *K* can affect the stability of the system. For a three poles system with $G(s) = \frac{1}{s^2+s+1}$ and $P(s) = \frac{1}{s+1}$ the root locus is given in **Figure 4**.

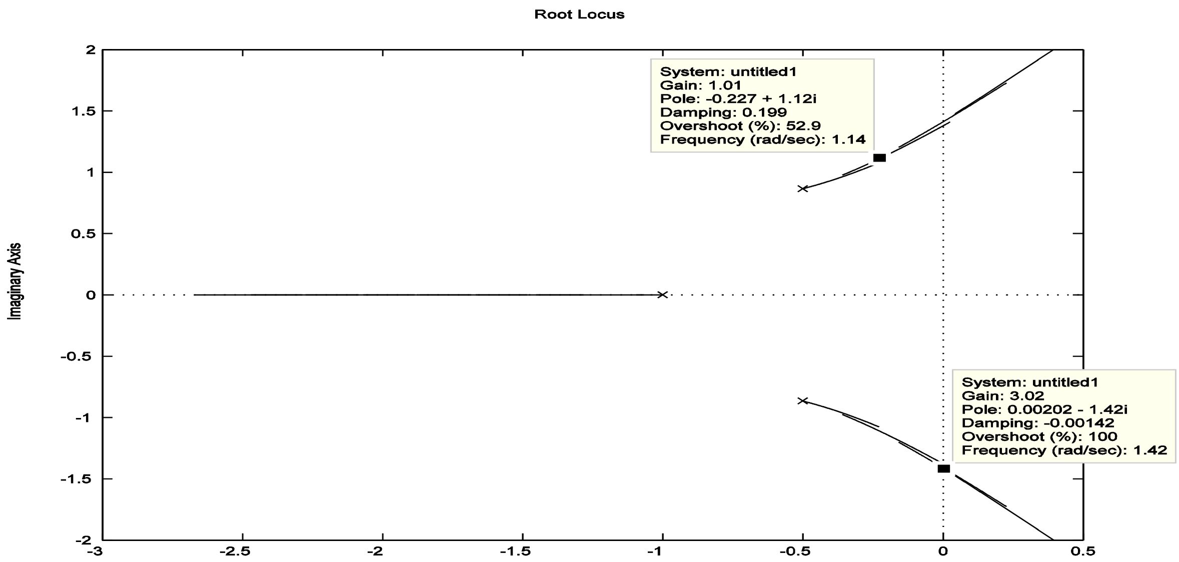

Figure 4: root locus of a three poles system.

From the root locus we can see that for K<3 the system poles is on the left hand plane which indicate a stable system while for any value above K>3 the system is unstable. When K=3 the system is marginally stable and the output will oscillate since the exponential decay is zero (zero real part). To show that PSpice simulation can obtain identical results as MATLAB *step()* function we change the value of R_f to 1kΩ, 3kΩ and 4kΩ to reflect the open-loop gain K value of 1, 3 and 4. **Figure 5** shows the simulation output of MATLAB and PSpice with three different values of K. As can be seen from the output both sets of results are identical which indicates that PSpice circuit model do accurately model the transfer function defined in MATLAB. By changing the R_2 value we can also see how the single real pole can affect the system stability

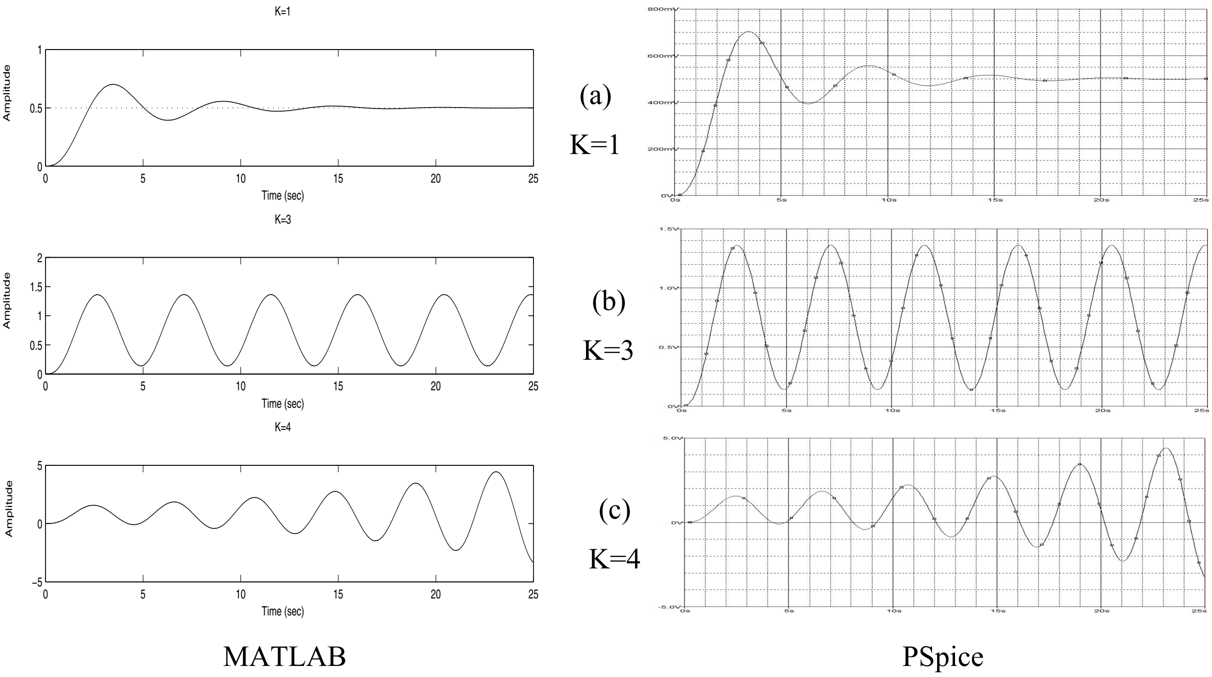

Figure 5: Simulation results comparison between MATLAB and PSpice for (a) K=1 (b) K=3 (c) K=4.

Note that PSpice simulation is not without its own pitfall; when the circuit simulation involves an unstable system where the output is unbounded PSpice will have convergence problem unless the simulation time is restricted to a very short duration of the output signal. Although such a scenario is rare and only occur if an unstable system is simulated; student needs to be aware of this fact to avoid simulation error.

Conclusion

In this paper we describe a model for PSpice to be used as a simulation tool and how it can be used to complement the *step()* function results from MATLAB. We show that by constructing the circuit in PSpice according to the transfer function defined in MATLAB we can reproduce the same output as MATLAB simulation. This idea can be extended further in future work where a PID controller implemention with op-amps can be used as a way to demonstrate the working of a compensator and controller tuning. While the frequency sweep function of PSpice can be used for frequency response analysis and assist in the construction of Nyquist plot.

References

Bellur, Dakshina M. & Kazimierczuk, Marian K. (2008). PSpice and MATLAB Applications in Teaching Power Electronics to Graduate Students at Wright State University. *Proceedings of the 2008 ASEE North Central Section Conference*.

Burke, T.G. & Schiller, D.R. (2003). Using PSPICE for electrical heat analysis. IEEE Potentials (pp 35-38).

Hoang Le-Huy & Sybille, G. (2000). MATLAB/Simulink and PSpice as modelling tools for power systems and power electronics. *IEEE Power Engineering Society Summer Meeting* (pp 766-767). Seattle, WA, USA.

McKinney, M. (2006). Using PSpice Behavior Modeling to Teach Communications Topics. *36th Annual Frontiers in Education Conference* (pp 9 - 14). San Diego, CA, USA.

Nise, N. S. (2008). *Control System Engineering*. John Wiley & Sons, Inc.

Sedra, A. and Smith K. (2004). *Microelectronic Circuits*, *5th edition*. Oxford Press.

Zhao Tao & Wang Qunjing. (2005). Application of MATLAB/SIMULINK and PSPICE Simulation in Teaching Power Electronics and Electric Drive System. *Proceedings of the Eighth International Conference on Electrical Machines and Systems* (pp 2037 – 2041). Nanjing, China.

Acknowledgements

The author would like to thank Dr Wai Chan for his valuable help throughout the development of this paper.