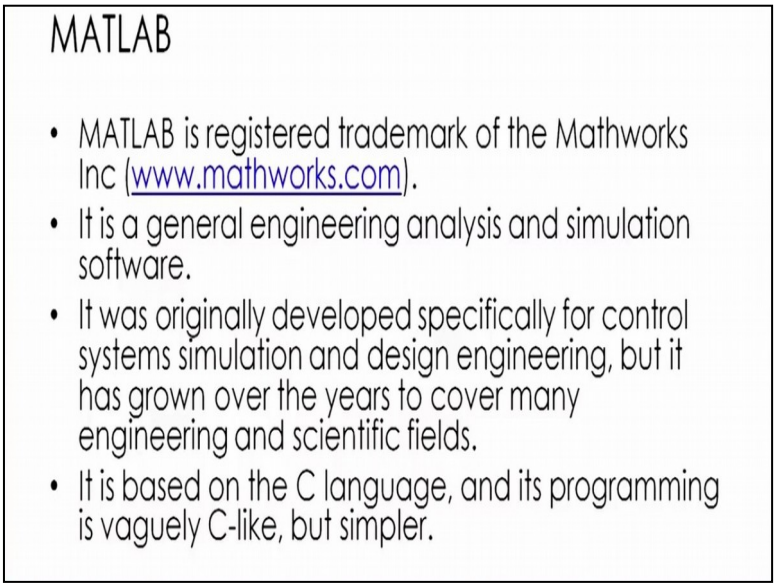
Modeling and Simulation of Dynamic Systems
Dr. Pushparaj Mani Pathak
Department of Mechanical and Industrial Engineering
Indian Institute of Technology - Roorkee

Lecture - 5
MATLAB as a Simulation Tool

I welcome you all on this lecture on as a simulation tool, which is a sub module for the course on modelling and simulation of dynamic system. The thing which I am going to talk today in this lecture, is a very general term and I will be talking of the general features of MATLAB and at the end of the presentation I will take an example of a second order system, simple spring mass damper system and we will see how we can simulate that system using mat lab.

During this course wherever the simulation exercises will be there often we will be using MATLAB as a tool either directly MATLAB or we will be using the simulation to simulate our system during the future lectures in this course.

(Refer Slide Time: 01:41)

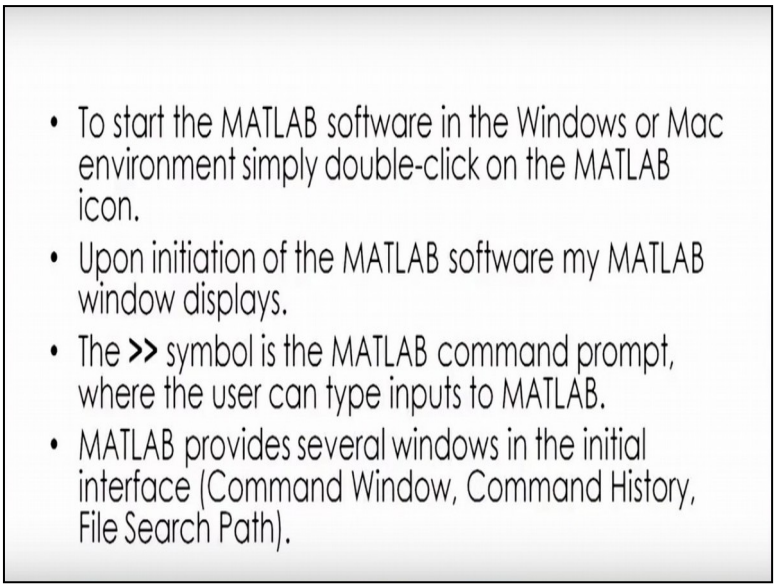


So, MATLAB is very popular software used by academia engineers for simulation purpose it basically means mathematical laboratory. It is a registered trademark of math work. So, you can get further details about it at www.mathswork.com. It is general engineering analysis and simulation software. It was originally developed specifically for control systems simulation and design engineering, but it has grown over the years to cover many engineering and scientific fields.

These days we should be finding many tool boxes which are associated with the mat lab, and with the help of these tool boxes you can do your computation simulations very easily. As I said, it was essentially developed specifically for control system simulations to see the behavior of the system and how the system behaves with changing the control parameters. Fine, so this is what the intention for the development of this software was. It is based on the c language and its programming is vaguely c like but of course it is much simpler.

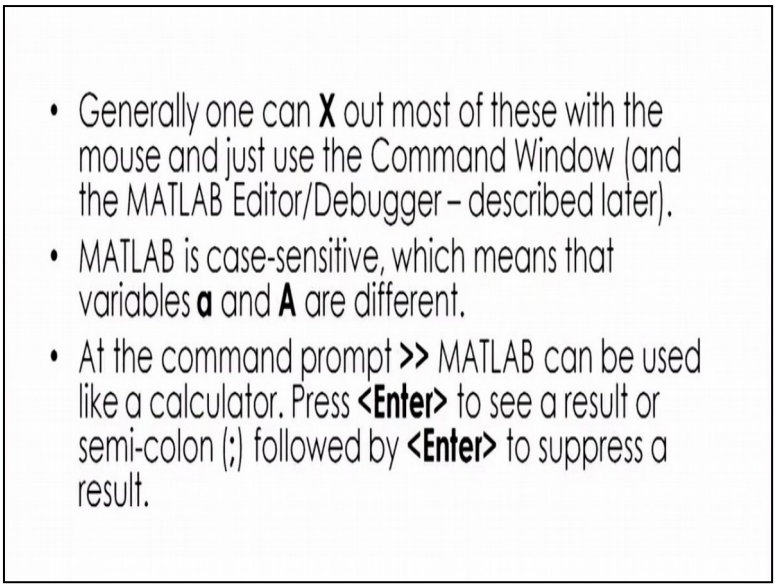
To start the MATLAB software in the windows or Mac environment; one needs to simply double click on the MATLAB icon and upon initiation of the MATLAB software, MATLAB window display occurs. The 2 arrow symbol is the MATLAB command prompt and here the user can type inputs to the mat lab. So, as soon as MATLAB starts you will be getting these symbols, which is the command prompt and where the user can input the commands for mat lab.

(Refer Slide Time: 04:06)



Now, MATLAB provides several windows in the initial interface we have the command window, command history, file search path and so on. Generally, one can cross out most of these with the mouse and just use the command window.

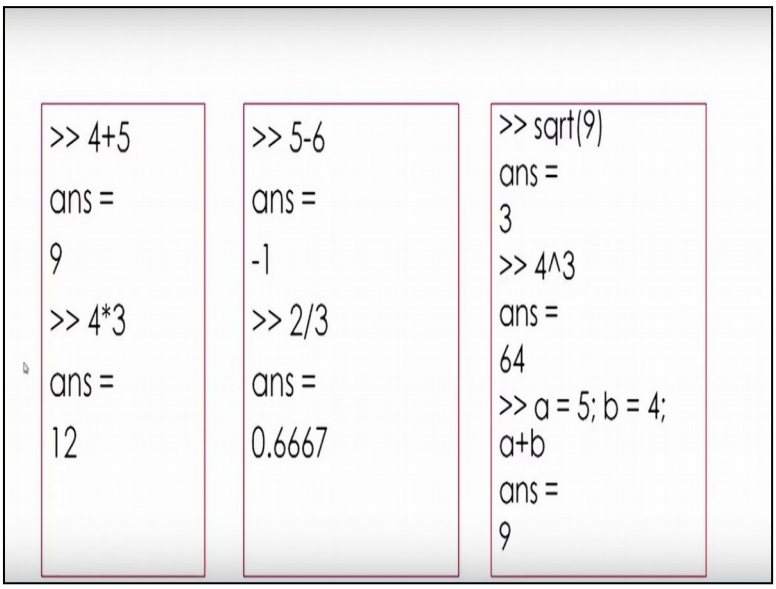
(Refer Slide Time: 04:20)



Here the MATLAB editor or debugger which we will be describing later. The MATLAB is case sensitive which means that the variables, small a and capital A are different. At the command prompt MATLAB can be simply used as a calculator. So, one can just press enter to see a result or if you want to suppress some results.

If, you do not want to see it then you can just use the semi colon followed by enter to suppress a result. So this is what we can do at the command prompt itself. Here are some basic although very primitive examples but just giving you an idea

(Refer Slide Time: 05:20)



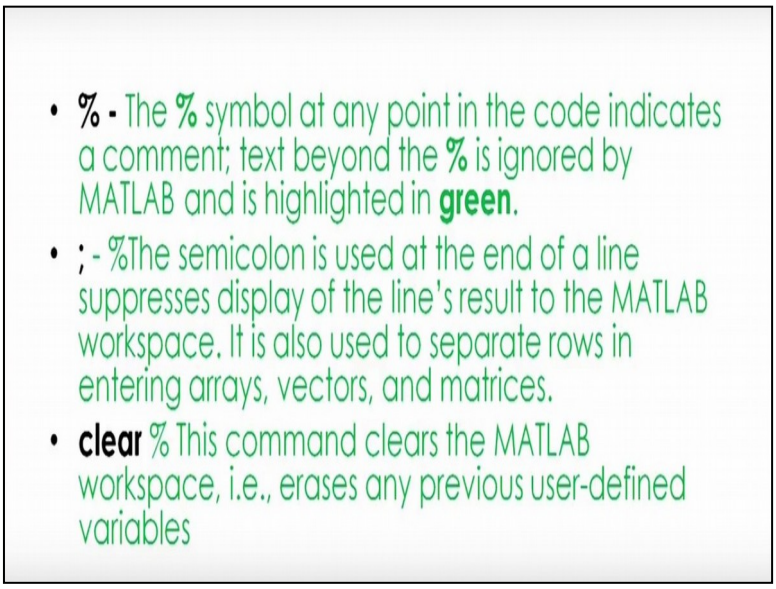
4+5 you will be getting an answer as 9. So, you can have addition you can do the multiplication you can do the subtraction and division just like what we do in our calculators. So this way it can be done then there is a function sqrt from which you can find out by square

root or you can have the power, The power is there so you can evaluate the 4 Power 3 64 or some simple algebric evaluation also can be done.

For example, if I define a is equal to 5 and b is equal to 4, a+b will return me the 9 so this way using the command prompt of the MATLAB we can do the simple calculations. Now, as when we do programming in any of the programming languages Fortran or in c or c plus to make your program readable we use to pat the comment statements, so that the user who is going to use that programme is guided by what is actually happening in the programme.

So likewise in the MATLAB also, we can put a comment statement and that is usually put by putting the percentage symbol at the beginning.

(Refer Slide Time: 07:00)



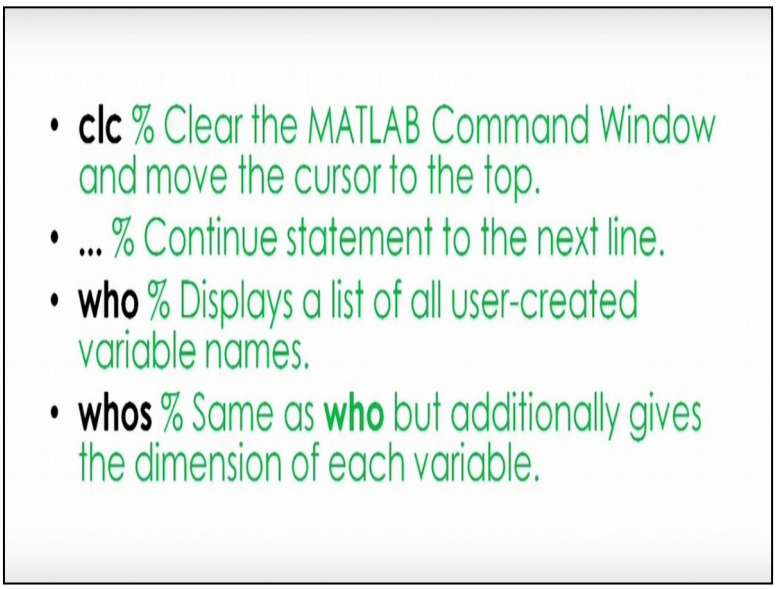
Here the percentage symbol at any point in the code indicates a comment and text beyond the percentage symbol is not evaluated or it is ignored by the MATLAB and it is highlighted in the green just to indicate that it is a comment statement. As I said, in the last line the semi colon is used at the end of the line to suppress the display of the line's result to the MATLAB work space. It is also used to separate rows or in entering arrays vectors and matrices.

So, these things may be in our lectures. I will be talking about arrays vectors and matrices also. Likewise we can the clear command or this command clears the MATLAB work space that is, it erases any previous user defined variables. As I said, there are many books written on using the MATLAB or how to use the MATLAB for different type of computation

activities. My aim here for this 30 minutes presentation is just to give you a brief over view of the MATLAB and how we can use the MATLAB for system simulation purpose.

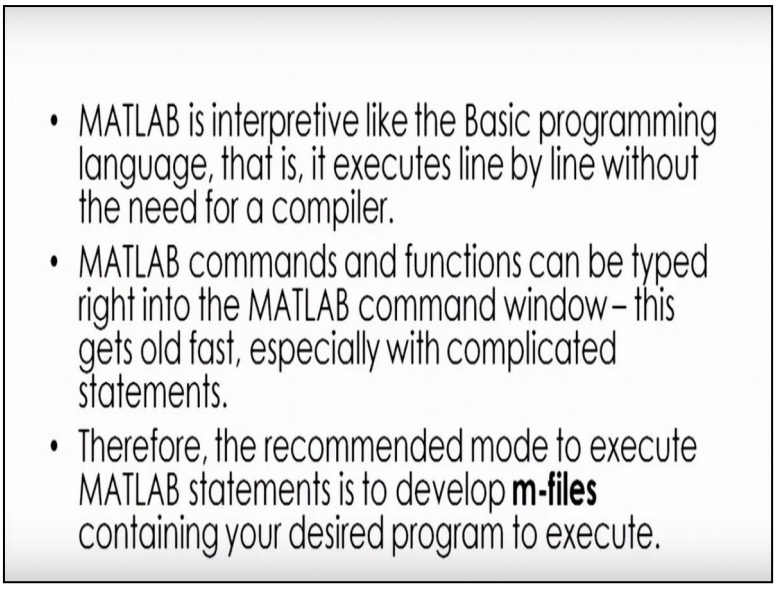
So, it is only to make you start using the MATLAB so, that is the sole purpose of this lecture. Likewise there are some other important commands.

(Refer Slide Time: 08:50)



clc it clears the MATLAB command window and moves the cursor at the top then the three dots this is basically use to continue statement to the next line, then we have a who command which essentially displays the list of all user created variable names and whose tells you it is the same as basically but additionally gives the dimension of each of the variable is interpretive like the basic programming language that is, it executes line by line without the need for a compiler.

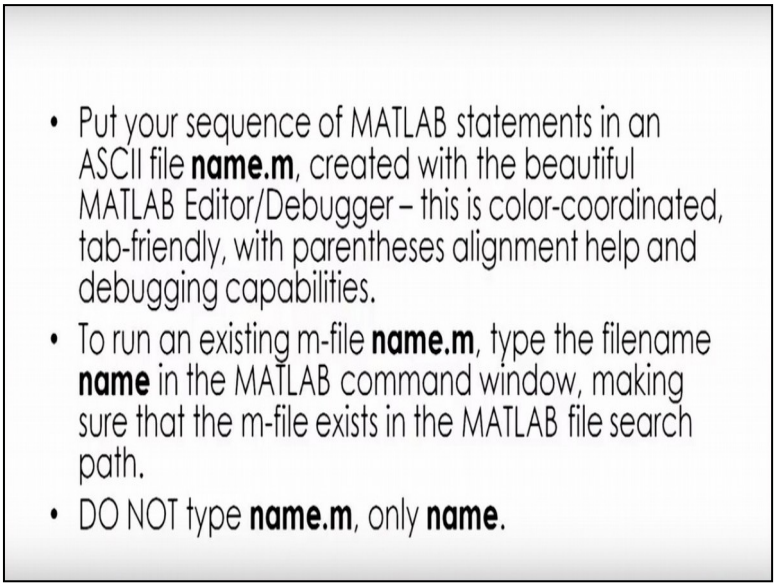
(Refer Slide Time: 09:26)



MATLAB commands and functions can be typed right into the MATLAB command window. This gets old fast especially with complicated statements and many times it is difficult to work this way that, every time on the MATLAB command prompt you type a programme. So, to make our life easier what is done is that there is a file that is one can create a file which we call it m file and on this m file we can write the quotes and then this m file can be executed by just writing the file name in the MATLAB command window.

So, this way we can execute the m file.

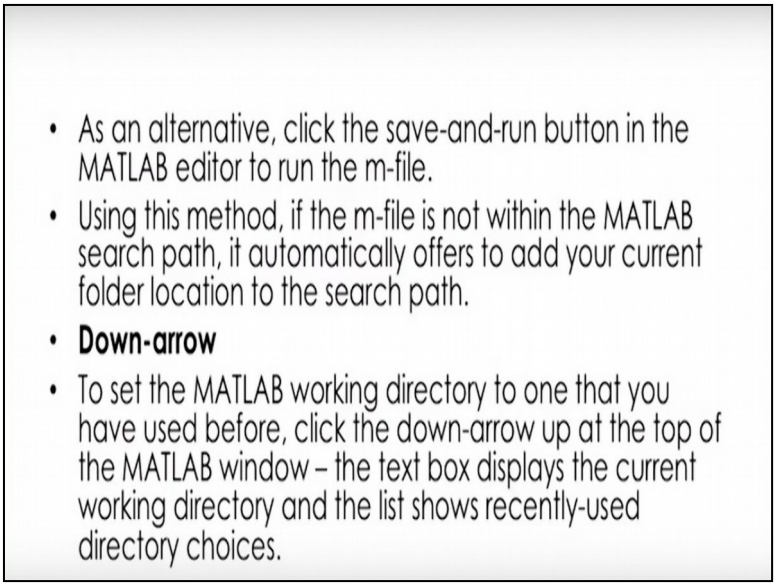
(Refer Slide Time: 10:36)



So, put your sequence of MATLAB statement. Suppose, I put in some ASCII files name dot m created with beautiful MATLAB editor or debugger and of course, this is color coordinated tab friendly with parentheses alignment help and debugging capabilities all these things are

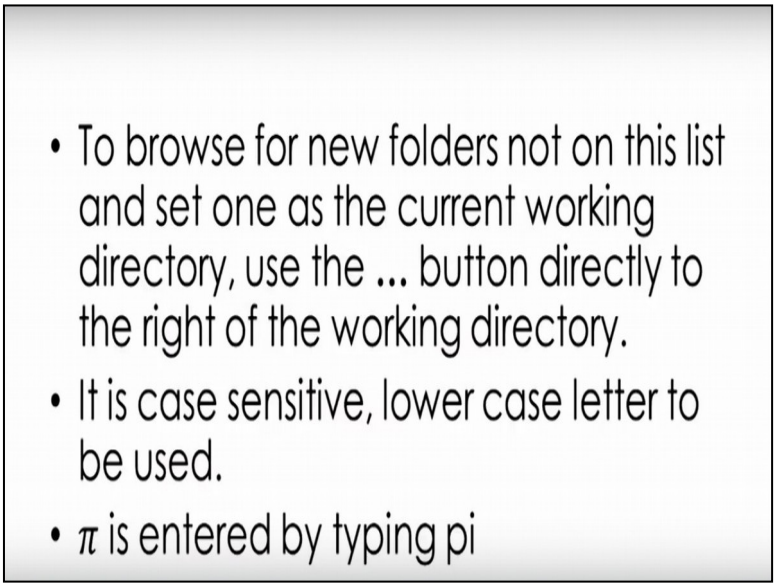
there and to run the existing m file name m type only. The file name in the MATLAB command window making sure that the m file exists. Of course, in the MATLAB file search path that has to be there set do not type dot m rather than just type name of the file.

(Refer Slide Time: 11:18)



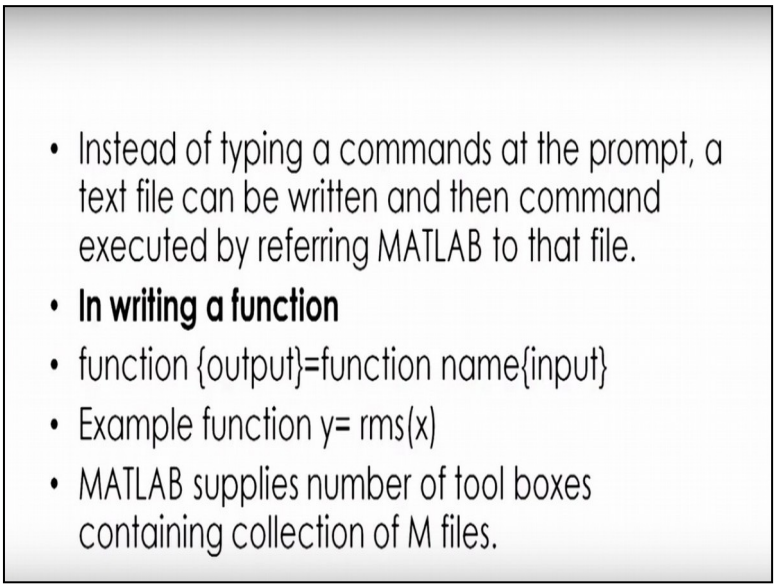
As an alternative, click the save and run button in the MATLAB editor to run the m file this can be done using this method if the m file is not within the MATLAB search path it automatically offers to aid your current folder location to the search path so this way your current folder location will be added into your search path, to set the MATLAB working directory to one that you have used before click the down arrow at the top of the MATLAB window the text book displays the current working directory and list shows the recently used directory choices.

(Refer Slide Time: 12:07)



to browse the new folders not on this list and set one as the current working directory use the three dot button directly to the right of the working directory. As I told you earlier, here the commands are case sensitive so we need to use the lower case letter. We can enter the value of pi just by typing the whole pi

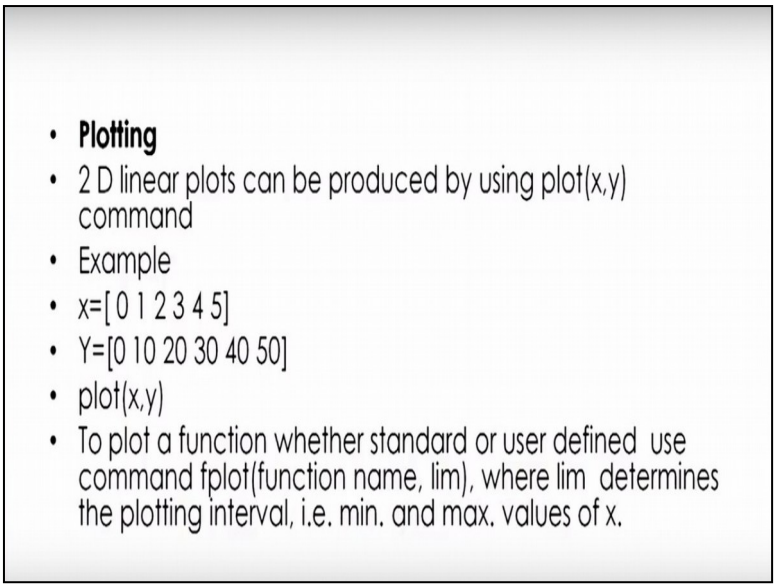
(Refer Slide Time: 12:46)



Now instead of typing a command at command prompt this is what I said earlier a text file can be written and then the command executed by referring MATLAB to that file. In the debugging window, now if, we write a function; this way some output is equal to function name and input. So that type of thing can also be done here, for example, if I write function y is equal to rms x. MATLAB supplies number of, then I can do that. and MATLAB supplies number of tool boxes having collection of m files.

One of the important activities in simulation is the plotting of the results in order to see the behavior of the dynamic system. So to plot the results we can just 2 D plot, can be produced by using the command plot x y so, we can use this plot x y command for example if, I specify x as 0 1 2 3 5 and then I specify y again as 0 10 20 30 40 50 so the plot x y will be plotting this.

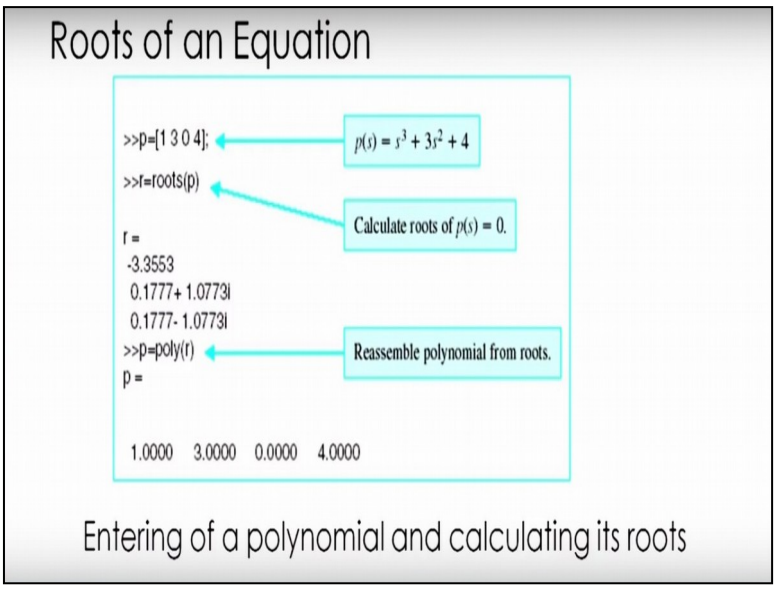
(Refer Slide Time: 12:07)



So, to plot a function whether standard or user defined use command f plot so for plotting of the function we can use just put a f at the beginning of the plot. So that way we can plot the function of course.

You have to give the function name and the lim. lim has to be defined where lim determines the plotting intervals that is the minimum and maximum values of x fine we can use mat lab. Also, for many operations here this slide just shows you to find how we can use MATLAB to find the roots of an equation.

(Refer Slide Time: 15:08)



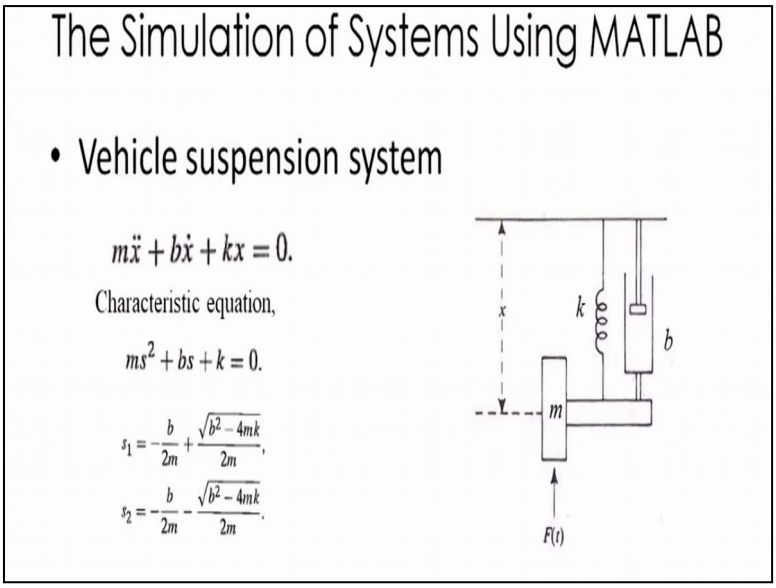
So here suppose I have got a polynomial s cube plus 3 s square plus 4 and I need to find out the root of these then I can just enter here p is equal to one for of course the coefficient of s

cube then three as coefficient of s square 0 as the coefficient of s and 4 which is nothing but the constant and then we can just call the function roots.

So, roots I am putting p and defining this by r then I will be given the roots I can do the reverse also that is given these roots I can generate my polynomial. So to do that, to generate the polynomial I have to just write the command or call the function poly break it up so using these roots it will generate the polynomial so this is what you are going to get and basically this shows the coefficient of s cube s square and s and the constant term.

So, this way we can evaluate the roots of an equation, as well as from these roots we can create the equation again. Let us take an example of simulation of a system using mat lab, here I will be taking a benchmark problem that is a simple spring mass damper system, let us assume that this is the excitation of the vehicle suspension

(Refer Slide Time: 17:02)



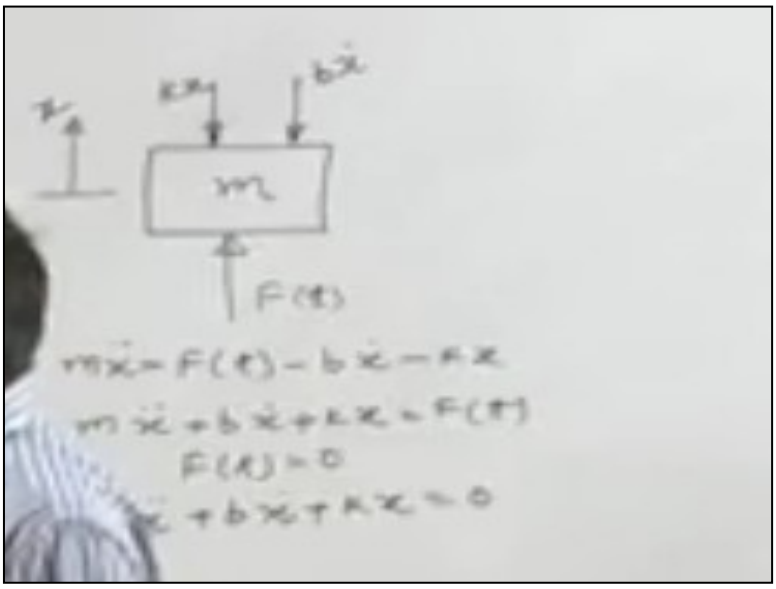
System consisting of stiffness k of the suspension system damping coefficient b of the suspention system and the mass of the wheel is m and this is subjected to some ground excitation some force coming from ground f t.

So, this vehicle suspension systems suppose if, I want to solve these using MATLAB and if I want to simulate it and if I want to see the behavior of the system or the same thing of course, I can use it for the, finding out the stiffness or damping values in order to get the desired characteristics of the suspense system. So, all those can be done with the help of this

simulation exercise. So here suppose this is my x direction that is the direction for displacement of this spring mass.

Damper system then, I can write the equation for this system by just drawing the free body diagram of it and I can find out the forces which are acting in the direction of motion, and those forces I can equate with the mass into acceleration, and I can get the system equation.

(Refer Slide Time: 18:52)



So, here the conventional way which we use it is this that if, this is my mass and I have to let us assume that this is moving in this direction and there is a force acting from here f t then the resisting force from the spring will be of course, given by k into x where x is this displacement.

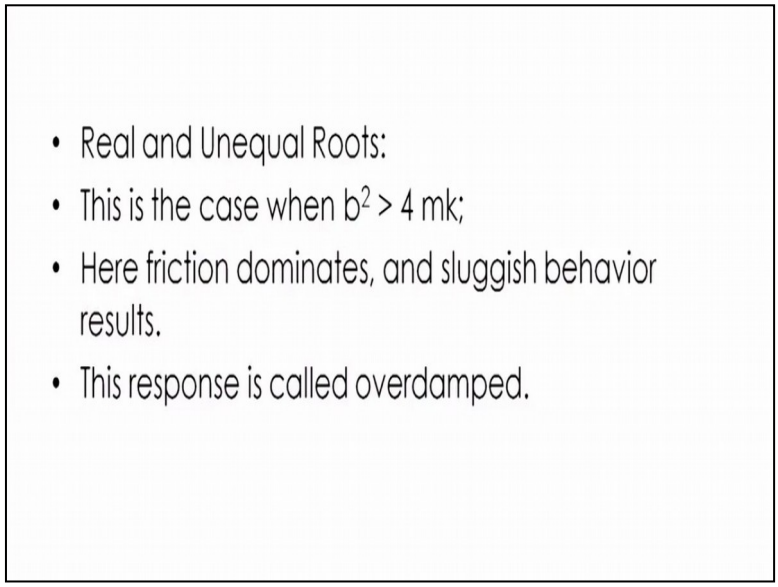
Of course, I am considering my system to be linier that is my spring is linier and spring force is proposal to the displacement likewise, I can evaluate the damping force here and the damper force will be, b x dot because damper force is going to be proportional to the velocity. So from here I can find out the net forces which are acting on the system and this net forces will be ft minus b x dot minus k x and this net force I can equate to the mass into the acceleration for this system m x double dot.

So, my system equation becomes m x double dot plus b x dot plus k x is equal to ft. Now, for seeing the unforced response if, I want to see the natural response I can take this ft is equal to zero. So, what I get is m x double dot plus b x dot plus k s is equal to zero.

So, this is the same equation here which has been shown in the slide m x double dot plus b x dot plus k x is equal to zero. Remember my aim here is to simulate this system using mat lab. Just to have little background of it I can write the characteristic equation for this one by writing as ms square plus b s plus k is equal to 0 I hope all of us are aware of how this characteristic equation are written now this characteristic equation as You can see is a quadratic equation.

Now I can find out the roots of this quadratic equation. So, this will be s1 and s2 they are the two roots so minus b by 2 m plus under root b square minus 4 m k upon 2m so other one will be minus b by 2 m minus under root b squared minus 4 m k upon 2 m so these are the 2 roots for this equation. Now, before here if you look at the nature of the roots here whether they are going to be equal root, or they are going to be distinguished root or whether they are going to be complex root that depends on the value of b square minus 4 m k so this parameter basically determines the nature of the roots.

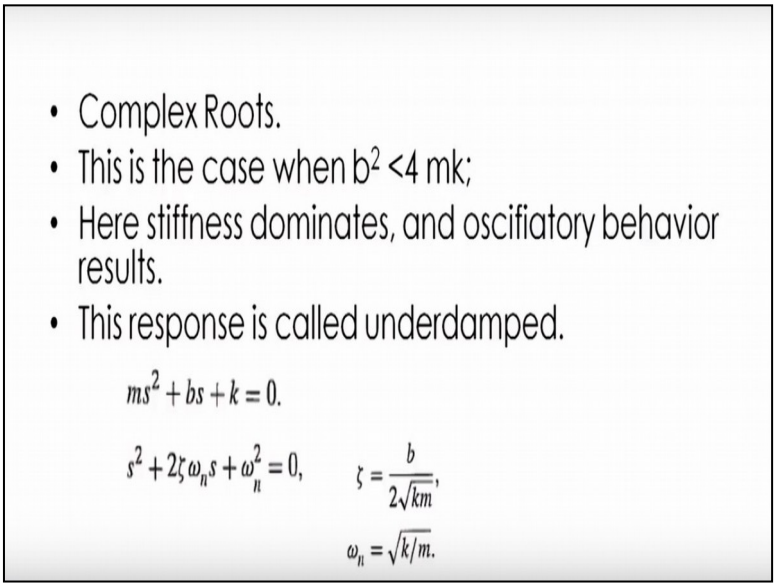
Refer Slide Time: 23:04)



So, here if b squared is greater than 4 m k then what we are going to have is the case of real and unequal roots that is, here if we look at this equation b square greater than 4 m k what does this mean, this basically means that here b is going to dominate or the friction is going to dominate and if friction is going to dominate the behavior of the system is going to be sluggish and this type of response is what is called as the over damped response this behavior we can see in the MATLAB simulation

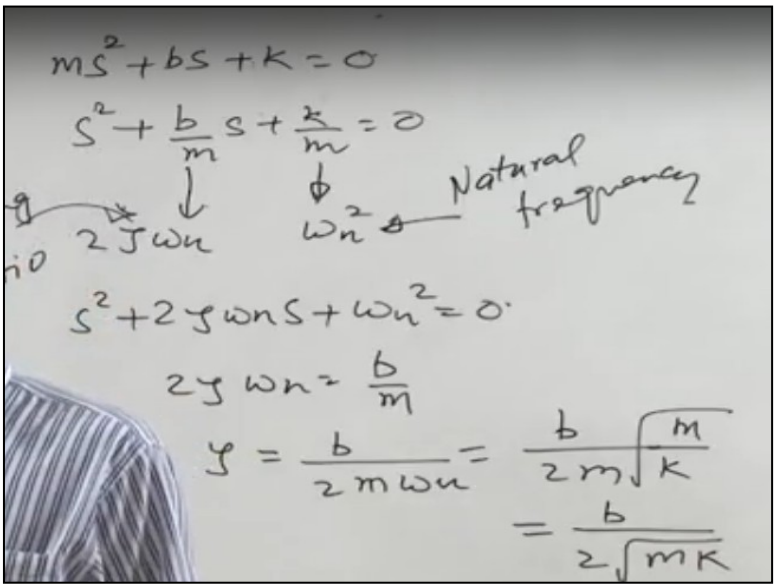
Window similarly if, the b square is equal to 4 m k then what will happen then roots are going to be real and equal and this is here if you look at this thing b square is equal to 4m k and here basically friction and stiffness are equally balanced and yielding the fastest possible non oscillatory response of the system and this response is what is called the critically damped response of the system. The third condition could be that b square is less than 4 m k.

(Refer Slide Time: 24:27)



Now, in this case you see that b squared is less than 4 m k that stiffness it means that the stiffness is going to dominate the oscifiatory behavior of the system results and mathematical, we can see that by getting the complex roots of the characteristic equation. So, this type of response is what is called as the under damped response.

(Refer Slide Time: 17:02)

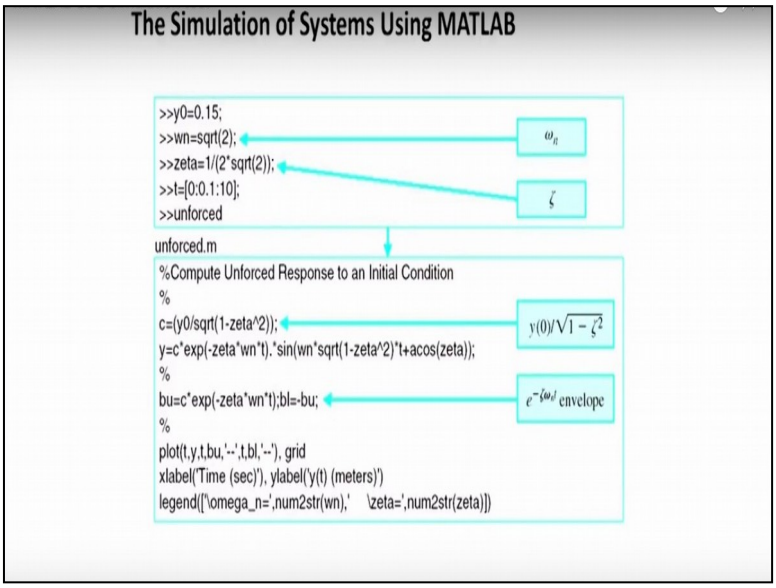


So, I have the equation m s square + b s + k equal to zero and if, I write this equation into, this from if I divide the terms of it by m my basic equation is ms square plus bs plus k is equal to 0 so, if I divide by m what I get s square plus b by m s plus k by m is equal to 0 and this k by m I represent by omega n square and b by m I represent by 2 zeta omega m if I do this my equation will become x square plus 2 zeta omega and s plus omega n squared is equal to 0.

So, this is this equation which I get now from here if, we define omega n this is called the natural frequency of the system and here this zeta is actually what is called as damping ration and of course, I can find out this 2 zeta omega n is equal to b by m so from here I can find out zeta b by 2 m omega n. So, this is how I can write it and I can further write as b by 2 m omega n is under root k by m this will be further written as b by 2 under root m k so, this is how I can write it now we can get the solution for this equation we can get the general solution for this equation.

That general solution basically will be of this form or the one which I have just written here and it will be of this form of course equation back general solution basically will be of this the one which is written here.

(Refer Slide Time: 27:40)



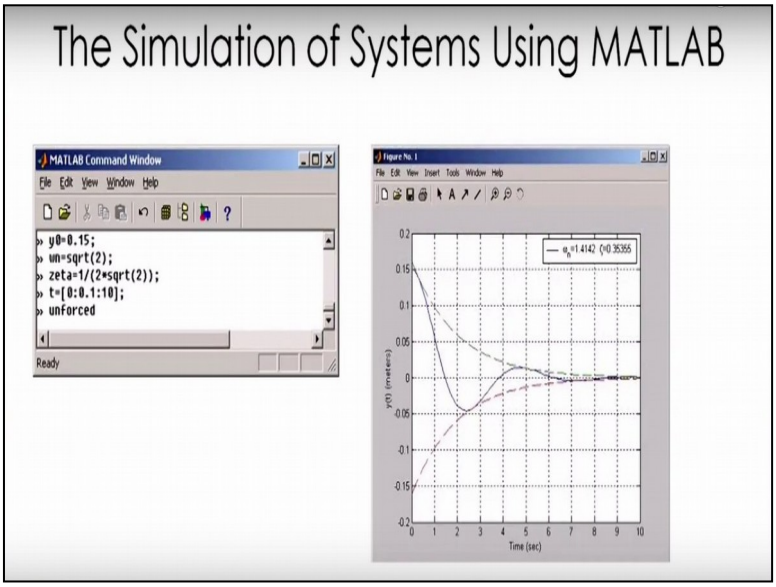
This will be form of course, in the general solution term. We are going to have the constant terms and those constant terms will be evaluated from the initial conditions of the mass. So, you are given the initial position of the mass and the initial velocity of the mass so from these two conditions you can evaluate the constant and one can get the general solution of this

system. Now, to simulate these what is done is that, we can create a m file here like this where we defined this c and of course, the general solution is written and after this just to envelop the general solution the b u and b l are evaluated and then we plot basically here.

We, plot the general solution y and we plot the two envelope bu and bl and some labeling is done and the labeling for time and the displacement y is done and we put legend for omega and zeta for a value of omega and zeta. Now, as I was telling u to evaluate or to execute this file it requires certain values parameters to be specified so what is done is that we can specify the initial value here by 0 then.

We can specify the natural frequency of the system here, this one we can specify the zeta value and of course here I can specify the plotting that is the starting time 0 the final time 10 and the plotting interval point one second. So, this is how as I said we can provide the MATLAB window and in the I can plot it. So this is how see set

(Refer Slide Time: 29:48)



we can provide MATLAB window in the MATLAB command window I can enter the values of initial value; I can enter the natural frequency; I can enter the zeta value; and I can enter the time branch I for which I want to plot and I can call the m file as the name as unforced on of this m file can be called for the execution and you can see that this is what your simulation result is there.

So, here these are two envelops which we have seen in the previous slide and this one is basically the response of your system so this way, we can use MATLAB for simulation of our

system equation these simulation could have also been done otherwise we could have done it otherwise by writing the second order differential.

Equation, which we have got, in case I would have converted those differential equation one second order differential equation into two first order differential equation and then I would solve the first order differential equations and then also, we could have plotted those graphs so the same thing we will be looking at when I will be talking about the bond graph modeling of the system and then we will see how can we simulate the system. Thank you