EE302: Root locus, Nyquist plot, Bode plots, Matlab/Octave/Scilab

Please write about corrections/typos to belur@iitb.ac.in. Document date: 26th March 2020

This note has information about Matlab, Octave and Scilab for use in control. Some comparison and some basic commands (for root locus, etc) is also included in this note.
All three: Matlab, Octave and Scilab are numerical computational packages with many advanced features/capabilities too. Several basic/complicated plots are possible in all three. Accuracy and speed are exactly same for all important numerical computation (like eigenvalues, roots of polynomials, control theory computation).
Vector, matrix, array, for-loop, while-loop, syntax is exactly the same.
Our recommendation: choose any one of: Octave/Matlab (pair) or Scilab: and don't keep debating and wasting valuable time.
All three packages are available in Windows, Linux (and most likely in Mac). Read below about pros/cons and then the syntax for Octave and Scilab for root-locus, Bode plot, Nyquist plot.

Matlab
Pros: Excellent proprietary package: has many toolboxes. Including Simulink. Simulink is pretty unmatched. Nearest FOSS alternatives are not as good (as far as advanced users' opinion goes).
Cons: Price: exorbitant. Matlab skills make us very very dependent on our company/institution renewing this expensive package/toolboxes. Increasingly, FOSS packages are making instututes debate abour renewing Matlab (and thus making your Matlab capabilities redundant and hence you vulnerable). Do not use pirated version. Institute gets into trouble by these bad habits/mal-practices (and hence Institute would not bail out a defaulter in this matter. Each one of you has signed the IT-policy of the institute and hence you are subject to stringent laws about pirated package usage. Please do not take chances with pirated packages.)
See later below about how it is very easy to manage with Octave/Scilab for the requirements in this course (and of course for advanced things too).

Octave
Pros: For all basic (and not so advanced) things: Octave works exactly like Matlab on the code that is written for Matlab. (Octave's strength is that Octave aims to run your Matlab code and give output like Matlab would.)
Cons: All toolboxes of Matlab do not yet have Octave versions. (Most developers of Octave spend time voluntarily and guaranteeing Matlab's output needs effort.) Simulink-alternative not yet in Octave.

Scilab
Pros: Scilab has many advanced toolboxes (control systems, signal processing, statistics, image processing, video processing, optimization and many more). Simulink-equivalent (called Xcos) is reasonably good.
Cons: Syntax not exactly same like Matlab. (But very similar for matrix/vector/basic-operations purposes.)
Control systems specific commands There is a lot of help on all basic requirements (including control) on the Internet and within the package. We are listing only the basic commands. There are excellent spoken tutorials (on www.spoken-tutorial.org) on Scilab: basic and advanced.

Octave (This would work for Matlab too)

System with transfer function $G(s) = \frac{s+4}{s^2-5s+6}$ (say) is defined by using

```
 % Octave/Matlab *comment* follows a % on that line
 % Commands/variables are case-sensitive:  be careful
 % Files are called file.m files
pkg load control % AFTER installing control package in octave
n=[1 4] % numerator polynomial coefficients
d=[1 -5 6] % denominator polynomial coefficients
sys=tf(n,d) % one way of defining a system called sys (using 'tf')
rlocus(sys) % plots root locus
bode(sys) % plots Bode plot (though system is unstable:  interpret cautiously!!
nyquist(sys) % Nyquist plot
% Considering closing the figure before next plot (else, it 'super-imposes'?)
```

Scilab

```
 //Scilab *comment* follows //on that line
 //Commands/variables are case-sensitive:  be careful
 //Files are called file.sci (or file.sce) files
 s=poly(0,'s') //Define s as the polynomial in the variable 's' with root 0
 s=%s //alternatively, some *built-in* items, like %pi, %e (for 2.718),
//More built-ins:  %i,%j (for sqrt(-1)),%eps, etc
 n = s+4 //There are other ways too of defining polynomial n (see poly)
 d = s^2 -5 * s + 6 //denominator
 evans(n,d) //for rootlocus:  named after its inventor:  Walter Evans
 sys=syslin('c',n,d) //continuous time system:  2 arguments means num, den
 evans(sys) //evans works with both:  num/den (2 arguments) or sys (1 argument)
 bode(sys) //plots Bode plot (though system is unstable:  interpret cautiously!!
 nyquist(sys) //
```