

**Control and Tuning Methods in Switched Mode Power Converters**  
**Prof. Santanu Kapat**  
**Department of Electrical Engineering**  
**Indian Institute of Technology, Kharagpur**

**Module - 09**  
**Large-signal Model and Nonlinear Control**  
**Lecture - 42**  
**Dynamics of LTIs and Vector Field with MATLAB Demonstration**

Welcome this is lecture number 42. In this lecture, we are going to talk about Dynamics of Linear Time Invariant system and Vector Field with MATLAB Demonstration.

(Refer Slide Time: 00:34)

**Concepts Covered**

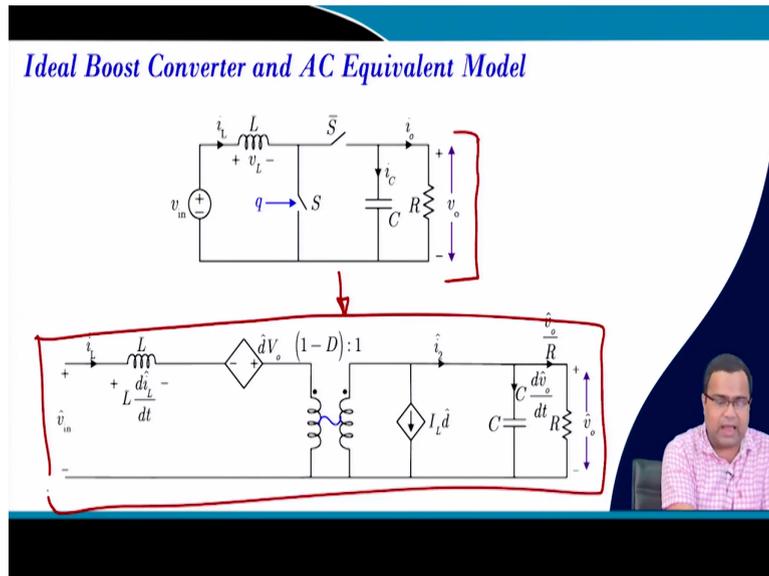
- Recap of state space model of perturbed LTI model of a boost converter
- Phase plane geometry using small-signal model
- Understanding vector field of LTI 1<sup>st</sup> and 2<sup>nd</sup> order systems
- Drawing vector fields with MATLAB demonstration

NPTEL

In this lecture we want to recapitulate state space model of perturb, basic particularly the perturbed LTI model of a boost converter and then perturb LTI model of a boost converter then phase plane geometry using small-signal model.

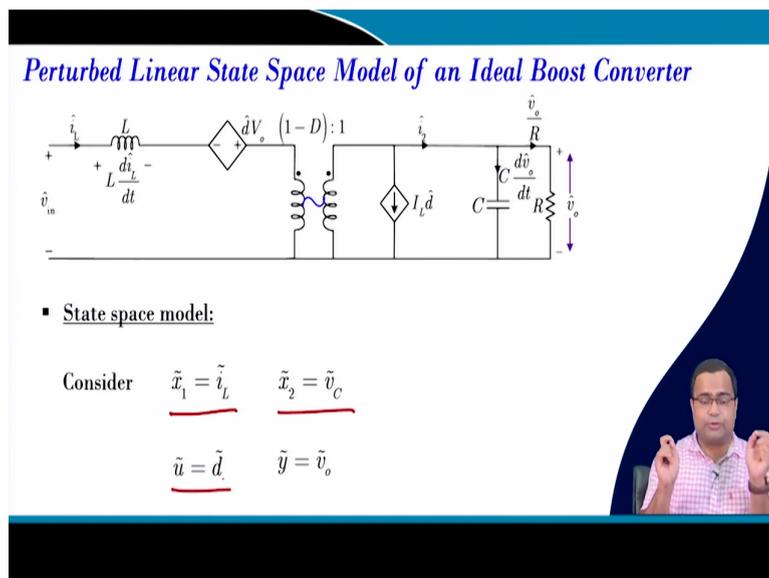
Then understanding vector field of linear time invariant 1st order as well as 2nd order system and then how to draw vector field along with MATLAB demonstration.

(Refer Slide Time: 00:59)



So, first we will start with an ideal boost converter with AC equivalent circuit and this is well known. So, this is an ideal boost converter that we have explained multiple times in this lecture and we also we have also discussed that how to obtain AC equivalent circuit of an ideal boost converter. And this circuit represents a linear circuit. This particular circuit the AC equivalent circuit and what we are going to do.

(Refer Slide Time: 01:24)



If we want to write the state space model by considering inductor current to be one state and the capacitor voltage to be the other state. In this case, the capacitor voltage and the output

voltage are same. But in a practical boost converter, there can be slightly difference because of the ESR and we are considering the duty ratio perturbation to be input; that means it is a control input because in state space model we generally write  $\dot{x}$  equal to  $a x$  plus  $b u$ . So, the control input here is the duty ratio.

(Refer Slide Time: 01:57)

*Perturbed Linear State Space Model of an Ideal Boost Converter*

$$L \frac{d\tilde{i}_L}{dt} = \tilde{v}_{in} + V_o \tilde{d} - (1-D) \tilde{v}_o$$

$$\dot{\tilde{x}}_1 = \frac{[-(1-D)\tilde{x}_2 + V_o \tilde{u} + \tilde{v}_{in}]}{L}$$

$\tilde{x}_1 = \tilde{i}_L$

So, if we write the equation of the inductor current dynamics, you know if we write the dynamics equation of the inductor current, then this is a well-known equation this we have referred earlier in the AC equivalent circuit. So, if we write in terms of state space form  $\dot{x}$  which is nothing, but  $d i_L / dt$  because we have taken  $x_1$  tilde is nothing, but inductor current tilde, then we can obtain this model of a boost converter inductor dynamics.

(Refer Slide Time: 02:28)

**Perturbed Linear State Space Model of an Ideal Boost Converter**

$$C \frac{d\tilde{v}_o}{dt} = (1-D)\tilde{i}_L - I_L\tilde{d} - \frac{\tilde{v}_o}{R}$$

$$\dot{\tilde{x}}_2 = \frac{\begin{bmatrix} (1-D)\tilde{x}_1 - I_L\tilde{u} - \frac{\tilde{x}_2}{R} \end{bmatrix}}{C}$$

We can also write the dynamics of the output voltage like the capacitor voltage and this equation is also straight forward and then we can write this x 2 dot where x 2 tilde dot is sorry is nothing, but d v 0 d t here it is a voltage tilde dot. So, we can obtain this equation.

(Refer Slide Time: 02:51)

**Perturbed Linear State Space Model of an Ideal Boost Converter**

$$\dot{\tilde{x}}_1 = \frac{\begin{bmatrix} -(1-D)\tilde{x}_2 + V_o\tilde{u} + \tilde{v}_{in} \end{bmatrix}}{L}$$

$$\dot{\tilde{x}}_2 = \frac{\begin{bmatrix} (1-D)\tilde{x}_1 - I_L\tilde{u} - \frac{\tilde{x}_2}{R} \end{bmatrix}}{C}$$

And the finally, the state space equation x 1 dot and x 2 dot these are all tilde because these are perturb quantity around their steady state quantity.

(Refer Slide Time: 03:00)

**Perturbed Linear State Space Model of an Ideal Boost Converter**

$$x \triangleq \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} \quad \dot{\tilde{x}} = \begin{bmatrix} 0 & -(1-D) \\ (1-D) & -1 \\ C & RC \end{bmatrix} \tilde{x} + \begin{bmatrix} V_o \\ L \\ I_L \\ C \end{bmatrix} \tilde{u} + \begin{bmatrix} 1 \\ L \\ 0 \end{bmatrix} \tilde{v}_{in}$$

$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} + E\tilde{v}_{in}$

So, we can obtain the overall state space model of the perturb overall perturbed state space model of the AC equivalent circuit of the boost converter and this A matrix is a system matrix. So, this is a standard representation of a state space model where x is the state vector. We have taken here the inductor current and the output voltage x 1 and x 2 and u tilde we have taken as the duty ratio, which is a control input. Here we have a supply disturbance, which is a disturbance input. That is why it is coming to be E matrix.

(Refer Slide Time: 03:36)

**Perturbed LTI Model of an Open-Loop Boost Converter**

Handwritten notes:  $d = D$ ,  $\tilde{d} = 0$ ,  $\tilde{v}_{in} = 0$ ,  $\tilde{u} = \tilde{d} = 0$ ,  $\tilde{v}_{in} = 0$

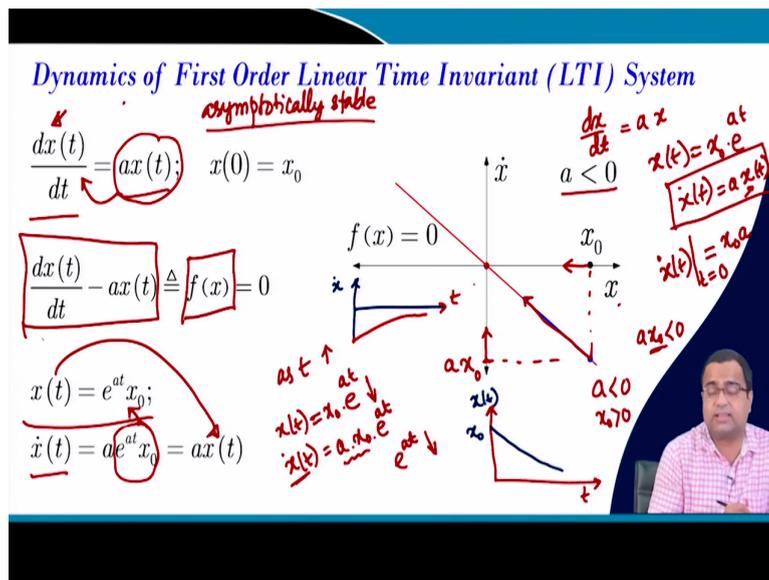
$$\dot{\tilde{x}} = \begin{bmatrix} 0 & -(1-D) \\ (1-D) & -1 \\ C & RC \end{bmatrix} \tilde{x} \quad \dot{\tilde{x}} = A\tilde{x}$$

Open-loop boost converter (perturbed model) – autonomous LTI system

Now, if we consider an open loop boost converter, where we are considering the instantaneous duty ratio to be the fixed duty ratio; that means, the perturbation in the duty ratio is 0, then the supply disturbance is also considered to be 0 open loop ok. So, this is 0; that means, this part is shorted and this part where we had that  $v_0$  into  $d$  tilde that is 0 because we have ignored the perturbation the duty ratio because we are using a fixed duty ratio.

Then we will only have this equation because we will have  $u$  tilde which is nothing, but  $d$  tilde equal to 0 and then  $v$  tilde  $v$  in tilde equal to 0. So, we will get  $x$  tilde hat equal to  $A$  into  $x$  tilde. So, this represents  $\dot{x}$  equal to  $A x$  which is nothing, but this represents an autonomous system linear time inference system where there is no perturbation in the input signal ok.

(Refer Slide Time: 04:40)



Now we want to study the dynamics of linear time invariant autonomous system ok. So, first we will start with the first order system where  $x$  is the state, and it is a first order equation  $\frac{dx}{dt} = ax$  where  $a$  whether it is positive or negative or it is 0 depending upon that the stability of this system will be determined. So, if we write  $\frac{dx}{dt} - ax = 0$  and if we take  $-ax$  term to the left side, then you can represent this and this quantity we are representing in terms of  $f(x)$  which is equal to 0 ok.

Now, if we solve this first-order differential equation because we have  $\frac{dx}{dt} = ax$  then you will get  $x(t) = e^{at}x_0$  where  $x_0$  is the initial condition and  $a$  is the system parameter. This is standard.

form right. So, this is what we have written here. And if you differentiate  $\dot{x}$  then what you will get? You will get  $a$  into  $x$  and that you can write because if you write  $\dot{x}$  it will be  $a$  into  $x$  and this term is already it is written. It is nothing but this term. So, we can write.

Now, we got  $\dot{x}$  equal to this equation. So, we want to draw the state space form; that means,  $x$  and  $\dot{x}$  plane right. So, in this plane if we consider initially  $a$  to be negative ok then we know that if  $a$  to be negative for this system, then this system will be asymptotically stable asymptotically stable. Here it acts like an eigenvalue because it is a first order system.

Now, the starting point for this  $x$  axis is  $x_0$ , then if you take  $\dot{x}$  and if you obtain what is my  $\dot{x}$  at  $t$  equal to 0, you will find it is nothing, but what is that  $x_0$  right into  $a$  because this  $x$  will be  $x_0$ .

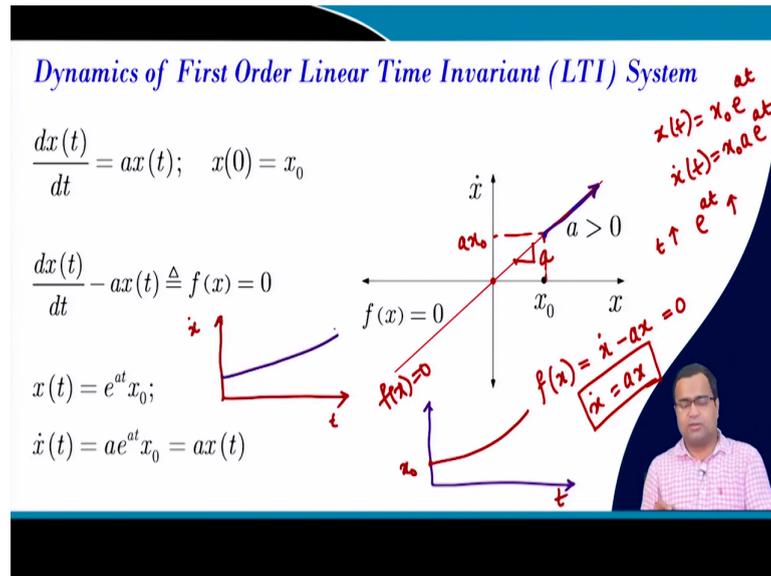
So, it will be like this quantity will be  $a$  into  $x_0$ , since  $a$  is negative and  $x_0$  is positive. So, this quantity will be negative right because we have to consider  $a$  to be negative and  $x_0$  to be positive. So, there  $a$  into  $x_0$  must be negative and this is coming here. Then, as time you know increases; that means, as  $t$  increases, what we will find that  $x$  of  $x$  of  $t$  which is nothing, but  $x_0$  into  $e$  to the power  $a$   $t$  since  $a$  is negative this quantity will decrease right.

So, this quantity will be decreased in this direction. What about  $\dot{x}$ ?  $\dot{x}$  if you write it is simply what we have obtained?  $a$  into  $x_0$  into  $e$  to the power  $a$   $t$  right. It is simply like this. So, this quantity since it is already negative, this quantity is negative. So, we can say that  $e$  to the power  $a$   $t$  this quantity will decrease. So, naturally this quantity will be increase. So, it will go in this direction.

So, effectively their combination will move along this line and this is the direction along which that the trajectory that means this vector will move and it will finally approach towards origin as limitation (Refer Time: 08:42) it will go to 0; that means, if we draw the time domain axis if we draw the time domain axis. So, this will be like this. So, this is my  $x_0$ , and this is my  $x$  of  $t$  axis.

And if we draw  $\dot{x}$  sorry if we draw  $\dot{x}$ , then what we will find that we have considered to it will start from negative and it will slowly approach towards this. So, this is my time axis  $\dot{x}$  right. So, it will approach to this.

(Refer Slide Time: 09:26)

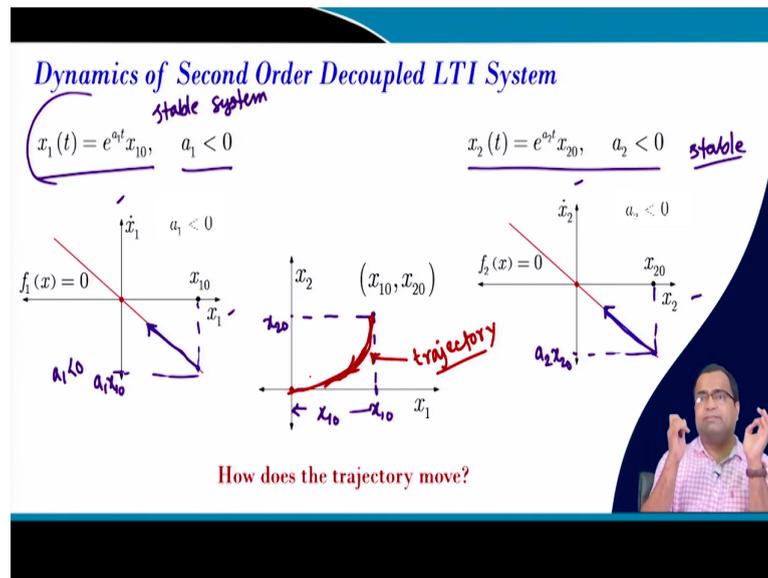


Now, if we consider  $a$  to be positive the same thing, then we know what is our  $x$  of  $t$ . It is again  $x_0$  into  $e$  to the power  $a t$ ,  $\dot{x}$  will be  $x_0$  into  $a$  into  $e$  to the power  $a t$ . Since  $a$  is positive; that means, as  $t$  increases  $e$  to the power  $a t$  will increase. And what is the equation of this line?

So, this is  $f(x) = 0$  and what is  $f(x)$  that we have chosen? It is nothing, but  $\dot{x}$  minus  $a x$  that is equal to 0. So,  $\dot{x} = a x$ . So, it is a straight line with a positive slope because  $a$  is positive. That is exactly that slope is positive here; that means, if you start with a point  $x_0$  initial condition, then this will be  $a x_0$  since  $a$  is positive and then it will move along this vector and it will go away because it is unstable.

So, it will go away from the origin. That means, if you use a different colour, it will go away from origin and the system is unstable because if we see the time axis then if you start from this point  $x_0$ , it will slowly increase like this as time  $t$  increases and similarly if we take  $\dot{x}$  with respect to  $t$ , then this will if we take it will also move along like this ok.

(Refer Slide Time: 11:16)



Now, if we consider two stable systems, we have taken one first order system with a negative; that means, it is a stable system. It is a stable system. We have considered another system. Again it is a negative, it is also stable system. It is not necessary that two systems have to be stable; one can be stable while the other can be unstable, both can be unstable, both can be stable. Multiple scenario will happen, but if we understand how to draw the vector field, then we should be able to draw for any condition on a 1 and a 2 then we can draw the vector field.

And we know for the individual vector field if we start with  $x_1 \neq 0$ , since  $a_1 < 0$  so; that means, this is this is my  $a_1 \times x_1 \neq 0$ . So, the point will move along this line. This is value. Similarly, if we take this one; that means, this is my  $a_2 \times x_2 \neq 0$  again. This will move along this line and both the case is stable. So, they will approach towards origin.

Now if we combine and draw  $x_1$  versus  $x_2$  here we have drawn  $x_1$  versus  $\dot{x}_1$ ,  $x_2$  versus  $\dot{x}_2$  now if you draw  $x_1$  versus  $x_2$  then this component is  $\dot{x}_1 \neq 0$  this length right this one along this and this is nothing, but our  $\dot{x}_2 \neq 0$  right. So, this component is  $\dot{x}_1 \neq 0$  and  $\dot{x}_2 \neq 0$ .

So, now it will start and then it will slowly come towards origin right so, towards origin. But, the question is how does the path look like? And that path we call as a trajectory. If you start with the initial condition, then for a given value of a 1 and a 2 you will get an unique path along which it will move along towards the origin since a 1 and a 2 are negative because the system is stable and we are interested to draw this vector field how to draw it.

(Refer Slide Time: 13:40)

*Dynamics of Second Order Decoupled LTI System*

$$x_1(t) = e^{a_1 t} x_{10}$$

$$x_2(t) = e^{a_2 t} x_{20}$$

Identity matrix

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{a_1 t} & 0 \\ 0 & e^{a_2 t} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

State vector  $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Initial vector  $\mathbf{x}_0 = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$

initial state vector or initial vector

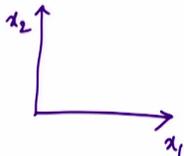
Now, we have considered a second order system and we know the solutions of individual subsystems are of first-order, which can be written in this form. Now we can combine together to create a kind of vector. So, this is like a state vector where this is my state vector and these are the initial vector or you can say initial vector or I can say initial state vector or simply I can say initial vector ok initial vector.

And this is now it looks like a square matrix and it is a decoupled system because each system like an individual subsystem which does not depend on the other; that means, they are totally separated. Now, we can write in a compact form; that means, this matrix, this particular matrix we are writing again here, now we can multiply with an identity matrix.

This is an identity matrix. We are multiplying with an identity matrix. Why we are doing that? And this we call vector  $\mathbf{x}$  this is a vector, this is also vector  $\mathbf{x}_0$  which is an initial state vector and both cases they are a combination of two states  $x_1, x_2$  and this is an identity matrix. Why we have considered that?

(Refer Slide Time: 15:12)

*Dynamics of Second Order Decoupled LTI System*

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{a_1 t} & 0 \\ 0 & e^{a_2 t} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$


$$\begin{matrix} \xrightarrow{x_1} \\ \xrightarrow{x_2} \end{matrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \boxed{1} & \boxed{0} \\ \boxed{0} & \boxed{1} \end{bmatrix} \begin{bmatrix} e^{a_1 t} & 0 \\ 0 & e^{a_2 t} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

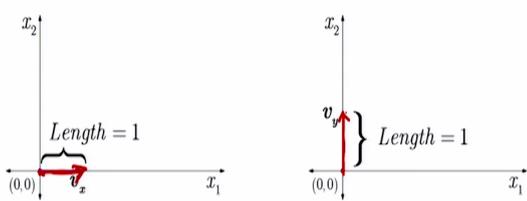
$v_x \quad v_y$



So, if we write down again, then I am rewriting the equation again. I can consider this to be a column vector. For example, we will see the significance of this column vector, I will consider another column vector to  $v_y$  so; that means, there are two column vector  $v_x$  and  $v_y$  ok. Why I have considered this? Since it is a decouple system why I have chosen this column vector? Let us first draw in  $x_1 \times x_2$  axis how does the vector look like.

(Refer Slide Time: 15:46)

*Column Vectors*

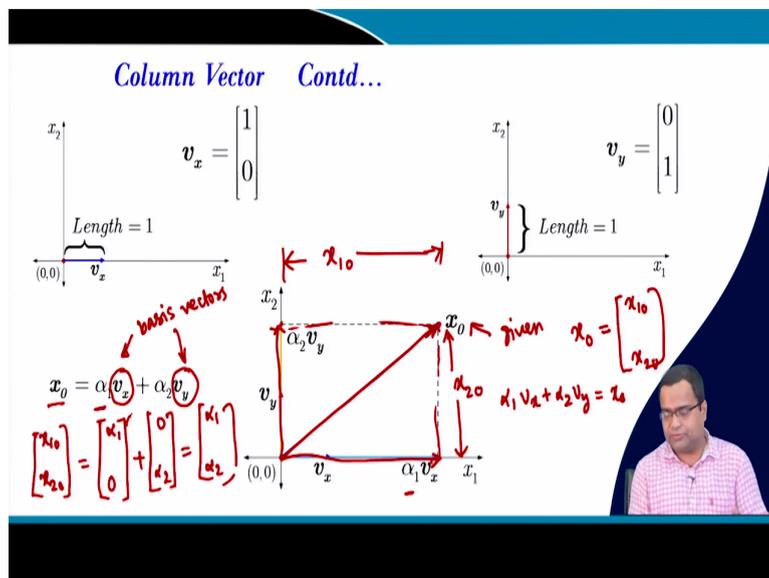
$$v_x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v_y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



So, it is a column vector. And you see, if you go back, this is the axis which corresponds to  $x_1$ . This is the axis which corresponds to  $x_2$ ; that means, if we draw  $x_1, x_2$  axis

then we are interested to know what is the component of this what is the component of this vector along x 1 axis along x 2 axis?

Similarly, what is the component of this vector along x 1 axis along x 2 axis? So, this is why we are interested. So, the first vector has 1-unit component along x 1 axis 0-unit component around x 2 axis. So, you can write 1-unit length here. So, this is the vector. This vector looks like this; that means, this v x vector will be simply like this; this vector consisting of 1-unit length along x 1 axis, similarly v y vector has 1 unit along x 2 axis that is why it will look like this 1-unit length.

(Refer Slide Time: 17:01)



Now, we can draw two vectors and we want to write that what is my initial state that we will see soon. Suppose if we want to create a vector using a linear combination of v x and v y vector and you see since it is along x 1 axis 1-unit length, this is along y x 2 axis 1-unit length we call them as a basis vector. So, these are the basis vectors, the basis vector.

So, this basis vector since we are scaling this vector alpha times. So, this alpha time v x is this and alpha 2 times v y is the v y is this. Why we have chosen? Because their resultant should combine to generate this. This is my initial vector or in other way if the initial vector is given, if this is given because what is this? This is x 0 which has 2 component x 1 0 and x 2 0; that means, I can say in this axis it has x 1 0 unit and in this axis it has x 2 0 unit.

So, then this is the coordinate and I want to match this coordinate is horizontal and the vertical displacement and from there I can solve alpha 1 alpha 2. That means, what is the bottom line? The alpha 1 v x plus alpha 2 v y is my x 0 which is nothing, but from this case it is nothing, but x 1 0, x 2 0 is nothing, but alpha 1 0 plus 0 alpha 2. This is simply nothing, but alpha 1 alpha 2 because each of these v x their magnitude is unit length right. So, that is why we are writing alpha 1 alpha 2. So, alpha 1 here is simply x 1 0, alpha 2 is here nothing, but x 2 0.

(Refer Slide Time: 19:23)

*Column Vector Contd...*

$$\mathbf{x}_0 = \alpha_1 \mathbf{v}_x + \alpha_2 \mathbf{v}_y$$

- Any point  $x_0$  in 2-D space can be connected by spanning vectors  $v_x$  and  $v_y$  using co-efficient  $\alpha_1$  and  $\alpha_2$
- What happens if vectors  $v_x$  and  $v_y$  are not orthogonal?
- What happens if  $v_x$  and  $v_y$  are not linearly independent?

So, we can get this column vector any point in x 0 axis; that means, any point. So, any point x 0 in two-dimensional space, these are two-dimensional planes x 1, x 2 can be connected by spanning this unit basis vector. By spanning this basis vector, we can generate any point in this two-dimensional plane ok. So, what happens if the vector v x and v y are not orthogonal? What happens if the v x and v y are not linearly independent?

So, here they are linearly independent because in the vector diagram when we will call linearly dependent? When they are parallel or they are basically one is a subset of other right, but since here v x and v y they have certain angle, they are not overlapped or they are not parallel. So, they are linearly independent.

(Refer Slide Time: 20:18)

*Column Vectors are Not-Orthogonal*

$$v_x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$v_y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

Now, the column vector which are not orthogonal. So, we can consider another scenario A where the first vector is  $v_x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$  so; that means you take here 1 unit length here and you take another unit length. So, this is my 1-unit length and this one I have 1-unit length. So, that together they combinedly create this  $v_x$ . Similarly, if I take  $v_y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$  1-unit length along this axis 1-unit length and along this axis 2-unit length ok. So, they combine if you join them. This is my  $v_y$  ok. So, we can draw any vector in two-dimensional plane for any two-dimensional vector right.

(Refer Slide Time: 21:11)

*Column Vectors are Not-Orthogonal*

*basis vectors  
 $v_x$  &  $v_y$   
not  
orthogonal.*

➤ Any point  $x_0$  in 2-D space can be connected by spanning  $v_x$  and  $v_y$  using co-efficient  $\alpha_1$  and  $\alpha_2$

$x_0 = \alpha_1 v_x + \alpha_2 v_y$

➤ Not possible if  $v_x$  and  $v_y$  are linearly dependent !!

So, we got now for this particular case. These are the basis vector. So, earlier the basis vector was along  $x_1$  and  $x_2$ , but those vectors are orthogonal, but now they are no longer orthogonal for this case. Here the basis vector  $v_x$  and  $v_y$  and we have to represent our initial condition, our  $x_0$  in two-dimensional plane using this two basis vector. So, here our basis vectors in this case  $v_x$ ,  $v_y$  these are the basic vector and they are not orthogonal right. They are not orthogonal because they have angle, but that angle is not 90 degree in this case.

So, the angle is less than 90 degree, but greater than 0. Now if I again take an initial condition here which has a distance let us say  $x_{10}$  from here and which has a distance  $x_{20}$  from here. So, we can resultant vector can be mapped by scaling this  $\alpha_1 v_x$ , the  $v_x$  vector this by  $\alpha_1$  spanning and this vector which was  $v_y$  here, it was our  $v_y$  we can span in the negative direction so, that the resultant this plus this will give you this  $v_x \times x_0$ .

So, we can generate this vector from these two vectors. That means, we can in fact, any point  $x_0$  can be mapped by  $v_x$  and  $v_y$  by spanning them; that means, they are scaling those vectors and we can obtain the resultant vector. So, again, we can  $x_0$  can be mapped or connected in this two-dimensional plane by  $v_x$  and  $v_y$  by spanning using their coefficient  $\alpha_1$ ,  $\alpha_2$ , but this is not possible if two vectors are linearly dependent. So, our first requirement, the vector must be linearly independent.

(Refer Slide Time: 23:18)

*Dynamics of Second Order Decoupled LTI System*

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{a_1 t} & 0 \\ 0 & e^{a_2 t} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$v_x \quad v_y$

$$\mathbf{x}(t) = e^{a_1 t} \alpha_1 \mathbf{v}_x + e^{a_2 t} \alpha_2 \mathbf{v}_y$$

$$\alpha_1 \mathbf{v}_x + \alpha_2 \mathbf{v}_y = \mathbf{x}_0$$

> 2-D vectors  $v_x$  &  $v_y$   
 > Linearly independent vectors

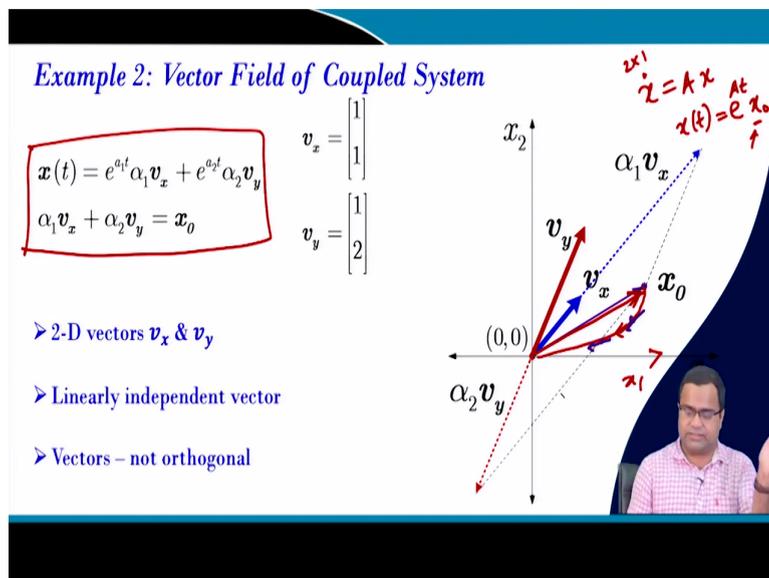
Now, we are going a decouple system writing again  $v_x$ ,  $v_y$ . So, we can write the solution; that means, our original equation  $\dot{x}$ . So, for  $\dot{x}_1$  we know it was a  $1 x_1$ , for  $\dot{x}_2$  it

was a 2 x 2. So, you can write x dot to be a 1 0 0 a 2 into x. So, for this system, even for any second order system arbitrary system, you know whether it is in this form or other form, if we can find we have not yet discussed what are the eigenvalues.

So, for this case we will consider a 1 a 2 they are given and in subsequent in the next lecture we will see we will when you will find eigenvalues then this will be written in terms of e to the power the eigenvalue times t alpha into v x and alpha 2 into v y. So, we can solve this system by writing this equation. This is very standard well known a second order system even if you have a nth order system you can write e to the power a 1 2 alpha 1, v 1, e to the power a 2 t alpha 2 v 2 and so, on we can add it e to the power a n t alpha n v n.

So, you can write n number of combination. For this second order at t equal to 0 initial condition this quantity will be 1 this will be 1. So, it will be alpha 1 v x plus alpha 2 v y equal to x 0 and that we have already drawn that we can get we can draw this x 0 vector by a combination of v x and v y by spanning them using scalar quantity alpha 1 alpha 2. So, this v x v y the two-dimensional vector they are linearly independent and orthogonal.

(Refer Slide Time: 25:18)



Now, we are considering another case the coupled system, that we have already discussed that v x and v y they are not orthogonal, but again if we have the equation which we have not written yet suppose we consider the same a 1 a 2 for the system, but now the vectors are different. We want to see how does the vector look like and we have already discussed that if

this is my x 1 axis v x, v y you can get x 0 by a linear combination, it is also v x v y they are second like a two-dimensional vector linearly independent, but they are not orthogonal.

So, now how to draw a vector field? So, whatever we have drawn here we have as if drawn just the initial condition right if you take x dot equal to A x. So, we generally get x of t to be e to the power A t into x 0. So, far we are able to at t equal to 0 we are able to locate because this is a two-dimensional this is 2 cross 1 two-dimensional vector, we are able to locate the initial vector in the two-dimensional plane.

But, we want to see how does this point move it means the system is stable. How does this point move in this two-dimensional plane? What is the trajectory and how to draw the vector field? So, this trajectory can be drawn by using vector field; that means, every point if you draw a tangent if you draw a tangent. So, you will get the direction of the vector field. So, by that computing the direction of vector field, we can slowly move the point and we can eventually draw the trajectory from where that like x 0 will that will follow that path to reach to origin at limit t tends to infinity for a stable system.

(Refer Slide Time: 27:16)

*How to Draw Vector Field of Dynamical System?*

$$\dot{x} = \underline{Ax} = \underline{f_x(x)}, \quad A = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$$

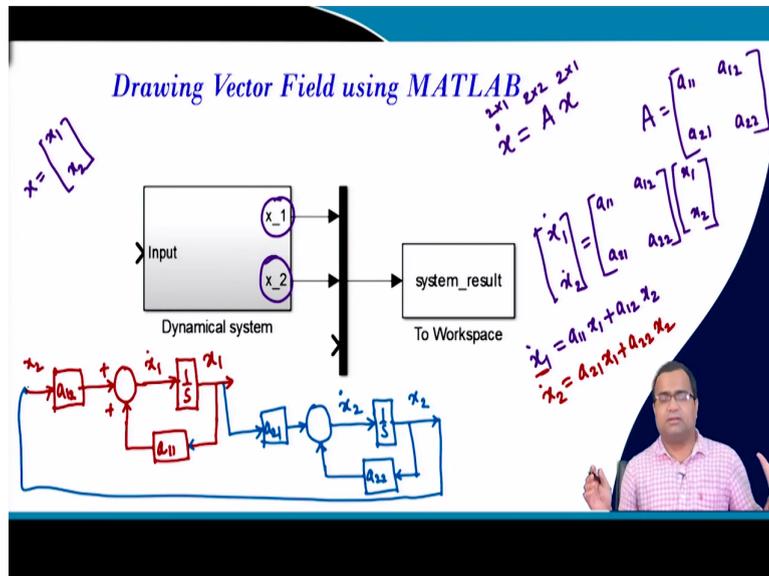
- Approach 1 – using MATLAB custom simulation
- Approach 2 – using analytical approach with vector field

*(A small video inset of a speaker is visible in the bottom right corner of the slide.)*

So, how to draw the vector field? Let us start x dot equal to A x where the A x can be written like a vector, f x as a function of x where we are starting with a decoupled system a 1 a 2 ok. So, one approach using MATLAB custom simulation and other approach using analytical approach using vector field. So, we will first consider how to draw a vector field in MATLAB

because we want to get some visual understanding before we draw the real vector field in the in this like a presentation.

(Refer Slide Time: 27:48)

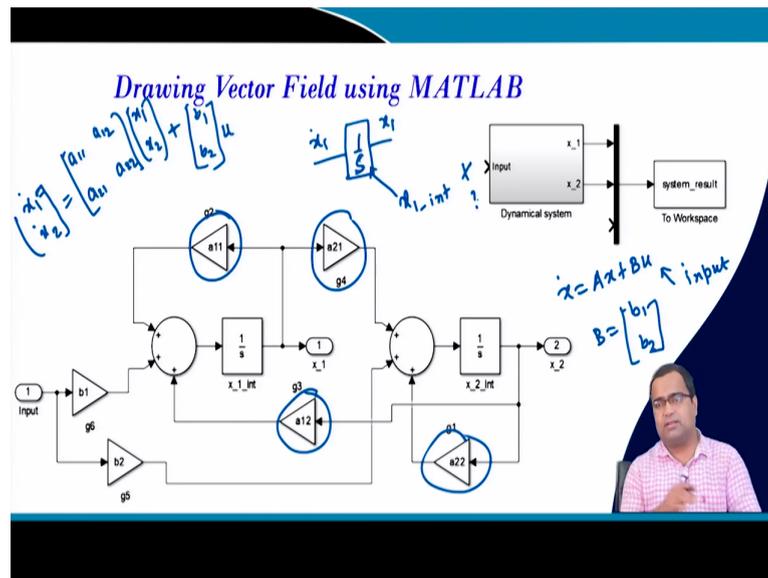


So, first I have considered one dynamical system and what is the dynamical system?  $\dot{x}$  equal to  $A x$  right and since it is we have taken only two state you can see  $x_1$  and  $x_2$ . So, this is also 2. So, this has to be 2 cross 2 and that means, what is my state vector? It is nothing but  $x_1$  and  $x_2$ . And what is my  $A$  matrix I have taken? Here I have taken a 11 a 12 a 21 a 22. I will get  $\dot{x}_1$   $\dot{x}_2$  is equal to a 11 a 12 a 21 a 22 into  $x_1$   $x_2$  which means  $\dot{x}_1$  is equal to a 11  $x_1$  plus a 12  $x_2$  plus yeah because there is no input so, that is it.

And what about  $\dot{x}_2$ ?  $\dot{x}_2$  will be a 21  $x_1$  plus a 22  $x_2$ . Now we already know that how to plug in this equation in MATLAB because the first thing for  $x_1$  we need one integrator right. So, this is one integrator and if this side is our  $\dot{x}_1$  then this will be  $x_1$ . This we have already learned multiple times right because we started this presentation in lecture number 4 in the very first week. Then what is my  $\dot{x}_1$  dot?

So, it is a combination of it is a combination of a 11  $x_1$  and there will be a 12 into  $x_2$ . From where we are getting this  $x_2$ ? Again, we will create another one. So, let us draw it here; that means, we will take another integrator. This is my  $x_2$ . This side we will have  $\dot{x}_2$  again this side we will get, it will have its own self coefficient which is a 22 ok then it will also have the other coefficient which is a 21 and in this case it is simply this one right. Then we can connect this one to this point. So, this gives us the complete equivalent right.

(Refer Slide Time: 31:05)



And this is exactly what I did a 11 like sorry a 11 into this a 21 into this a 12 a 22 whatever I have explained and we already know that if you go inside the integrator like here, we will have  $\dot{x}_1$  it is  $x_1$ . So, you can set the initial condition  $x_1$  initial condition right inside the integrator. If you go click it we will show you very soon.

Now, suppose if we take  $\dot{x}$  is equal to  $Ax + Bu$  where  $u$  is the input this is a general representation of a state space system where  $B$  matrix is nothing, but  $b_1$   $b_2$ . Then what will be our equation?  $\dot{x}_1$  will be  $\dot{x}_2$  will be  $a_{11} a_{12} a_{21} a_{22} x_1 x_2$  plus  $b_1 b_2$  into  $u$ .

So, this is what I have exactly realized, but we are not going to consider input, but this model we made because if we want to use because in DC-DC converter ultimately we need to incorporate input right. So, there will be input voltage gate signal all this will come, but to start with we can have we can remove the input. So, you see, I have not connected any input. So, for the input term, there is no effect. Nothing will come.

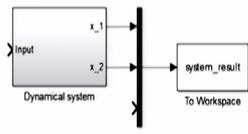
(Refer Slide Time: 32:49)

### Drawing Vector Field using MATLAB

```
clc; close all; clear all;

%% System parameters
system_parameter;
a11=-100; a12=0; a21=0; a22=-200;
q=0; b1=q; b2=0;
A=[a11 a12; a21 a22]; B=[b1; b2];

%% Initialization
x_1_initial=2; x_2_initial=2;
figure(2),
plot(x_1_initial,x_2_initial,'or'); hold on;
```



File name – system\_simulation.m

```
L=2e-6; C=100e-6;
R=1; Vin=10;
```

File name  
system\_parameter.m



So, then I am in order to draw a vector field, I have just created 1 dot m file which is like a Simulink system simulation underscore simulation dot m it will ask for system parameter although I am not giving any parameter, but this is used if I want to incorporate a DC-DC converter model. So, for the time being you can ignore this because in this particular lecture we are not calling this parameter. We are writing all the a 1 a 11 a 12 a 21 a 22 by numerical value.

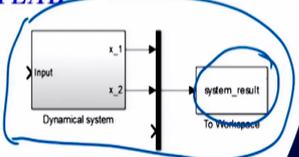
So; that means we are putting the values we are not using any symbol b 1 b 2 we are b 1 I am getting q 0. So, if q equal to 0 there is no input right. So, I am using A matrix and B matrix I told you and these are the initial condition figure and then you can plot for the initial figure.

(Refer Slide Time: 33:42)

### Drawing Vector Field using MATLAB

```
%% System simulation

t_sim=200e-3;
sim('system_dynamics.slx'); clc;
t=system_result.time; t_scale=t*1e3;
x=system_result.data;
x_1=x(:,1); x_2=x(:,2); u=x(:,3);
```



File name – system\_simulation.m



Now, if you go for system simulation, it will it is continuing because this whole file is. It is you know, length is quite large. I will show you.

So, I have broken into multiple lines. I you know another part I am calling running for 200 milliseconds. So, you can change the time. Now I am calling something called system underscore dynamic dot slx file this file consist of this Simulink model then I am getting the data from the system result; system result dot time, system result dot data and I have already explained how to access the data without putting a scope.

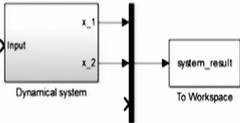
(Refer Slide Time: 34:21)

### Drawing Vector Field using MATLAB

```
%% Plotting
figure(1),
subplot(2,1,1)
plot(t_scale,x_1,'b','Linewidth', 2); hold on; grid on;
subplot(2,1,2)
plot(t_scale,x_2,'r','Linewidth', 2); hold on; grid on;

figure(2)
plot(x_1,x_2,'r','Linewidth', 2); hold on; grid on;

system_vector_field;
```



File name – system\_simulation.m



Then I want to plot them in the first figure. I am taking two sub figure like a subplot and the first plot is  $x_1$ , second plot is  $x_2$  with respect to time how they are changing. And second figure, I am taking phase plane  $x_1, x_2$  which I am trying to explain and along with that I am also putting a vector field. What is the vector field?

(Refer Slide Time: 34:44)

**Drawing Vector Field using MATLAB**

```

N_diff=100;
x_lower=-15; x_upper=15;
y_lower=-5; y_upper=5;
x_diff=(x_upper-x_lower)/N_diff;
y_diff=(y_upper-y_lower)/N_diff;
figure(2),
[x,y]=meshgrid(x_lower:x_diff:x_upper,y_lower:y_diff:y_upper);
d_x=(A(1,1)*x)+(A(1,2)*y)+B(1)*q;
d_y=(A(2,1)*x)+(A(2,2)*y)+B(2)*q;
quiver(x,y,d_x,d_y,'b');
xlabel('State x1');
ylabel('State x2');

```

File name - system\_vector\_field.m

Handwritten notes:  $100 \times 100$ ,  $\Delta x$ ,  $\Delta y$ ,  $x_1$ ,  $x_2$ ,  $x_{1min}$ ,  $x_{1max}$ ,  $x_{2min}$ ,  $x_{2max}$ .

So, it is a quiver plot; that means, if you take let us say suppose I want to draw a vector field along  $x_1$  and  $x_2$ . So, this is my  $x_1$  axis  $x_2$  axis. Now, I want to plot the vector field by some limit; that means, this is my  $x_1$  minimum, this is my  $x_1$  max, then this is my  $x_2$  max, and this is my  $x_2$  min.

Now, I want to create a grid. I want to create a grid with this. So, I want to take 1 2 3 4 5 6 point 1 2 3 4 5 6 point. So, this number of point 1, 2, 3, 4, 5, 6. So, this is where is the number. So, if I give 100; that means, 100 such point along the  $x$  axis and the same number of points should be taken along the  $y$  axis. So, it will create a matrix like a 100 cross 100; 100 cross 100 dot.

And for each it will look it looks like a point in the phase plane where we want to draw the vector field; that means we are interested in drawing the vector field the direction of vector field. How does it look like? So, I am just drawing some arbitrary vector field right. So, it is like magnetic lines of force, something like that right. So, that means, this part is the lower limit for the  $x$  axis lower limit for the  $y$  axis, then you have to create the  $\Delta x$ ; that means, the difference between these two points is my  $\Delta x$ .

The x diff and difference between y point is my delta y. So, y d diff then the figure. This is a mesh grid where we are creating this dot dot dot dot dot mesh grid is getting dot. Then you have to plug in the actual dynamical equation here which you have already considered. Then it will draw the quiver plot; that means, the quiver plot means this arrow marks it will plot automatically by computing their direction field because that if you start any initial condition at this point suppose if you start any initial condition, then the trajectory will start moving along this direction for short duration again you take a point.

Because even if you draw one point, it does not mean that trajectory will move along this line. So, trajectory direction will keep on changing. So, you have to take multiple point nearby to get more accurate trajectory information.

(Refer Slide Time: 37:28)

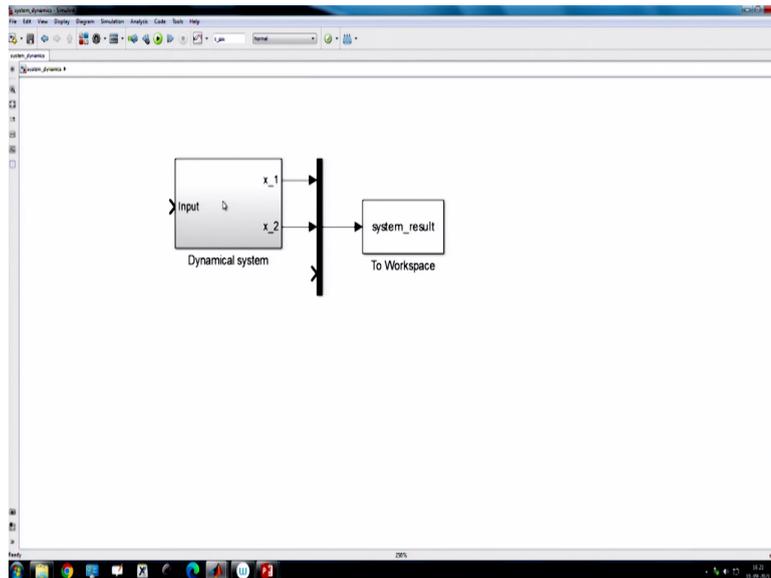
```

1  clc; close all; clear all;
2
3  %% System parameters
4  a11=-1000; a12=0; a21=0; a22=-2000;
5  q=0; b1=q; b2=0;
6  A=[a11 a12; a21 a22]; B=[b1; b2];
7
8  %% Initialization
9  x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=100e-3; T=2000e-6; D=50;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale=t*1e3;
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1)
24 plot(t_scale,x_1,'b'); hold on;

```

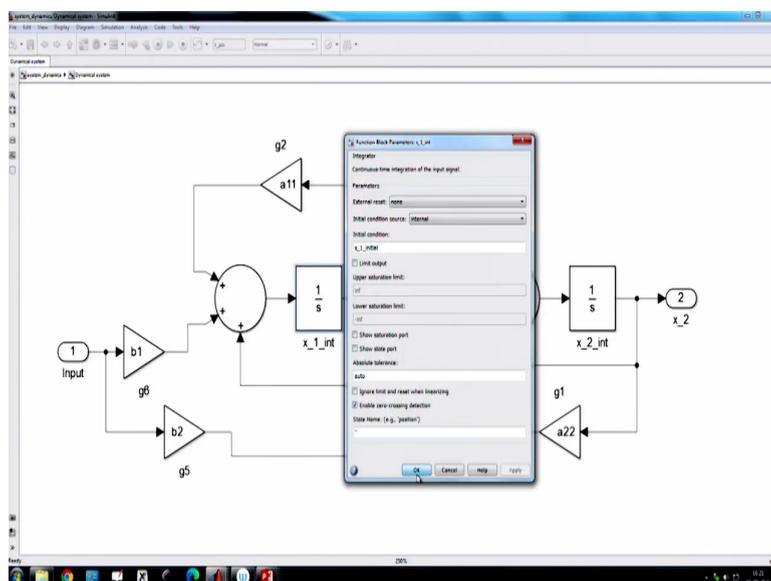
So, let us go to the MATLAB and how does it work? So, this is exactly what I have presented the whole plot.

(Refer Slide Time: 37:37)



And if we open this model.

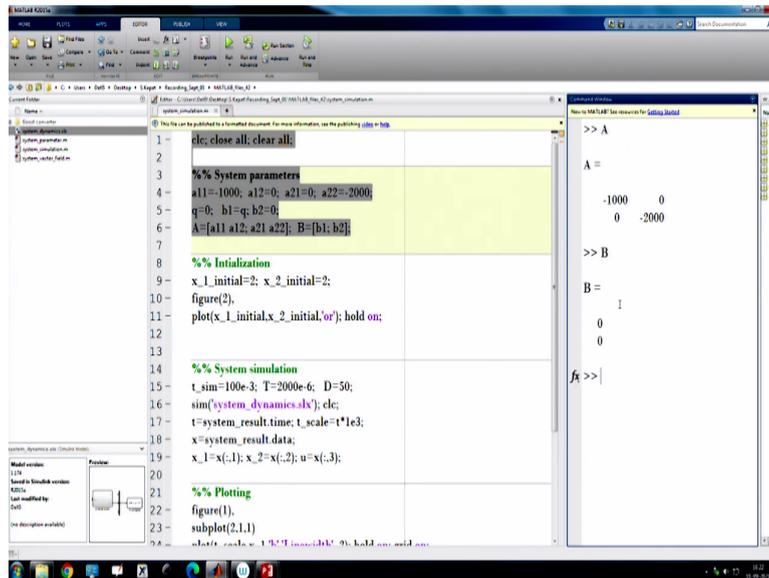
(Refer Slide Time: 37:38)



Again, if you go inside, we can plug in the initial condition here and we can plug in the initial condition here.



(Refer Slide Time: 38:01)



```
1 clc; close all; clear all;
2
3 %% System parameters
4 a11=-1000; a12=0; a21=0; a22=-2000;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=100e-3; T=2000e-6; D=50;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale=t*1e3;
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1);
24 axis('square','r','l','b','o'); hold on; end
```

```
>> A
A =
-1000    0
    0 -2000

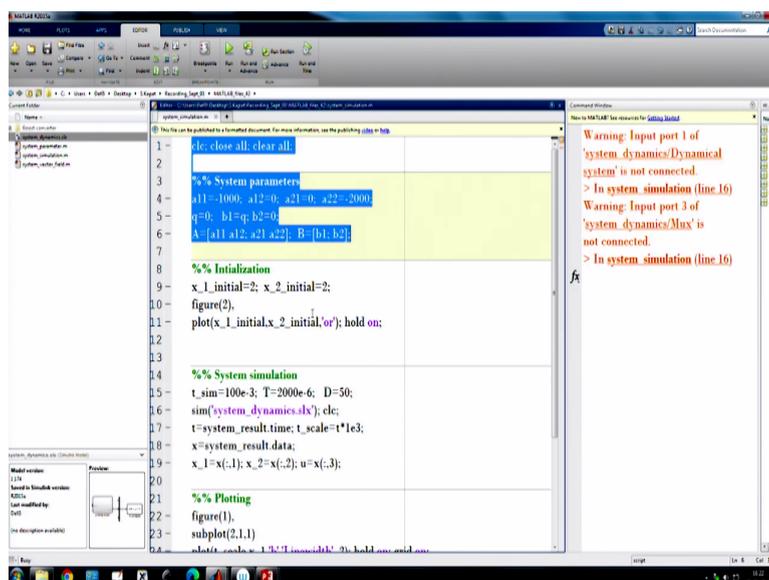
>> B
B =
    1
    0
    0

fx >>
```

And I am finding what is my A matrix. So, A matrix I have taken minus 1000 minus 2000 so that means, it is like a two decoupled system and you can check what is my B matrix.

So, B matrix 0 0. So, there is no input effect. So, we can imagine that A matrix has two diagonal term one minus 1000, another minus 2000. So, minus 1000 you can think of a 1; that means, you know a 1 and minus 2000 like a 2. So, a 1 a 2 both are negative. So, it is stable now. We want to draw the vector field let us say how does it look like.

(Refer Slide Time: 38:37)



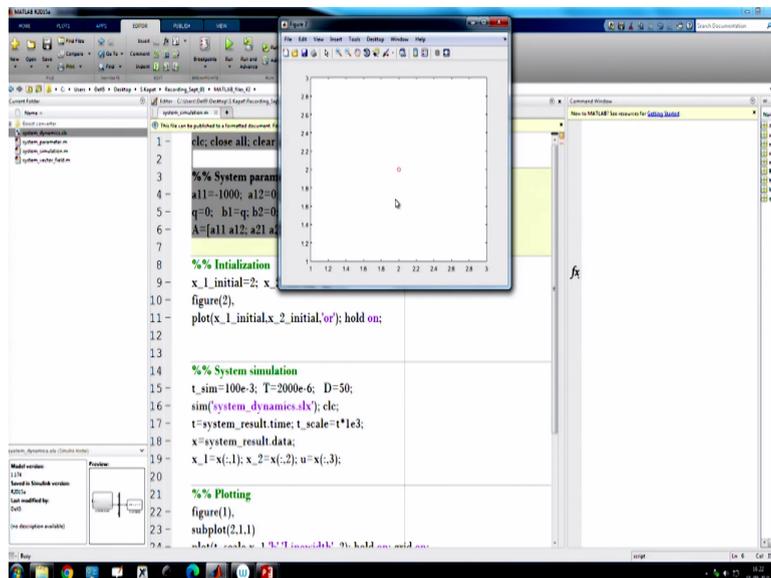
```
1 clc; close all; clear all;
2
3 %% System parameters
4 a11=-1000; a12=0; a21=0; a22=-2000;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=100e-3; T=2000e-6; D=50;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale=t*1e3;
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1);
24 axis('square','r','l','b','o'); hold on; end
```

```
Warning: Input port 1 of
'system_dynamics/Dynamical
system' is not connected.
> In system_simulation (line 16)
Warning: Input port 3 of
'system_dynamics/Mux' is
not connected.
> In system_simulation (line 16)

fx
```

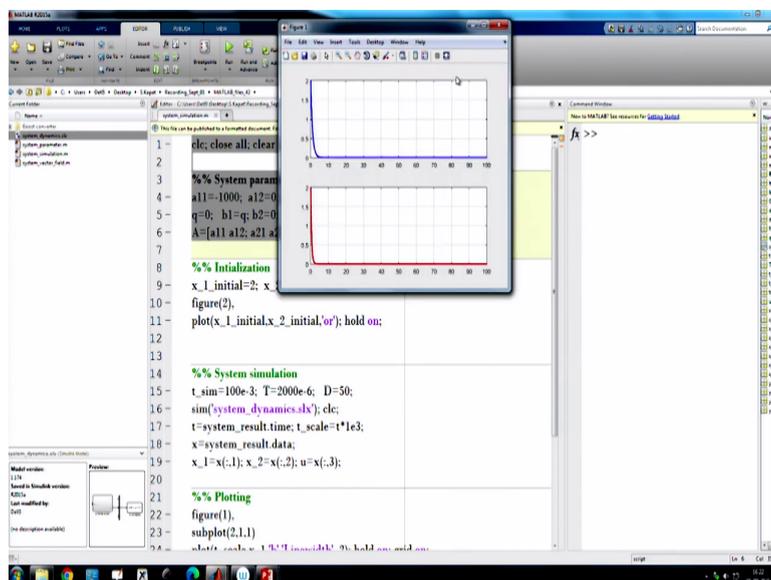
So, I want to run this program.

(Refer Slide Time: 38:39)



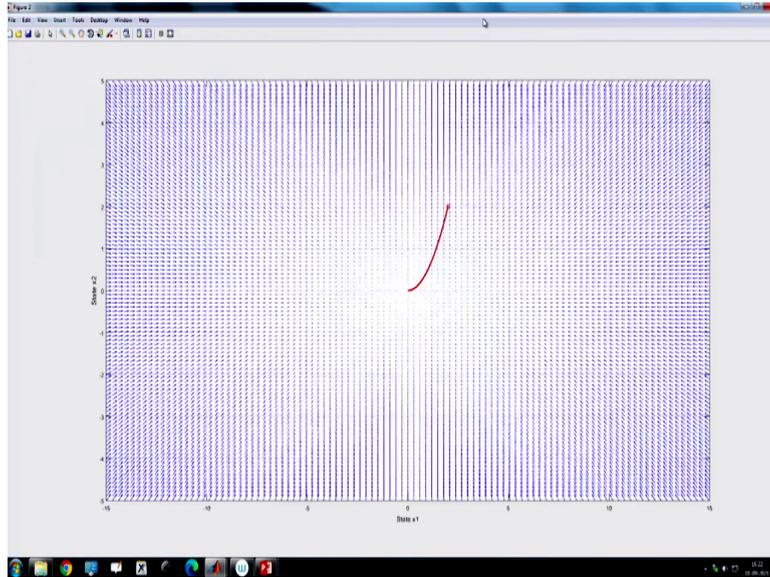
And this is the initial point I start with minus 2 and 2.

(Refer Slide Time: 38:41)



And let us see how does it move.

(Refer Slide Time: 38:44)



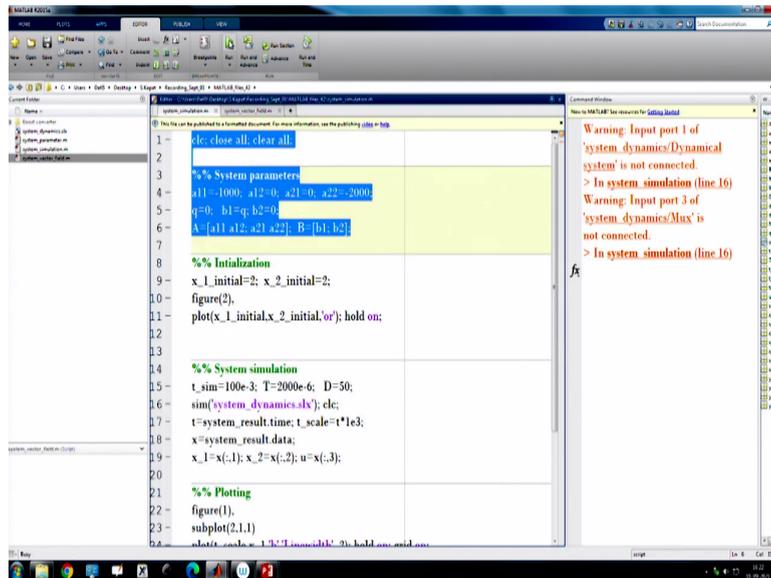
So, if you go you see I have taken 100 grid. So, let us reduce the number of grid because there are too many grids.

(Refer Slide Time: 38:52)

```
1- N_diff=10;
2- x_lower=-15; x_upper=15;
3- y_lower=-5; y_upper=5;
4-
5- x_diff=(x_upper-x_lower)/N_diff;
6- y_diff=(y_upper-y_lower)/N_diff;
7-
8- figure(2);
9- [x,y]=meshgrid(x_lower,x_upper,y_lower,y_upper);
10- d_x=(A(1,1)*x)+(A(1,2)*y)+B(1)*q;
11- d_y=(A(2,1)*x)+(A(2,2)*y)+B(2)*q;
12- quiver(x,y,d_x,d_y,1);
13- xlabel('State x1');
14- ylabel('State x2');
```

So, if I make let us say initially to understand ok. So, to understand this point.

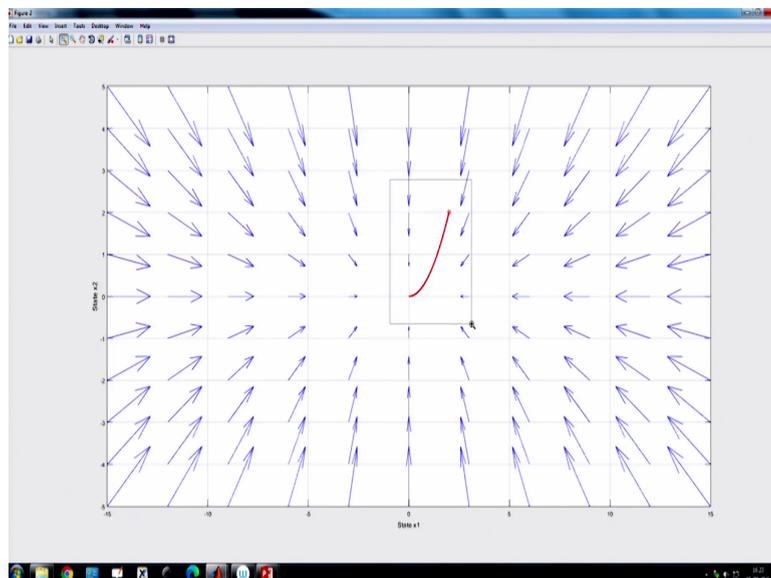
(Refer Slide Time: 39:02)



```
1- %; close all; clear all;
2-
3- %% System parameters
4- a1=-1000; a12=0; a21=0; a22=-2000;
5- q=0; b1=q; b2=0;
6- A=[a11 a12; a21 a22]; B=[b1; b2];
7-
8- %% Initialization
9- x_1_initial=2; x_2_initial=2;
10- figure(2);
11- plot(x_1_initial,x_2_initial,'or'); hold on;
12-
13-
14- %% System simulation
15- t_sim=100e-3; T=2000e-6; D=50;
16- sim('system_dynamics.slx'); clc;
17- t=system_result.time; t_scale='1e3';
18- x=system_result.data;
19- x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20-
21- %% Plotting
22- figure(1);
23- subplot(2,1,1)
```

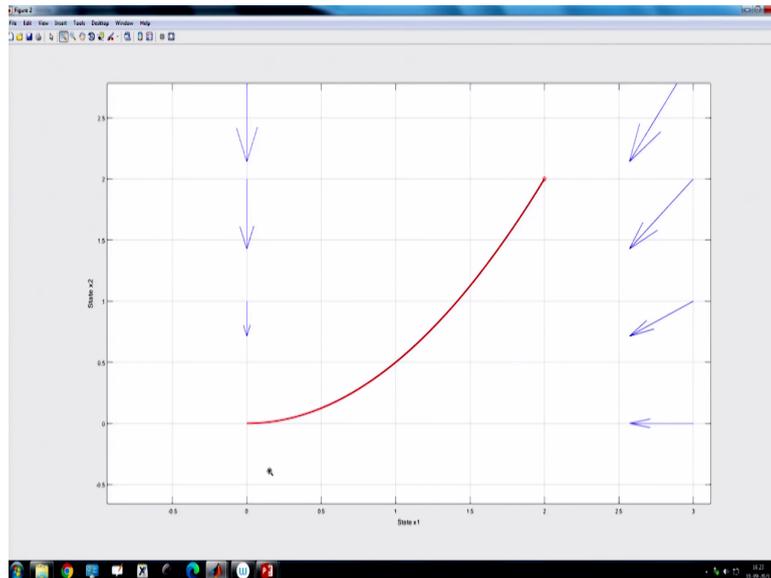
I am first running with 10 10 point grid point; that means, number of grids are less.

(Refer Slide Time: 39:07)



So, we will see if you these are the vector values. So, if you start from this point, the trajectory start moving, but it does not mean that throughout this path trajectory direction will remain same because it may keep on changing because if you see here.

(Refer Slide Time: 39:22)



The trajectory is slowly curving towards this origin and it is slowly trying to reach origin because this is a 0 0 point. So, it is starting from here and slowly coming and this direction field.

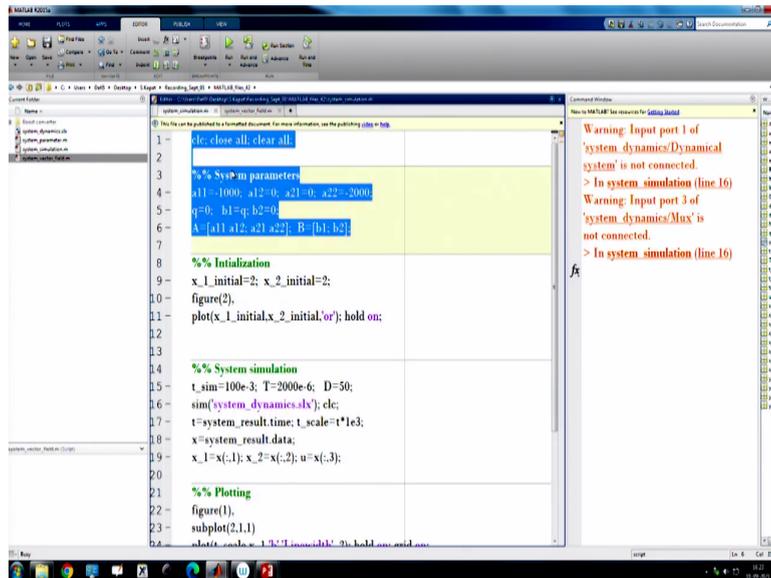
So, suppose I want to make the direction field. So, I make from minus 15 to plus 15 let us make for minus 5 to plus 5. So, if I go here.

(Refer Slide Time: 39:47)

```
1 - N_diff=10;
2 - x_lower=-5; x_upper=5;
3 - y_lower=-5; y_upper=5;
4
5 - x_diff=(x_upper-x_lower)/N_diff;
6 - y_diff=(y_upper-y_lower)/N_diff;
7
8 - figure(2);
9 - [x,y]=meshgrid(x_lower:x_diff:x_upper,y_lower:y_diff:y_upper);
10 - d_x=(A(1,1)*x)+(A(1,2)*y)+B(1)*q;
11 - d_y=(A(2,1)*x)+(A(2,2)*y)+B(2)*q;
12 - quiver(x,y,d_x,d_y,1);
13 - xlabel('State x1');
14 - ylabel('State x2');
```

And if I change the vector field like a minus 5 to plus 5 and now again, I want to draw again.

(Refer Slide Time: 39:54)



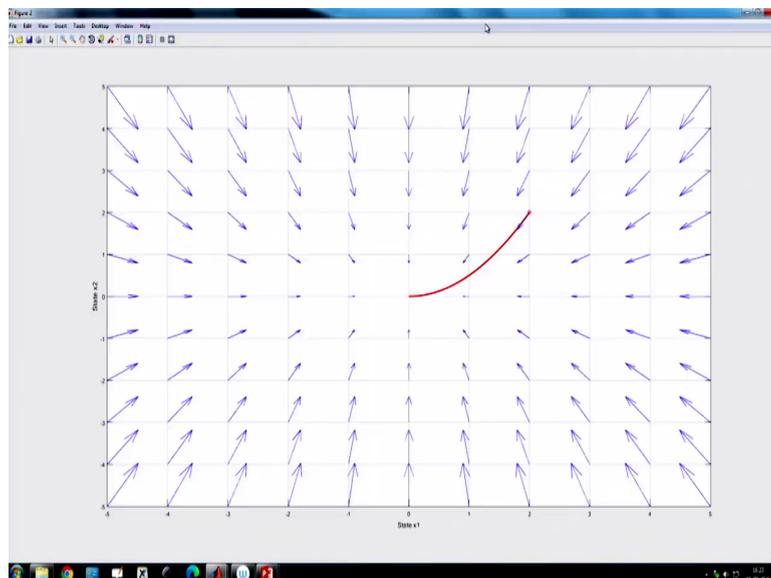
```
1= %; close all, clear all;
2=
3= %% System parameters
4= a1=-1000; a12=0; a21=0; a22=-2000;
5= q=0; b1=q; b2=0;
6= A=[a11 a12; a21 a22]; B=[b1; b2];
7=
8= %% Initialization
9= x_1_initial=2; x_2_initial=2;
10= figure(2);
11= plot(x_1_initial, x_2_initial, 'or'); hold on;
12=
13=
14= %% System simulation
15= t_sim=100e-3; T=2000e-6; D=50;
16= sim('system_dynamics.slx'); clc;
17= t=system_result.time; t_scale=1e3;
18= x=system_result.data;
19= x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20=
21= %% Plotting
22= figure(1);
23= subplot(2,1,1)
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)

Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

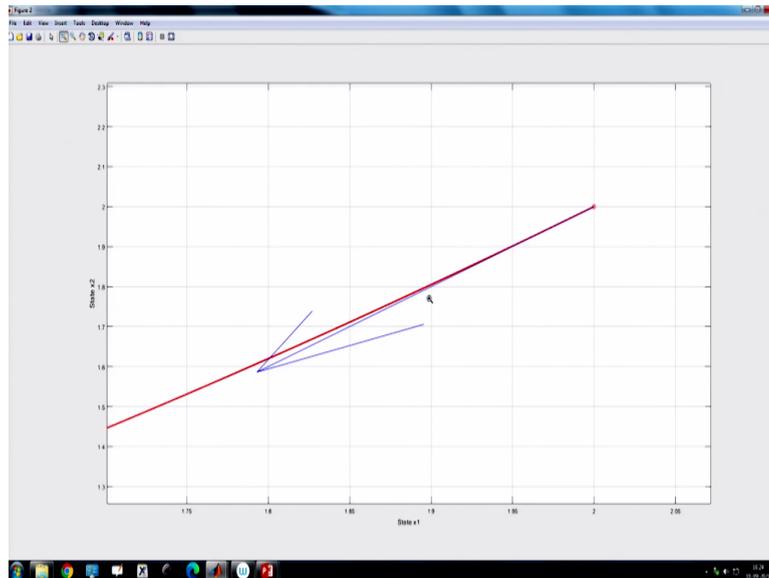
And want to see how does the trajectory look like in this plane.

(Refer Slide Time: 39:59)



So, you see initially. We have a starting point. You see if you check carefully if you draw this trajectory.

(Refer Slide Time: 40:11)



It is starting from this point and this point they are moving along the same line because this blue line indicate the derivative of the vector field or that tangent to the vector field along which the trajectory will move.

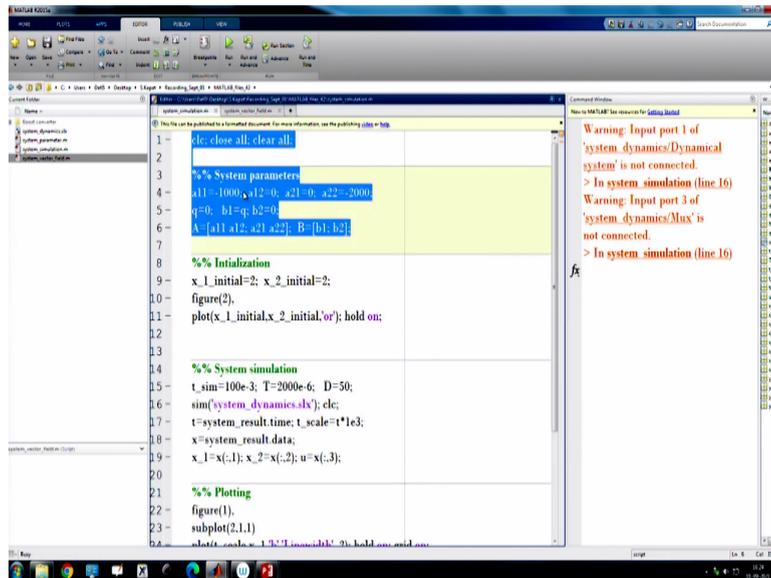
But, after some time, I mean once it passes, then it is slowly deviating; that means, in order to plot better you need to take more point. So, if we increase the number of point let us say.

(Refer Slide Time: 40:35)

```
1 - N_diff=20;
2 - x_lower=-5; x_upper=5;
3 - y_lower=-5; y_upper=5;
4
5 - x_diff=(x_upper-x_lower)/N_diff;
6 - y_diff=(y_upper-y_lower)/N_diff;
7
8 - figure(2);
9 - [x,y]=meshgrid(x_lower:x_diff:x_upper,y_lower:y_diff:y_upper);
10 - d_x=(A(1,1)*x)+(A(1,2)*y)+B(1)*q;
11 - d_y=(A(2,1)*x)+(A(2,2)*y)+B(2)*q;
12 - quiver(x,y,d_x,d_y,1);
13 - xlabel('State x1');
14 - ylabel('State x2');
```

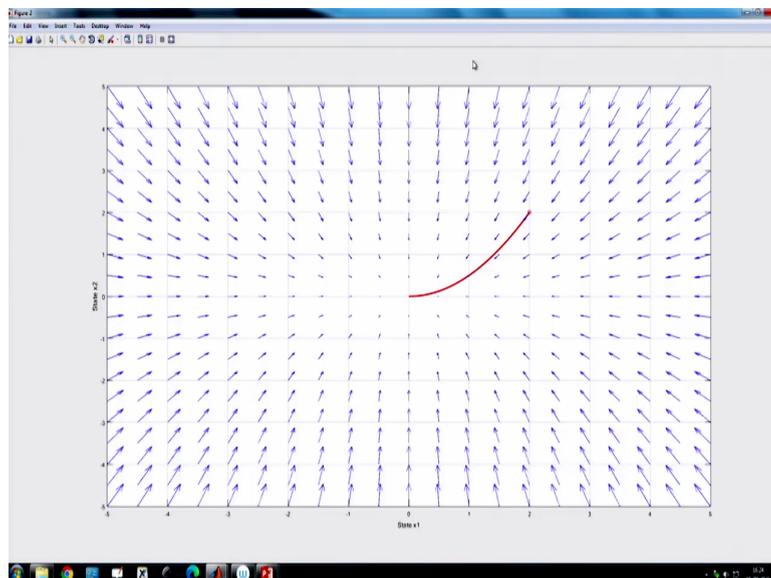
Instead of 10 let us say we make 20 and see whether we are.

(Refer Slide Time: 40:42)



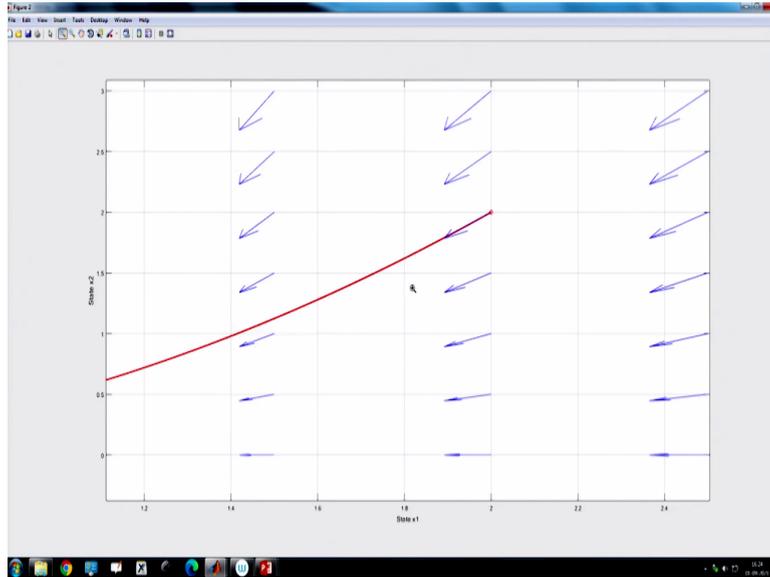
We can get a better you know representation of the vector field or not.

(Refer Slide Time: 40:47)



So, you see we have taken a nearby, but not exactly on the same path.

(Refer Slide Time: 40:51)



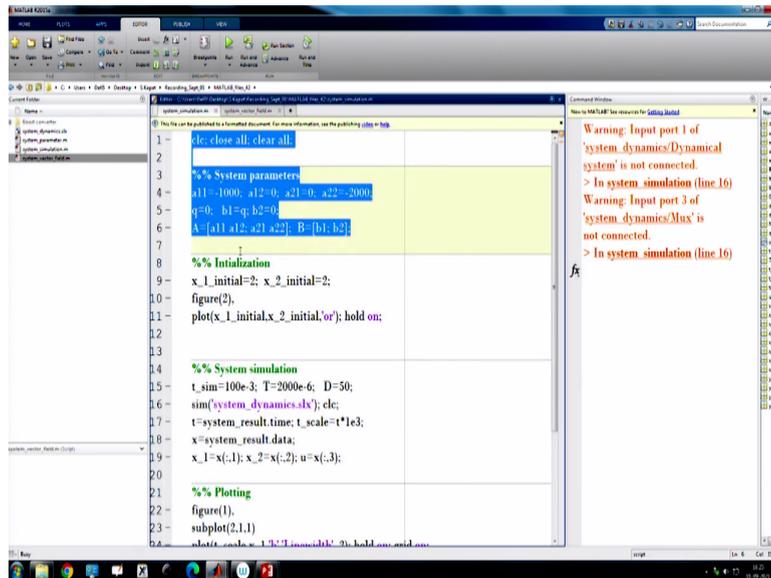
It is slow. If you could have taken here, you will get a better representative.

(Refer Slide Time: 40:59)

```
1 - ! N_diff=50;
2 - x_lower=-5; x_upper=5;
3 - y_lower=-5; y_upper=5;
4
5 - x_diff=(x_upper-x_lower)/N_diff;
6 - y_diff=(y_upper-y_lower)/N_diff;
7
8 - figure(2);
9 - [x,y]=meshgrid(x_lower,x_diff,x_upper,y_lower,y_diff,y_upper);
10 - d_x=(A(1,1)*x)+(A(1,2)*y)+B(1)*q;
11 - d_y=(A(2,1)*x)+(A(2,2)*y)+B(2)*q;
12 - quiver(x,y,d_x,d_y,b);
13 - xlabel('State x1');
14 - ylabel('State x2');
15
```

So, we can take you know maybe 50. So, because we are increasing the grid point.

(Refer Slide Time: 41:05)

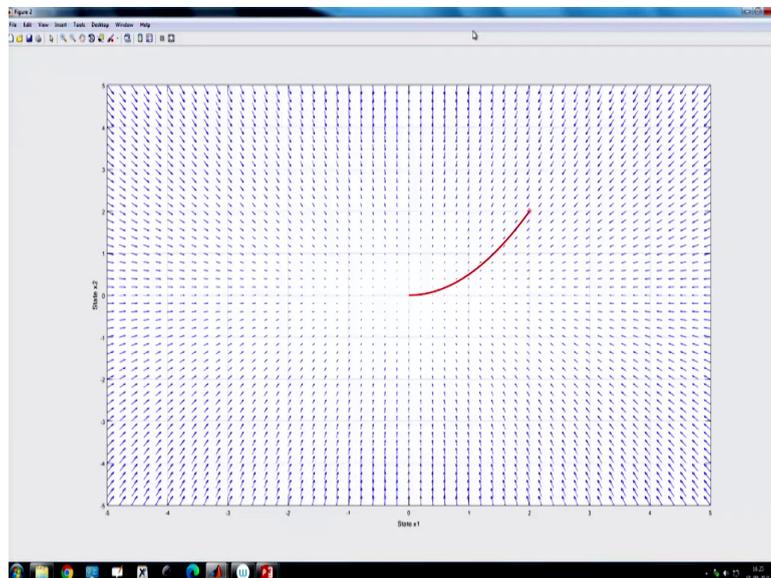


```
1 clc; close all; clear all;
2
3 %% System parameters
4 a11=-1000; a12=0; a21=0; a22=-2000;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=100e-3; T=2000e-6; D=50;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale=*1e3;
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1)
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)  
Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

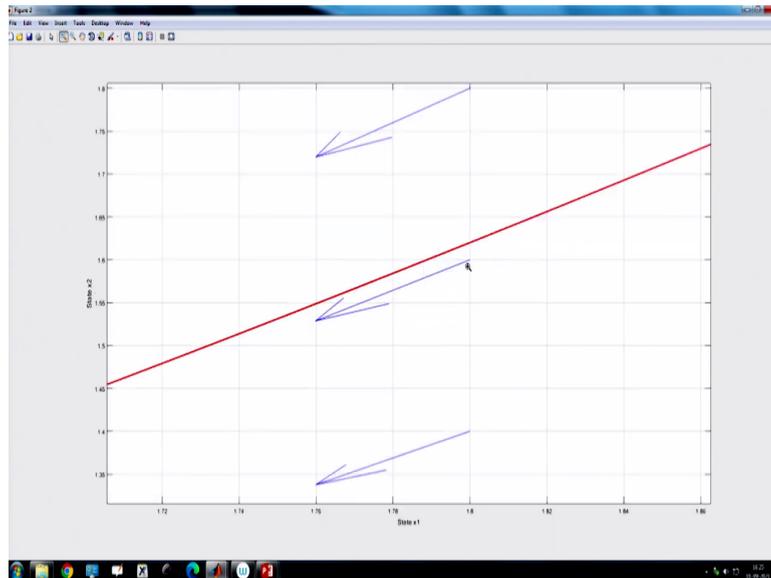
So, we are taking more and more vector that means point to draw the vector field. So, this can give you a better representative of the vector field.

(Refer Slide Time: 41:11)



So, if you go there; that means you see this point started at the same and if you take this particular path. So, we are taking nearby. So, it is showing the direction of vector field.

(Refer Slide Time: 41:27)



That means, even if we take the vector field here. So, this close by vector field will be almost nearly parallel ok; that means, this is how it shows the trajectory of the vector field. Now, we can check with multiple point that means, we can check with now.

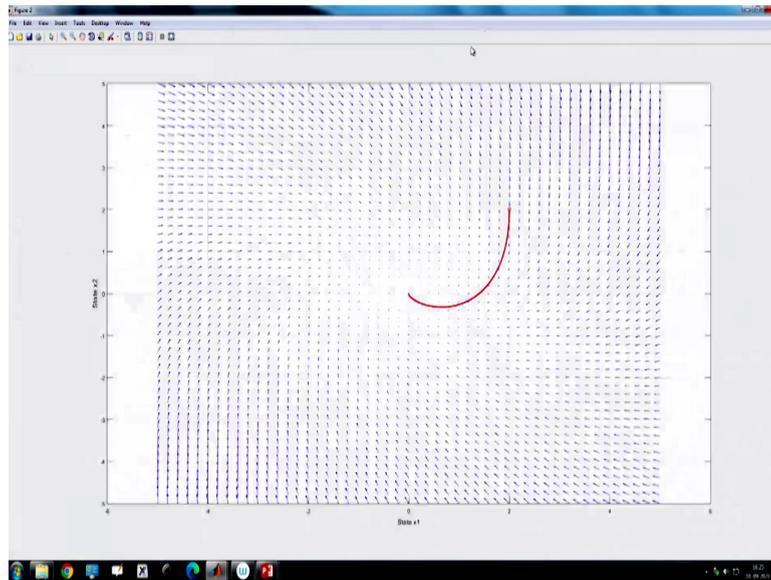
(Refer Slide Time: 41:43)

```
1: clc; close all; clear all;
2:
3: %% System parameters
4: a11=-1000; a12=1000; a21=-1000; a22=2000;
5: q=0; b1=q; b2=0;
6: A=[a11 a12; a21 a22]; B=[b1; b2];
7:
8: %% Initialization
9: x_1_initial=2; x_2_initial=2;
10: figure(2);
11: plot(x_1_initial,x_2_initial,'or'); hold on;
12:
13:
14: %% System simulation
15: t_sim=100e-3; T=2000e-6; D=50;
16: sim('system_dynamics.slx'); clc;
17: t=system_result.time; t_scale='1e3';
18: x=system_result.data;
19: x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20:
21: %% Plotting
22: figure(1);
23: subplot(2,1,1)
24: subplot('rows(1,3)'); hold on; end
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)  
Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

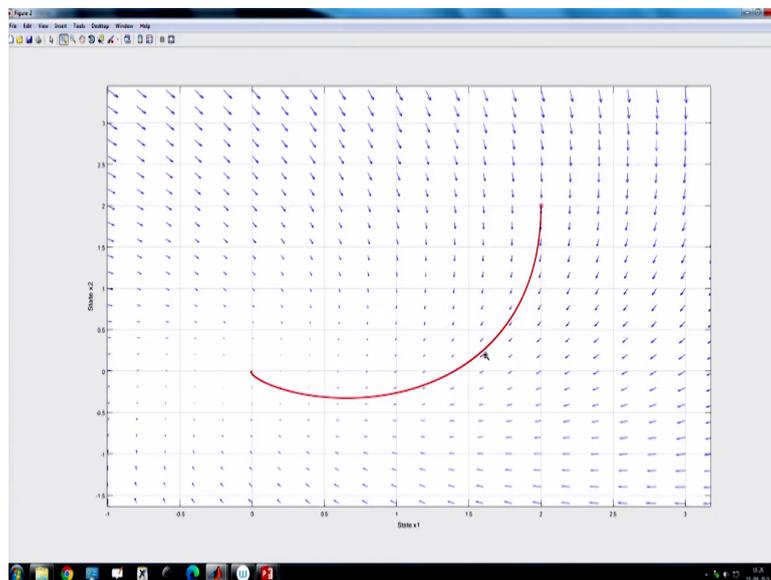
Let us say 1000 and we can give minus 1000. So, let us see how does it looks like I want to show interesting point and how does it looks like. So, this is another point I want to show.

(Refer Slide Time: 42:01)



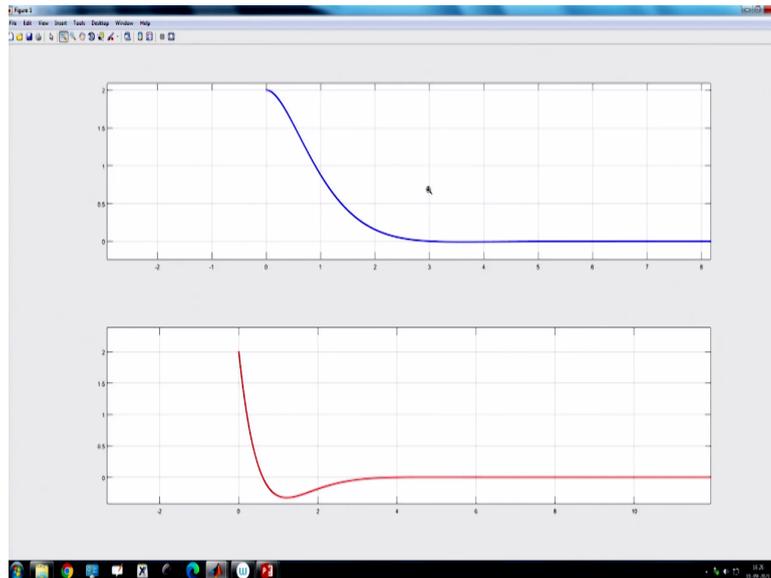
Now, trajectory is curving towards this right.

(Refer Slide Time: 42:06)



So, it is actually changing and it is curving.

(Refer Slide Time: 42:13)



And you see here you will get some overshoot undershoot because it is crossing this point for some time crossing this point. It might also cross a little bit so; that means some overshoot undershoot effect will be there, but if we could take the earlier one.

(Refer Slide Time: 42:31)

```

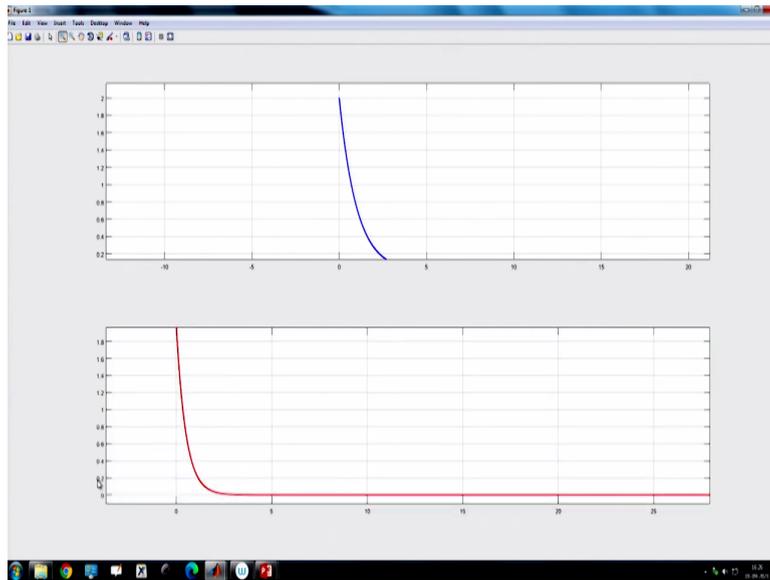
1 = clear; close all; clear all;
2
3 %% System parameters
4 a11=-1000; a12=0; a21=0; a22=-2000;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=100e-3; T=2000e-6; D=50;
16 sim('system_dynamics.slx'); clear;
17 t=system_result.time; t_scale='1e3';
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1)
24 subplot(2,1,2)

```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected.  
 > In system\_simulation (line 16)  
 Warning: Input port 3 of 'system\_dynamics/Mux' is not connected.  
 > In system\_simulation (line 16)

Then there will be no overshoot understood because the earlier was a decoupled system. Now we are talking about a coupled system. So, if you draw the earlier one, it is like a there is no overshoot.

(Refer Slide Time: 42:46)



It is like exponentially decaying function. So, this is decaying right. So, this is a decaying function decaying function right. So, we understood how to draw a vector field.

(Refer Slide Time: 43:04)

*Drawing Vector Field – Geometric Perspectives*

$$\dot{x} = Ax = f_x(x), \quad A = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} v_x & v_y \end{bmatrix} \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} a_1 v_x & a_2 v_y \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

*Handwritten notes:* "Identity matrix" with a bracket under the first matrix; "2x1" and "2x1" written above the second and third matrices respectively; a purple circle around the  $v_x$  and  $v_y$  terms in the second matrix.

Now, we want to see can we imagine how does the vector field looks like. So, we started with this point  $A \times f_x$  and again we consider that  $v_x$  and  $v_y$  are the two basis vector right. So, you can write because this is an identity matrix. This is like our identity matrix and this basis vector  $v_x$  has a dimension of 2 cross 1, this also dimension of 2 cross 1. So, now, I can write for this particular case a 1  $v_x$  a 2  $v_x$  right.

(Refer Slide Time: 43:46)

*Vector Field of Dynamical System*

$$\dot{x} = \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}}_A x = \underline{f_x(x)}$$

$$\underline{f_x(x)} = \begin{bmatrix} a_1 v_x & a_2 v_y \end{bmatrix} x$$

$$= \begin{bmatrix} \underline{f_{1x}} & f_{2x} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$f_{1x} = \underline{a_1 v_x}$$

Now, I want to draw the vector field. So, I told you in order to draw the vector field. So,  $f_x$  function this is a  $v_x$  is the basis vector with the scaling factor of a 1,  $v_y$  is a basis vector of the scaling factor of a 2 and  $x$  is the point where we are trying to draw the vector  $b$ .

So, you need to specify the point where we want to draw the tangent. Suppose you want to draw a tangent here whether it will come towards this which you have shown in the MATLAB or where it will go towards this. So, that can be analysed by drawing this vector field. So, let us first draw for  $f_{1x}$  it is nothing, but a  $1 v_x$ . So, this is my  $v_x$  and a  $1 v_x$  is this one.

(Refer Slide Time: 44:30)

### Vector Field of Dynamical System

$$\dot{x} = \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}}_A x = f_x(x)$$

$$f_x(x) = \begin{bmatrix} a_1 v_x & a_2 v_y \end{bmatrix} x$$

$$= \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$f_{1x} = a_1 v_x \quad f_{2x} = a_2 v_y$

Now, we are going for  $f_{1x}$  and  $f_{2x}$ ; that means, first this is my  $f_{1x}$  that we have already discussed and the other one is  $f_{2x}$  which is a two time  $v_x v_y$ .

(Refer Slide Time: 44:45)

### Drawing Vector Field in 2-D Plane

$$f_x(x_0) = \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$= \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$f_{x_0} = f_{1x}(x_{10}) + f_{2x}(x_{20})$$

$f_x(x_0) \triangleq f_{x_0}$

Then what we have to obtain? We are trying to find the vector field at  $x$  equal to  $x_0$ . So, it has this  $f_{1x}$ ,  $f_{2x}$  we have drawn and  $x_0$  that is given; that means, it has  $x_{10}$  and  $x_{20}$ ; that means, you know we are talking about this point this is  $x_{10}$  component and this is  $x_{20}$  component right.

So; that means,  $f \cdot x_0$  can be written as  $f_1 \cdot x_1$  vector into a scaling factor this,  $f_2$  as vector scaling factor of this. I am writing this vector to be my  $f_1 \cdot x_1$  into  $x_1 \cdot 0$ . I am writing this vector to be this vector to be I am talking about this vector to be  $f_2 \cdot x_2$  into  $x_2 \cdot 0$  and their resultant vector is the one which is my this one which is the vector computed at this point; that means, this is the direction of the vector.

But, remember we need to draw the vector field here because we wanted to find the direction of the vector field at this point. So, ultimately we got the vector field direction. So, this vector field will be parallel to this.

(Refer Slide Time: 46:05)

*Drawing Vector Field in 2-D Plane*

$$f_{x_0} = \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} x_0$$

$$= \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$f_{x_0} = f_{1x} x_{10} + f_{2x} x_{20}$$

That means, whatever we obtain it is the parallel field right because we first draw the vector and then because it is a parallel vector right because we want to find the vector  $v$  at this point.

(Refer Slide Time: 46:18)

*Drawing Vector Field in 2-D Plane*

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}}_A \mathbf{x} = \mathbf{f}_x(\mathbf{x})$$

$$\mathbf{f}_x(\mathbf{x}^1) = \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \mathbf{x}^1 \triangleq \mathbf{f}_{x^1}$$

$$\mathbf{x}^1 = \mathbf{x}(t_1) = \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} \quad \mathbf{f}_{x^1} = \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} = f_{1x}x_{11} + f_{2x}x_{21}$$

Similarly, we want to find the vector field at this point. We need to know what is my this point location as if this location is computed after trajectory moves towards I mean after time  $t_1$  interval and you got  $x_1$  on  $x_2$  component because this one is my  $x_{11}$  and; that means, this is my  $x_{11}$  comma  $x_{21}$  component.

So, then we can get this vector; this vector is nothing, but  $f_{1x}$  into  $x_{11}$   $f_{2x}$  into  $x_{21}$  and again we need to draw a parallel line to this because this is we are computing at this point right.

(Refer Slide Time: 46:58)

*Drawing Vector Field in 2-D Plane*

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}}_A \mathbf{x} = \mathbf{f}_x(\mathbf{x})$$

$$\mathbf{f}_x(\mathbf{x}^1) = \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \mathbf{x}^1 \triangleq \mathbf{f}_{x^1}$$

$$\mathbf{x}^1 = \mathbf{x}(t_1) = \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} \quad \mathbf{f}_{x^1} = \begin{bmatrix} f_{1x} & f_{2x} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} = f_{1x}x_{11} + f_{2x}x_{21}$$

So, by that way we can draw the vector field.

(Refer Slide Time: 47:02)

**Example 1: Vector Field**

$$\mathbf{x}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{a_1 t} & 0 \\ 0 & e^{a_2 t} \end{bmatrix} \mathbf{x}_0$$

$\mathbf{v}_x$     $\mathbf{v}_y$

$\alpha_2 \mathbf{v}_y$     $\mathbf{x}_0$     $a_1 < 0$

$\alpha_1 \mathbf{v}_x$     $\alpha_1 \mathbf{v}_x$

$a_1 \mathbf{v}_x + a_2 \mathbf{v}_y = \mathbf{x}_0$   
 $a_2 = 0$

> What happens if  $\mathbf{x}_0$  is located either on  $\mathbf{v}_x$  or on  $\mathbf{v}_y$  ?

$\mathbf{x}(t) = e^{a_1 t} \alpha_1 \mathbf{v}_x$     $\alpha_1 \mathbf{v}_x = \mathbf{x}_0$

> Invariant vector field

$\dot{\mathbf{x}}(t) = a_1 e^{a_1 t} \alpha_1 \mathbf{v}_x = a_1 \mathbf{x}(t)$

Now, this vector field example we have said  $\mathbf{v}_x$  and  $\mathbf{v}_y$  are the basis vector and we know how to draw the initial condition. What happens if you take  $\mathbf{x}_0$  in either on this basis vector or on this basis vector? So, if you take  $\mathbf{x}_0$  on that basis vector, you will find alpha; that means, our original equation was  $\alpha_1 \mathbf{v}_x + \alpha_2 \mathbf{v}_y = \mathbf{x}_0$  since we have chosen  $\mathbf{x}_0$  along this  $\mathbf{v}_x$  axis.

So,  $\alpha_2$  must be 0 in this case; that means, you will only have  $\alpha_1 \mathbf{v}_x = \mathbf{x}_0$ . So, your  $\mathbf{x}(t)$  will be simply  $e^{a_1 t} \alpha_1 \mathbf{v}_x$  because  $\alpha_2$  is 0. If you differentiate, you will get  $\dot{\mathbf{x}}(t) = a_1 \mathbf{x}(t)$ . The trajectory will continue to remain on this line. If you choose any point starting point here, it will remain here provided that your  $a_1$  is less than 0.

So, it will because the derivative of this vector is on the line; that means, the point will move the vector field direction will be along this line since  $a_1$  is negative it is towards this side left side right to left. So, the point will slowly move towards the origin.

So, it is called invariant vector field because if you start the initial condition on that line of the vector, the point will move along the line, it will not go out of the line. So, it will be along the vector. This is an invariant vector field and we will see this will actually lead to what is the concept of eigenvector ok but we are not talking here now.

(Refer Slide Time: 48:40)

**Example 2: Vector Field of Coupled System**

$$\mathbf{x}(t) = e^{a_1 t} \alpha_1 \mathbf{v}_x + e^{a_2 t} \alpha_2 \mathbf{v}_y$$

$$\alpha_1 \mathbf{v}_x + \alpha_2 \mathbf{v}_y = \mathbf{x}_0$$

$$\mathbf{v}_x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{v}_y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

➤ What happens if  $\mathbf{x}_0$  is located either on  $\mathbf{v}_x$  or on  $\mathbf{v}_y$ ?

$$\mathbf{x}(t) = e^{a_1 t} \alpha_1 \mathbf{v}_x \quad \alpha_1 \mathbf{v}_x = \mathbf{x}_0$$

➤ Invariant vector field

$$\dot{\mathbf{x}}(t) = a_1 e^{a_1 t} \alpha_1 \mathbf{v}_x = a_1 \mathbf{x}(t)$$

Now, if you take a coupled system, we have taken two vectors. Then for this again if we take any initial condition on this any of this vector. Suppose, we took the initial condition let us say here then if a 1 is less than 0 again this point will move along this side it will not come out of this line and it will slowly approach towards origin and this is my vector direction  $\mathbf{v}_x$  right again it is an invariant vector field.

So, whether the basis vector these are the basis vector whether they are orthogonal or not, but they must be linearly independent. If you choose any initial condition on any of this basis vector along that line, then it will remain on the line whether it will come towards origin or go away depending upon the stability property, but it will be on the line and this is the invariant vector field.

(Refer Slide Time: 49:34)

**Comparative Study - Vector Fields**

**System 1**

$$v_x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v_y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

**System 2**

$$v_x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad v_y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$x(t) = e^{\alpha_1 t} \alpha_1 v_x + e^{\alpha_2 t} \alpha_2 v_y$   
 $\alpha_1 v_x + \alpha_2 v_y = x_0$

So, if we compare the first case, this is the vector orthogonal, but they are not orthogonal right. So, these are the two diagrams that we set. Then we can map any  $x_0$  using this basis vector for system 2 as well as system 1.

(Refer Slide Time: 49:54)

**Comparative Study - Vector Fields**

**System 1**

$$v_x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v_y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

**System 2**

$$v_{x1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad v_{y1} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$x(t) = e^{\alpha_1 t} \alpha_1 v_x + e^{\alpha_2 t} \alpha_2 v_y$        $z(t) = M \times x(t)$

$= e^{\alpha_1 t} \alpha_1 M v_x + e^{\alpha_2 t} \alpha_2 M v_y$

$= e^{\alpha_1 t} \alpha_1 v_{x1} + e^{\alpha_2 t} \alpha_2 v_{y1}$

$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$

$v_{y1} = M v_y$

$v_{x1} = M v_x$

And then we want to combine like you know  $x(t)$  and let us see it is  $z(t)$ . Suppose we applied. So, this is something like that. Suppose we take an image. This image like this suppose you take an image the image is a circle right and this is a plane  $x_1 \times x_2$  plane  $x_1 \times z_1$  plane and we

are looking from vertically from like our viewpoint is from the you know three dimensional, it is like along the z axis we are looking at.

So, it will look like a circle, but now suppose we have changed the angle. Suppose we are looking from an inclination then the axis may appear slightly different the axis may appear slightly different and it may appear something like this and that we call it as a  $z_1 z_2$  because we are looking from some angle and then you will find this particular ball will look like an ellipse right because bent it because we are looking from another angle.

So, when you look from vertically it will look like a circle, but if you look from an angle it will look like an ellipse right. So, that axis. So, this is exactly what z. So, I can take z to be a combination of  $z_1, z_2$  which is an image version by either moving the image keeping the eye position same by changing the angle of this plane or by changing our eye angle itself. So, in each case, as if we are kind of by movement, we are making some image processing or some image transformation from here.

So, we are multiplying if operator M which is changing the image direction, either your view direction. So, it is ultimately the relative position I mean. It does not matter because if both the image plane and the vision plane change together, they still appear like a vertical. Then there is no problem, but suppose if one of them related to the other changes, then as if we are making like an operator M where x was our original one and the z is the 1 by operator plane. If you multiply, then this will resemble like this kind of thing where  $v_x 1$  and  $v_y 1$ .

So, here  $v_x 1$  is nothing but operator into the previous basis vector. So, this is a change in the basis vector by means of the projection and  $v_y 1$  is m times  $v_y$  again operator same operator.

(Refer Slide Time: 52:38)

**Comparative Study - Vector Fields**

<p><b>System 1</b></p> $\mathbf{x}(t) = e^{\alpha_1 t} \alpha_1 \mathbf{v}_x + e^{\alpha_2 t} \alpha_2 \mathbf{v}_y$ $\mathbf{v}_x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{v}_y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	<p><b>System 2</b></p> $\mathbf{z}(t) = e^{\alpha_1 t} \alpha_1 \mathbf{v}_{z1} + e^{\alpha_2 t} \alpha_2 \mathbf{v}_{y1}$ $\mathbf{v}_{z1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{v}_{y1} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
--	--

$$\underline{\mathbf{v}_{z1}} = M \underline{\mathbf{v}_x} \quad \underline{\mathbf{v}_{y1}} = M \underline{\mathbf{v}_y} \Rightarrow \begin{bmatrix} \underline{\mathbf{v}_{z1}} & \underline{\mathbf{v}_{y1}} \end{bmatrix} = M \begin{bmatrix} \underline{\mathbf{v}_x} & \underline{\mathbf{v}_y} \end{bmatrix} = M \underline{I_{den}}$$

$$M = \begin{bmatrix} \downarrow & \downarrow \\ \mathbf{v}_{z1} & \mathbf{v}_{y1} \\ \uparrow & \uparrow \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$


So, it is actually changing the angle. So, system 1 system 2 operator. So, if we write  $v \times 1$   $M$   $v \times v$   $i$   $1$   $M$   $v$   $y$ . So, if you combine  $v \times 1$   $v \times 2$   $M$  into  $v \times x$ . So, this identity matrix. So, it turns out this operator, which is a square matrix consists of  $v \times 1$  and  $v \times 2$  which is the inclined angle right. They are basis vector. So, ultimately this is and you will find soon that this is nothing, but these are the eigenvector of this system which we are going to deal with.

(Refer Slide Time: 53:14)

**Comparative Study - Vector Fields**

<p><b>System 1</b></p> $\mathbf{x}(t) = e^{\alpha_1 t} \alpha_1 \mathbf{v}_x + e^{\alpha_2 t} \alpha_2 \mathbf{v}_y$	<p><b>System 2</b></p> $\mathbf{z}(t) = e^{\alpha_1 t} \alpha_1 \mathbf{v}_{z1} + e^{\alpha_2 t} \alpha_2 \mathbf{v}_{y1}$ $\mathbf{z}(t) = M \left( e^{\alpha_1 t} \alpha_1 \mathbf{v}_x + e^{\alpha_2 t} \alpha_2 \mathbf{v}_y \right) = M \mathbf{x}(t)$
--	---

$$\boxed{\mathbf{z} = M \mathbf{x}} \Rightarrow \mathbf{x} = M^{-1} \mathbf{z}$$

$$M = \begin{bmatrix} \mathbf{v}_{z1} & \mathbf{v}_{y1} \end{bmatrix}$$

$M$  Full rank square matrix

$$M^{-1} = \begin{bmatrix} \mathbf{v}_{x1} & \mathbf{v}_{y1} \end{bmatrix}$$


So, what is the system transformation? That means, this image processing is changing angle. By means of an operator  $z$  equal to  $M$  time  $x$  or you can write  $x$  equal to  $M$  inverse  $z$  and this

inverse exist because we have chosen M which consist of two vector v x 1 and v y 1 and both the vectors are linearly independent as a result M must be I mean M will be invertible because the rank will be full rank because it consist of two vectors a two-dimensional vector which are linearly independent. Same it is a full length matrix square matrix.

(Refer Slide Time: 53:52)

**Comparative Study - Vector Fields**

<p><b>System 1</b></p> $\mathbf{x}(t) = e^{\alpha_1 t} \mathbf{v}_x + e^{\alpha_2 t} \mathbf{v}_y$ $\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}}_{A_{dec}} \times \underbrace{\begin{bmatrix} e^{\alpha_1 t} x_{10} \\ e^{\alpha_2 t} x_{20} \end{bmatrix}}_{\mathbf{z}}$ $\dot{\mathbf{x}}(t) = A_{dec} \mathbf{x}, A_{dec} = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$	<p><b>System 2</b></p> $\mathbf{z}(t) = e^{\alpha_1 t} \alpha_1 \mathbf{v}_{x1} + e^{\alpha_2 t} \alpha_2 \mathbf{v}_{y1}$ $\dot{\mathbf{z}} = \underline{M} \dot{\mathbf{x}} = \underline{M A_{dec} M^{-1}} \mathbf{z}$ $= \underline{A_{coup}} \mathbf{z}$ $M = \begin{bmatrix} \mathbf{v}_{x1} & \mathbf{v}_{y1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$
---	---

$A_{coup} = M A_{dec} M^{-1}$



Then, if we obtain x dot t it is the representation of the autonomous system for system 1 and if you obtain z dot t, you will get a couple and there because of the transformation. So, your couple system here is nothing, but M times M decouple system into M inverse. So, this is an image transformation has changed the system matrix ok.

(Refer Slide Time: 54:18)

**Comparative Study - Vector Fields**

<p><b>System 1</b></p> $\dot{\mathbf{x}}(t) = A_{dec} \mathbf{x}$ $A_{coup} = M A_{dec} M^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{-1}$ $M = \begin{bmatrix} \mathbf{v}_{x1} & \mathbf{v}_{y1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$	<p><b>System 2</b></p> $\dot{\mathbf{z}} = A_{coup} \mathbf{z}$
--	---



And if you decouple one, you will get this combination. Now, since we have not put any value here. So, we will see.

(Refer Slide Time: 54:29)

**Comparative Study - Vector Fields**

<p><b>System 1</b></p> $\dot{\mathbf{x}}(t) = A_{dec} \mathbf{x}$	<p><b>System 2</b></p> $\dot{\mathbf{z}} = A_{coup} \mathbf{z}$
---	---

For  $a_1 = -2$ ,  
 $a_2 = -3$

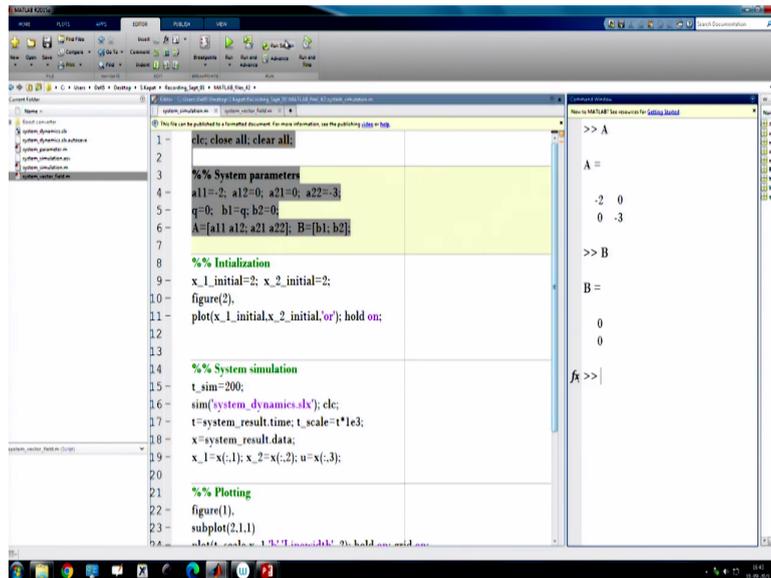
$$A_{coup} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} -1 & -1 \\ 2 & -4 \end{bmatrix}$$

*A<sub>dec.</sub>*                      *A<sub>coup</sub>*

Find eigenvalues and eigenvectors?

And we want to see what will happen in this particular case. So, suppose we have taken a 1 to be minus 2 minus 3 and the coupled system will be minus 1 minus 1. So, we want to see the vector field in decoupled system. So, this is my decoupled matrix and this is my coupled matrix. So, whatever we discussed now, we want to see what happens in this case.

(Refer Slide Time: 54:53)



```
1 clc; close all; clear all;
2
3 %% System parameters
4 a11=-2; a12=0; a21=0; a22=-3;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13 %% System simulation
14 t_sim=200;
15 sim('system_dynamics.slx'); clc;
16 t=system_result.time; t_scale='1e3';
17 x=system_result.data;
18 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
19
20 %% Plotting
21 figure(1);
22 subplot(2,1,1)
23 plot(t_scale*x_1,t_scale*x_2,'or'); hold on; end;
```

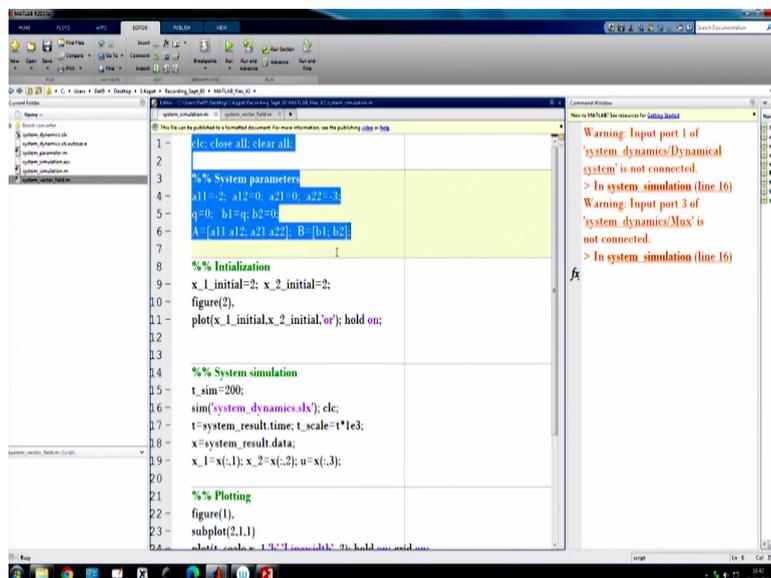
```
>> A
A =
-2 0
0 -3

>> B
B =
0
0

fx >>|
```

So, let us go to the MATLAB simulation here. We are considering minus 2 0 0. So, suppose let us run this and I will show you that if we run it A matrix and B matrix. So, what is our A matrix? Minus 2 minus 3 and what is my B matrix? 0. So, if I draw, run it.

(Refer Slide Time: 55:19)



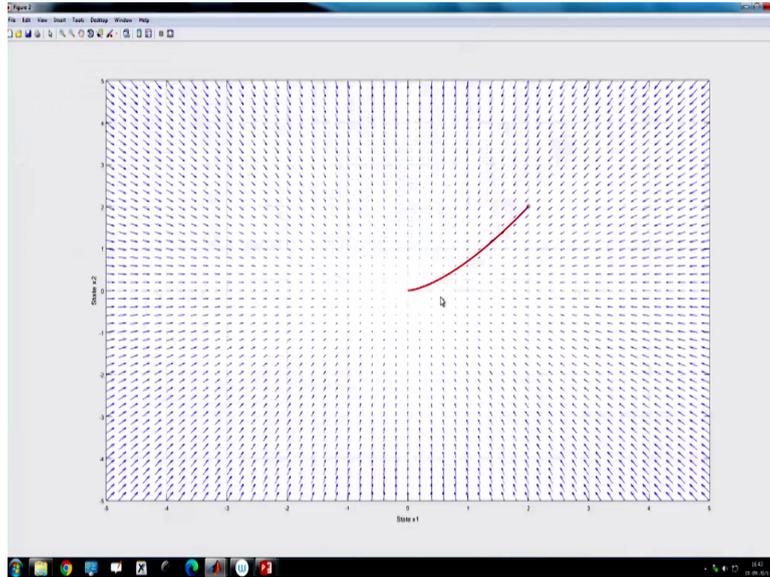
```
1 clc; close all; clear all;
2
3 %% System parameters
4 a11=-2; a12=0; a21=0; a22=-3;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=2; x_2_initial=2;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13 %% System simulation
14 t_sim=200;
15 sim('system_dynamics.slx'); clc;
16 t=system_result.time; t_scale='1e3';
17 x=system_result.data;
18 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
19
20 %% Plotting
21 figure(1);
22 subplot(2,1,1)
23 plot(t_scale*x_1,t_scale*x_2,'or'); hold on; end;
```

```
Warning: Input port 1 of
'system_dynamics/Dynamical
system' is not connected.
> In system_simulation (line 16)
Warning: Input port 3 of
'system_dynamics/Mux' is
not connected.
> In system_simulation (line 16)

fx
```

Then I will show you.

(Refer Slide Time: 55:21)



So, this is a vector field and you can see if you choose any vector.

(Refer Slide Time: 55:29)

```
1 - clear; close all; clear all;
2
3 %% System parameters
4 - a11=-2; a12=0; a21=0; a22=-3;
5 - q=0; b1=q; b2=0;
6 - A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 - x_1_initial=5; x_2_initial=0;
10 - figure(2);
11 - plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 - t_sim=200;
16 - sim('system_dynamics.slx'); clear;
17 - t=system_result.time; t_scale=1e3;
18 - x=system_result.data;
19 - x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 - figure(1);
23 - subplot(2,1,1)
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)  
Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

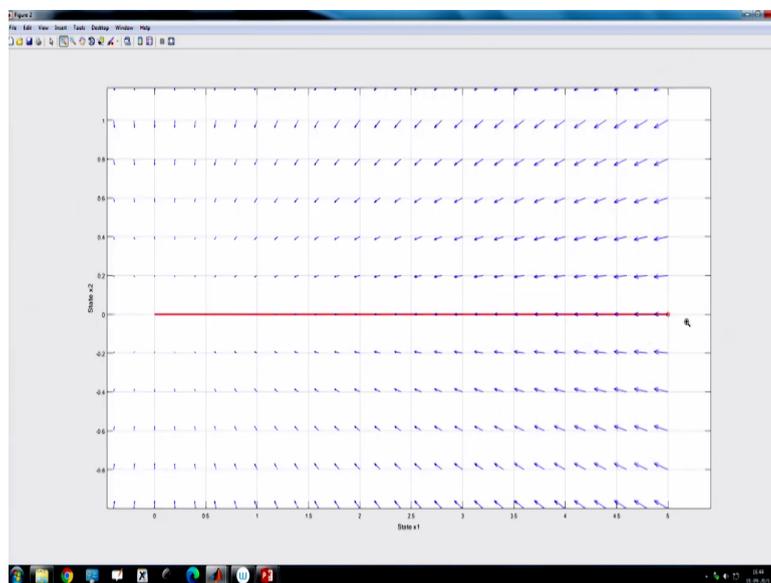
That means if I choose an initial condition, let us say on this axis; that means, I am choosing let us say 5 here 0 here. So, my initial condition will be along the axis.

(Refer Slide Time: 55:42)



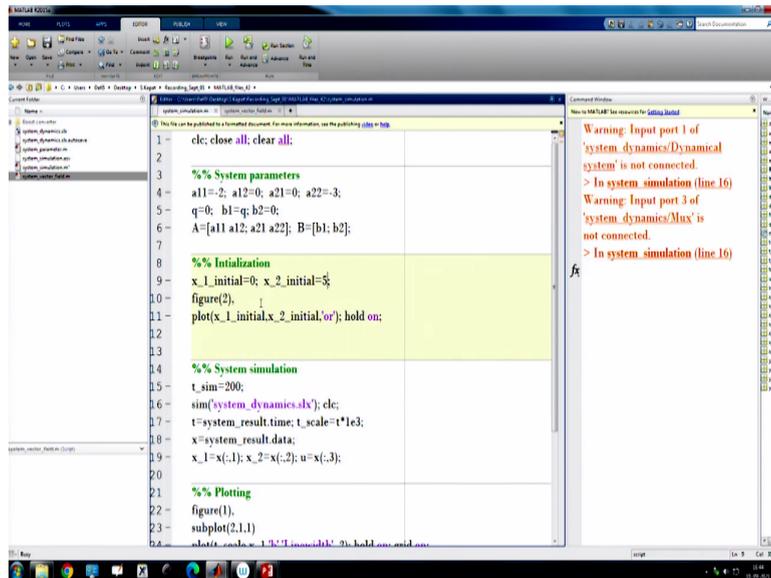
So, you see, it is along the axis. So, it will always go towards this and this is an invariant vector field that I have told.

(Refer Slide Time: 55:48)



It will move along this path.

(Refer Slide Time: 55:51)

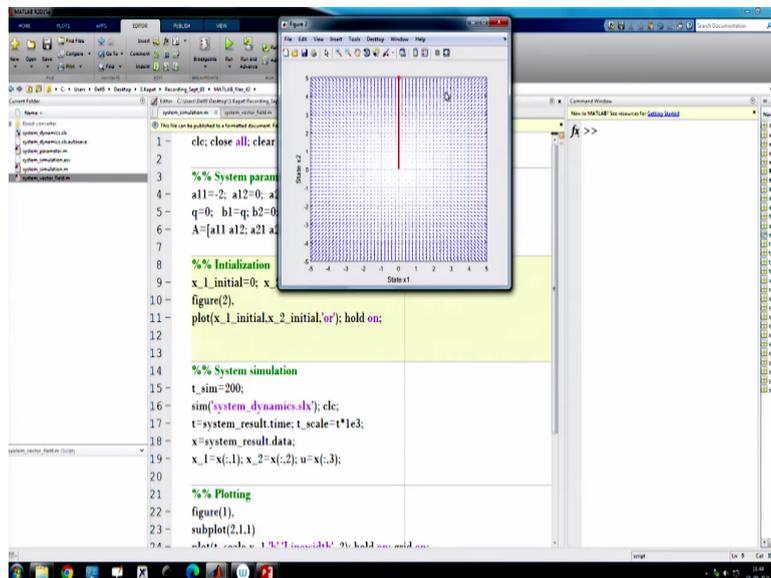


```
1 = clc; close all; clear all;
2
3 %% System parameters
4 a11=-2; a12=0; a21=0; a22=-3;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=0; x_2_initial=5;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13 %% System simulation
14 t_sim=200;
15 sim('system_dynamics.sls'); clc;
16 t=system_result.time; t_scale='t*1e3';
17 x=system_result.data;
18 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
19
20 %% Plotting
21 figure(1);
22 subplot(2,1,1)
23 plot(t,x_1,'b'); hold on;
24
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)  
Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

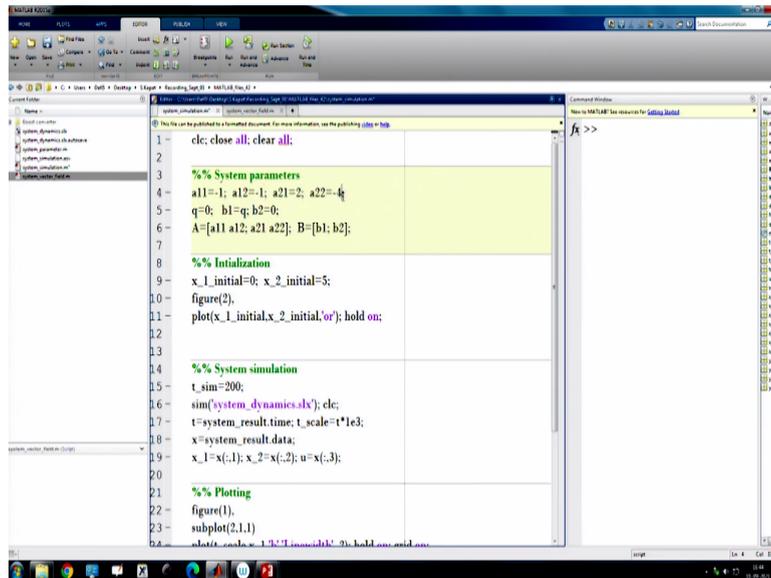
Similarly, if I choose the initial condition 0 and 5. So, you will see it will move along the vertical axis.

(Refer Slide Time: 55:59)



Yes, ok now what will happen if we consider this minus 1 minus 1 2 minus 4?

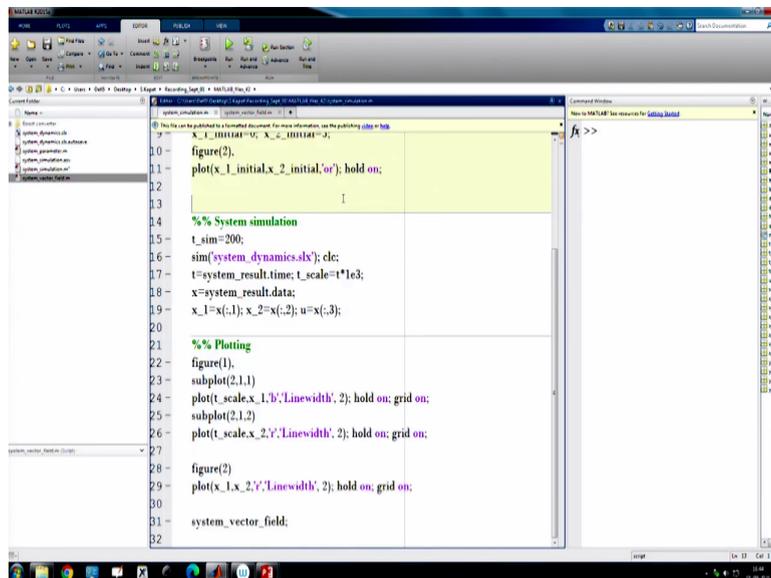
(Refer Slide Time: 56:09)



```
1 clc; close all; clear all;
2
3 %% System parameters
4 a11=-1; a12=-1; a21=2; a22=-4;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=0; x_2_initial=5;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=200;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale=1e3;
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1)
24 plot(t_scale,x_1, 'r','LineWidth', 2); hold on; grid on;
```

It is minus 2 minus 3 we have already considered. So, let us say minus 1 minus 1 2 minus 4. So, minus 1 minus 1 2 minus 4 let me cross check.

(Refer Slide Time: 56:25)



```
9 x_1_initial=0; x_2_initial=5;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13
14 %% System simulation
15 t_sim=200;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale=1e3;
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1)
24 plot(t_scale,x_1, 'r','LineWidth', 2); hold on; grid on;
25 subplot(2,1,2)
26 plot(t_scale,x_2, 'y','LineWidth', 2); hold on; grid on;
27
28 figure(2)
29 plot(x_1,x_2, 'y','LineWidth', 2); hold on; grid on;
30
31 system_vector_field;
```

Let me cross check minus 1 minus 2 minus 4 ok fine. Now we want to run it and see.

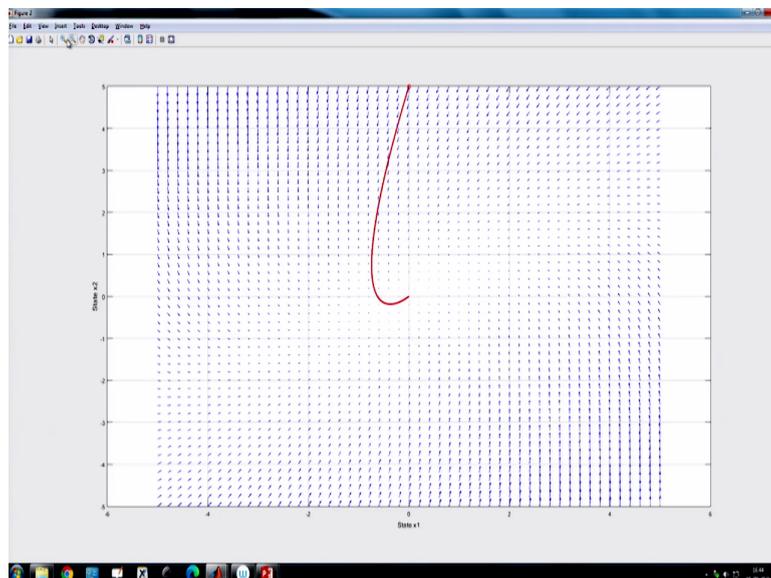
(Refer Slide Time: 56:33)

```
system_simulation.m | system_vector_field.m | Command Window
1 x_1_0111141='x'; x_2_0111141='-y';
2 figure(2);
3 plot(x_1_initial,x_2_initial,'o'); hold on;
4
5
6 %% System simulation
7 t_sim=200;
8 sim('system_dynamics.slx'); clc;
9 t=system_result.time; t_scale=1*1e3;
10 x=system_result.data;
11 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
12
13
14 %% Plotting
15 figure(1);
16 subplot(2,1,1)
17 plot(t_scale,x_1,'b','LineWidth', 2); hold on; grid on;
18 subplot(2,1,2)
19 plot(t_scale,x_2,'r','LineWidth', 2); hold on; grid on;
20
21 figure(2)
22 plot(x_1,x_2,'v','LineWidth', 2); hold on; grid on;
23
24 system_vector_field;
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)  
Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

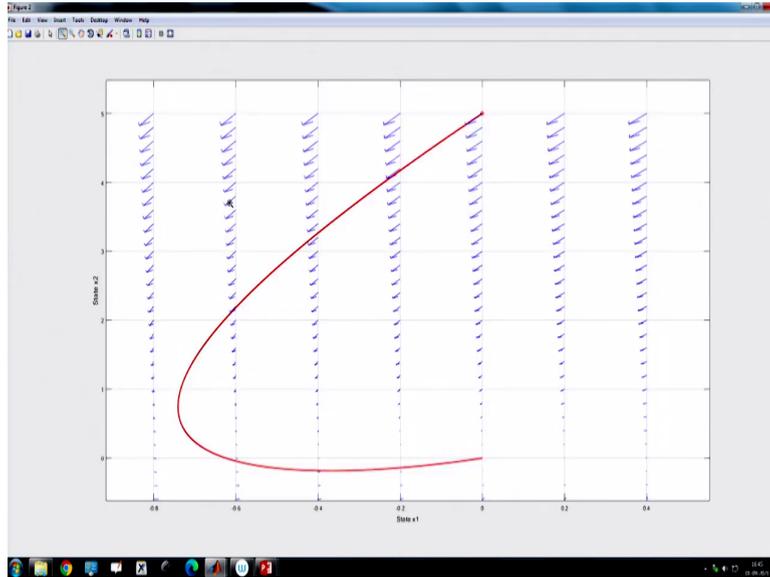
Now, it will no longer.

(Refer Slide Time: 56:37)



Even you take vertically because now the direction field has changed ok. So, we cannot expect that it will come because you can see earlier the basis vector was different now the basis vector is no longer here right so that means, our eigenvector has changed ok. So, the trajectory movement got changed because we have scaled down.

(Refer Slide Time: 56:59)



It is no longer earlier vertical axis was my you know invariant vector field, but it is not the case now right.

(Refer Slide Time: 57:09)

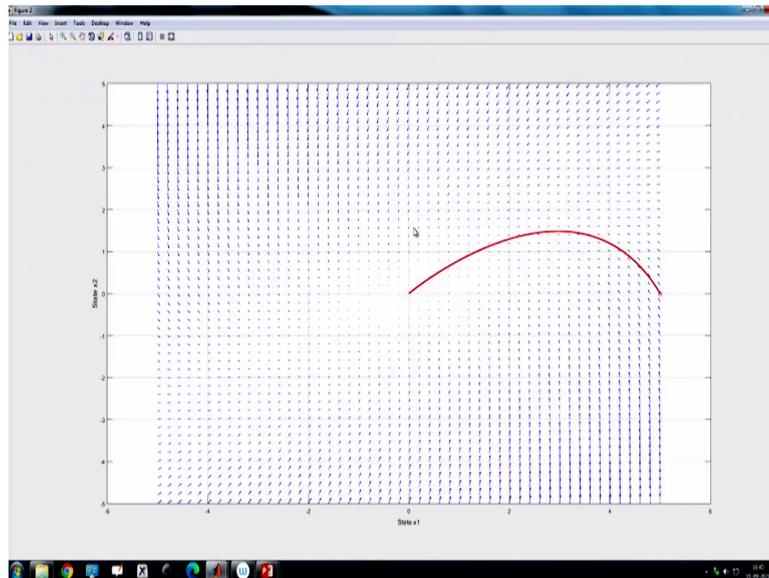
```
1 = clc; close all; clear all;
2
3 %% System parameters
4 a11=-1; a12=-1; a21=2; a22=-4;
5 q=0; b1=q; b2=0;
6 A=[a11 a12; a21 a22]; B=[b1; b2];
7
8 %% Initialization
9 x_1_initial=5; x_2_initial=0;
10 figure(2);
11 plot(x_1_initial,x_2_initial,'or'); hold on;
12
13 %% System simulation
14
15 t_sim=200;
16 sim('system_dynamics.slx'); clc;
17 t=system_result.time; t_scale='1e3';
18 x=system_result.data;
19 x_1=x(:,1); x_2=x(:,2); u=x(:,3);
20
21 %% Plotting
22 figure(1);
23 subplot(2,1,1)
24 plot(x_1,x_2,'or'); hold on; hold on; hold on;
```

Warning: Input port 1 of 'system\_dynamics/Dynamical system' is not connected. > In system\_simulation (line 16)

Warning: Input port 3 of 'system\_dynamics/Mux' is not connected. > In system\_simulation (line 16)

Similarly, if I choose, you know the same as along this 5 and 0. So, it will not run along this path ok it will not run along this path because that direction has changed.

(Refer Slide Time: 57:20)



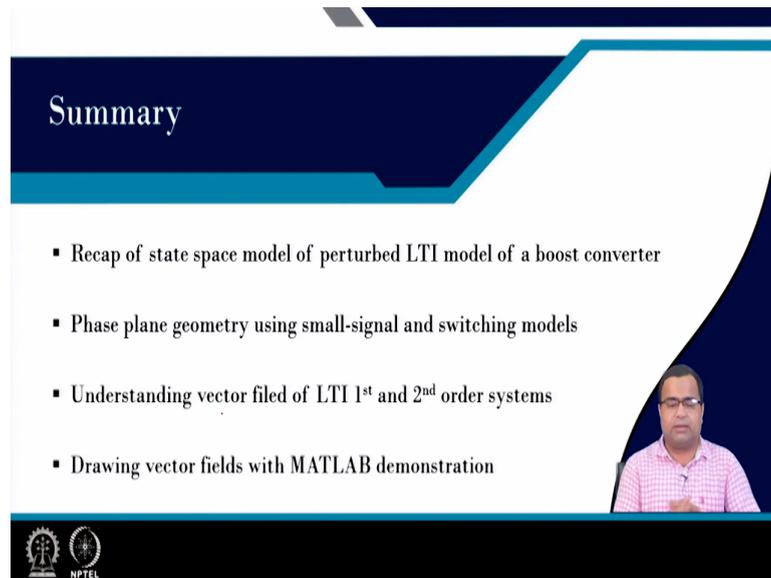
But, nevertheless we can always draw vector field and which will show the direction field right.

(Refer Slide Time: 57:26)



So, we can do ok so; that means the coupled system. So, we need to I know understand the eigenvalues and eigenvector and that we are going to discuss in the next lecture.

(Refer Slide Time: 57:36)



The slide features a dark blue header with the word "Summary" in white. Below the header, there is a list of four bullet points in black text. To the right of the text, there is a small video inset showing a man in a pink shirt and glasses. At the bottom left of the slide, there are two logos: one for NPTEL and another for a university.

## Summary

- Recap of state space model of perturbed LTI model of a boost converter
- Phase plane geometry using small-signal and switching models
- Understanding vector field of LTI 1<sup>st</sup> and 2<sup>nd</sup> order systems
- Drawing vector fields with MATLAB demonstration

So, in summary, we have a recapitulated state space model and perturbed LTI model of a boost converter. We have shown the phase plane geometry using small-signal model, but switched model we have not considered here that we are going to consider next. Understanding vector field for 1st and 2nd order system, then drawing vector field with MATLAB demonstration. So, with this I want to finish it here.

Thank you very much.