Linear algebra

Basics of vectors:
Every pair of numbers gives only one vector but one vector is associated with only one pair of numbers in two dimensions..
We have the general vector icap in x direction and jcap in y direction. Together, they are called the basis of the coordinate system.
Every triplet of numbers gives one vector and one vector is associated with 3 numbers in three dimensions.

Scaling of vectors:
We use scaling in vectors which means stretching or squishing or reversing the direction of a vector. The factor used for scaling is called scalar.

Linear combination:
Scaling two vectors and adding them is called the linear combination of vectors.
It's called linear as if we fix the origin and move the tip of the vectors, it moves along a line.
The set of all possible vectors that we can reach with the linear combination of a given pair of vectors is called the span of vectors. The span of vectors in 2 dimensions forms a type of flat sheet as the two vectors move along the sheet.
A linear combination of three vectors is described in the same way as that of two.
We scale the three vectors and then add them together.
In 3 dimensions, if the third vector moves along with the two vectors in the same plane, then again the span of the three vectors forms a same plane sheet.
But if the third vector is along a different direction then it moves the plane sheet throughout the space.
When the vectors multiplied by their scalars can be added, the vectors are linearly independent.

Linear transformation:
Linear transformation is basically the change of the direction of the grid lines keeping the origin constant.
The transformation depends on where the basis vectors land.
The transformation begins with the linear combination of the unit vectors along x and y directions and ends up with the same linear combination of where those to vectors landed.
The direction of vector rotation is counterclockwise if θ is positive (e.g. 90°), and clockwise if θ is negative (e.g. −90°)
If a standard right-handed Cartesian coordinate system is used, with the x-axis to the right and the y-axis up, the rotation of the grid lines is counterclockwise. If a left-handed Cartesian coordinate system is used, with x directed to the right but y directed down, rotation is clockwise.

Matrix multiplication as composition:
Composition is the combination of rotation and then applying shear through the icap and jcap vectors.

We multiply the shear matrix to the rotation matrix for the composition.
Matrix multiplication is associative.

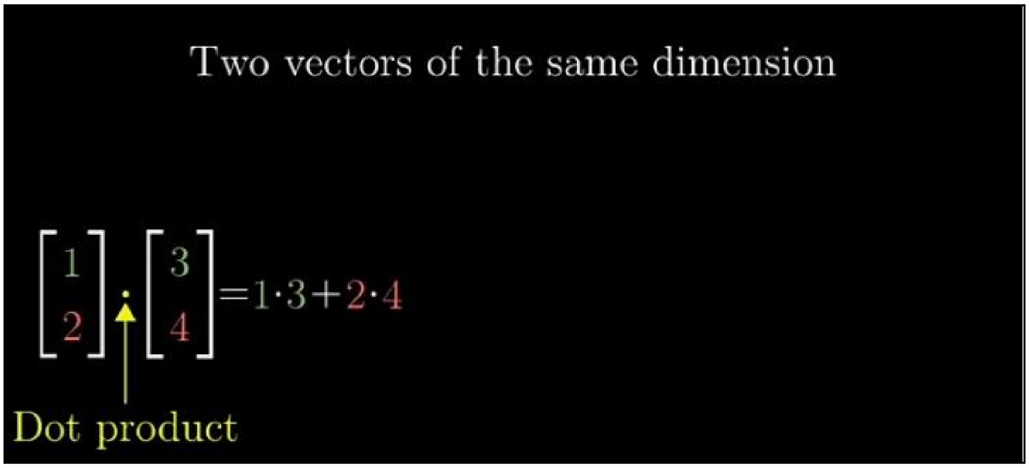
Determinant of a vector matrix.
To understand the scaling of a transformation, we have to understand the factor with which it stretches or squishes, and that factor is understood by the difference in the area caused due to the transformation in a two dimensional matrix.
The actual factor of the change in area is determined by the determinant value of the transformation.
If the determinant comes out to be negative, it means the orientation of the plane has been inverted.
In 3 dimensions, we get the scaling of the volume of the transformation.
As in 2 dimensions we are focussing on the 1×1 area scaling, in 3 dimensions we focus on the 1×1×1 cube scaling whose edges are on the unit vectors icap, jcap and kcap.
Even in 3 dimensions, if the determinant is negative, it means that the direction of the unit vectors has been inverted and can be checked using the thumb rule.

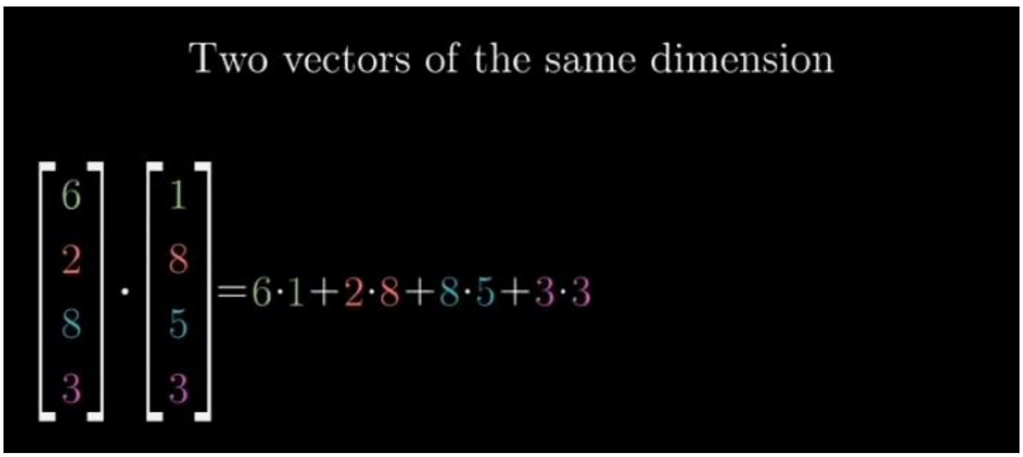
Matrix multiplication:
If we write a x →=v→ then it means that we are finding a vector x along the direction of the vector v after applying transformation.
If the determinant of matrix a is non zero then we can say that there will be only one vector that will land on vector v.
If we apply an inverse transformation, we get an inverse of the matrix a as a-1.
So if a is the rightward shear then a -1 would be a leftward shear and vice versa.
The multiplication of both these will give a transformation that does nothing i.e. an identity transformation.

Rank of a matrix:
Rank means the number of dimensions in the output of the transformation.
Or
Number of dimensions in the column space.
If the rank equals the number of columns in the matrix then it is called as full rank.
The columns of the matrix tell us where the basic vectors land and the span of those transformed basis vectors gives us all the possible outputs.
Column space is the span of the columns of the matrix.
Null space is the space where all the vectors become null in the sense that all vectors land on the zero vector.
Column space helps us understand whether a solution even exists but a null space helps us understand how the set of possible solutions look like.

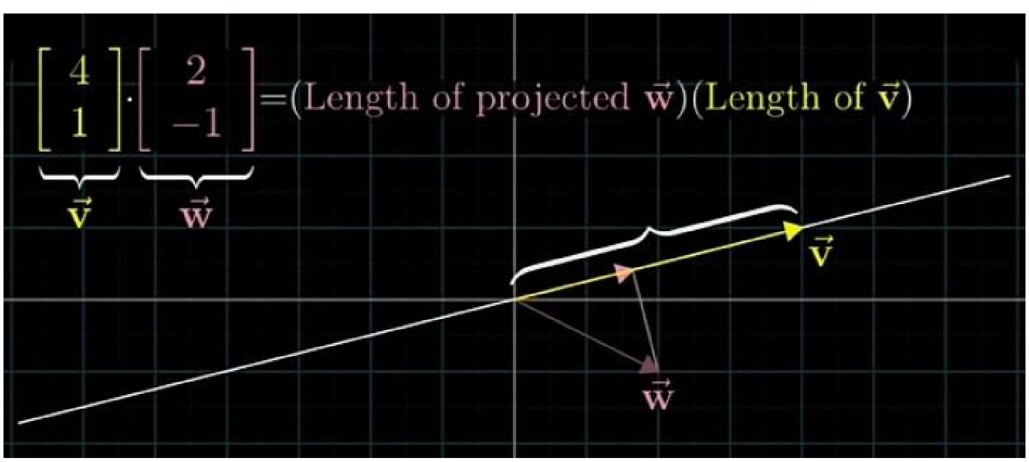
Non square matrices are simply those matrices whose number of rows and columns is not equal.

Dot product of matrices:

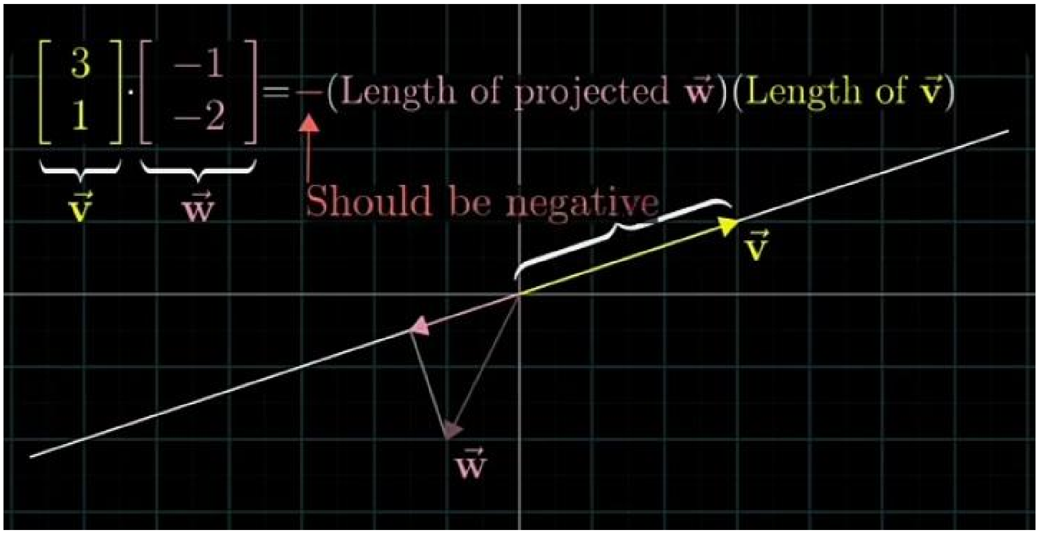




Meaning:
Imagine projecting vector w onto a line that passes through the origin and the tip of the vector v. Multiplying the length of projection to the length of v we have the dot product v•w.



If the direction of w is opposite to v, we have the dot product negative.

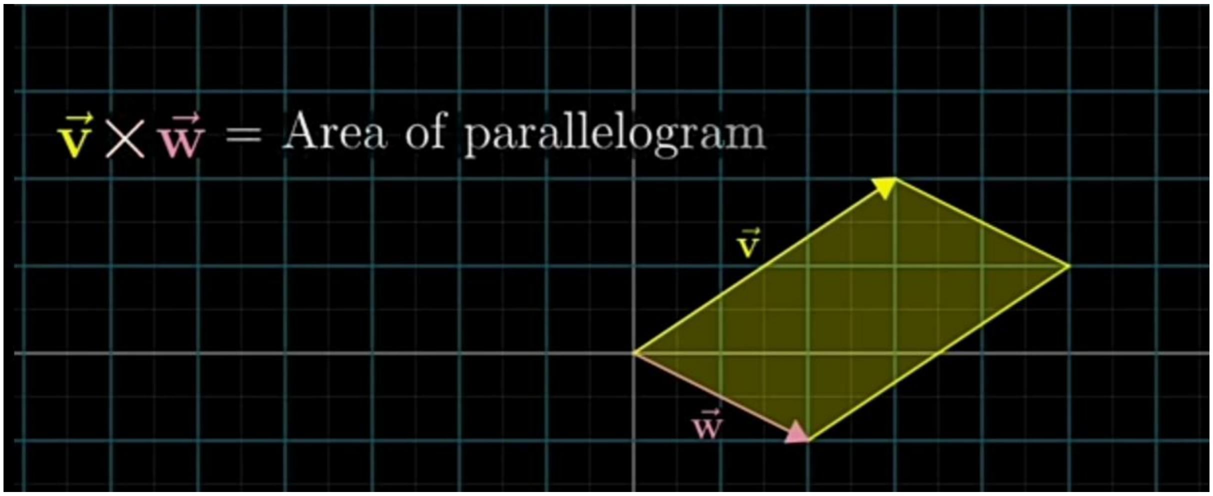


So,
v•w >0 if vectors are in the same direction.
v•w = 0 if vectors are perpendicular to each other.
v•w <0 if vectors are in the opposite direction.

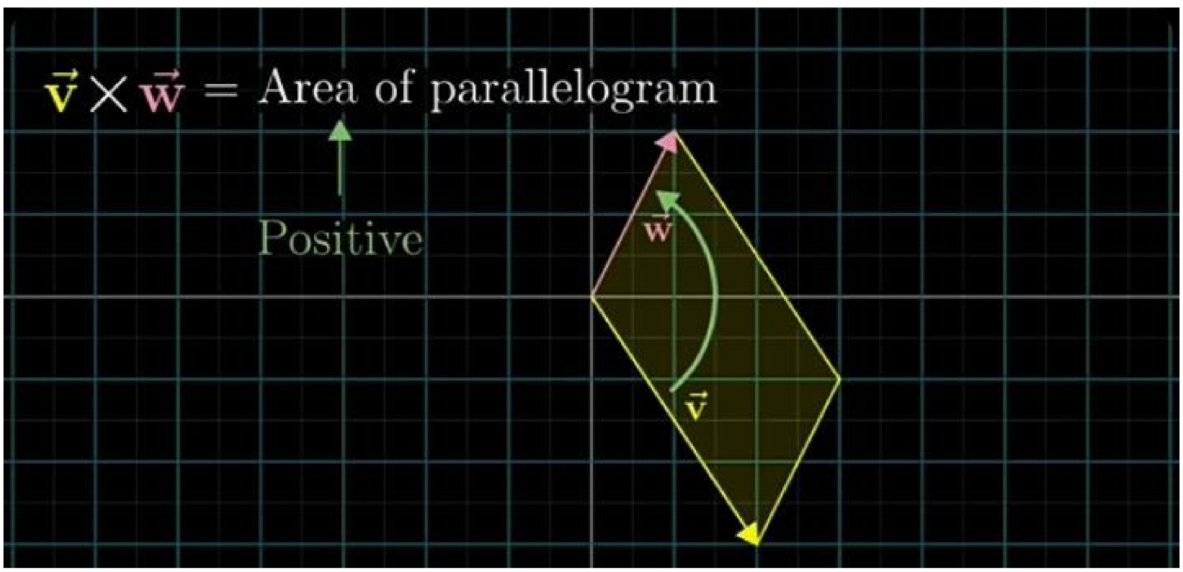
Dot product can also be interpreted as projecting a vector onto the span of the unit vector and taking the length.
Dot product with a non-unit vector can be interpreted as first projecting onto that vector then scaling up that length of projection by the length of the vector.

Duality
When we have a linear transformation from some space to a number line, it is associated with a unique vector in that space in the sense that performing the linear transformation is the same as taking a dot product of that vector.
Numerically this is possible because its is described by a matrix with just one row where each column tells us on which number each of the basis vector lands on and multiplying this matrix with vector v is computationally equal to taking the dot product between v and the vector we get by turning the matrix on its side. So, whenever there's a linear transformation to a number line, we will be able to match it to some vector known as the dual vector of that transformation.

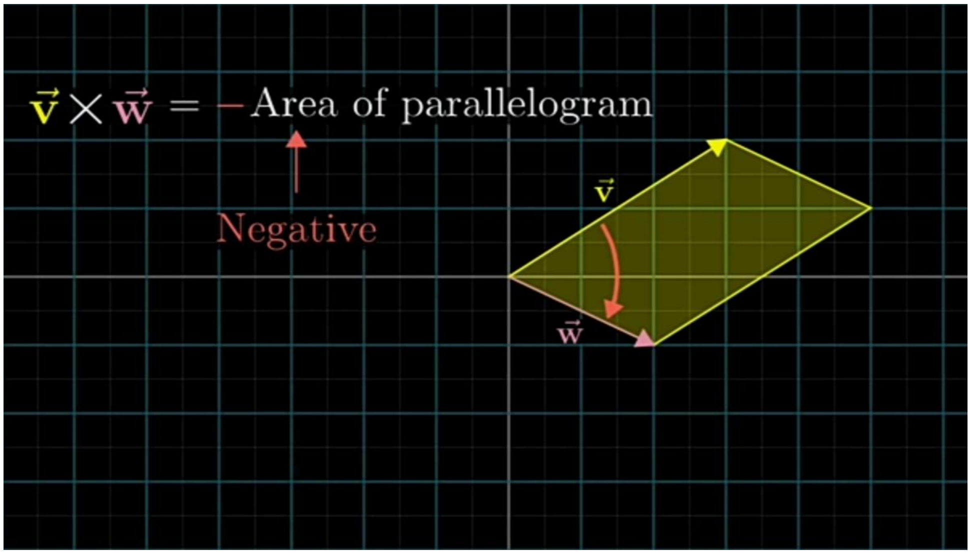
Cross product:
Cross product will be the area of the parallelogram formed between the two vectors.



The cross product will be +ve if the vector v is to the right of w

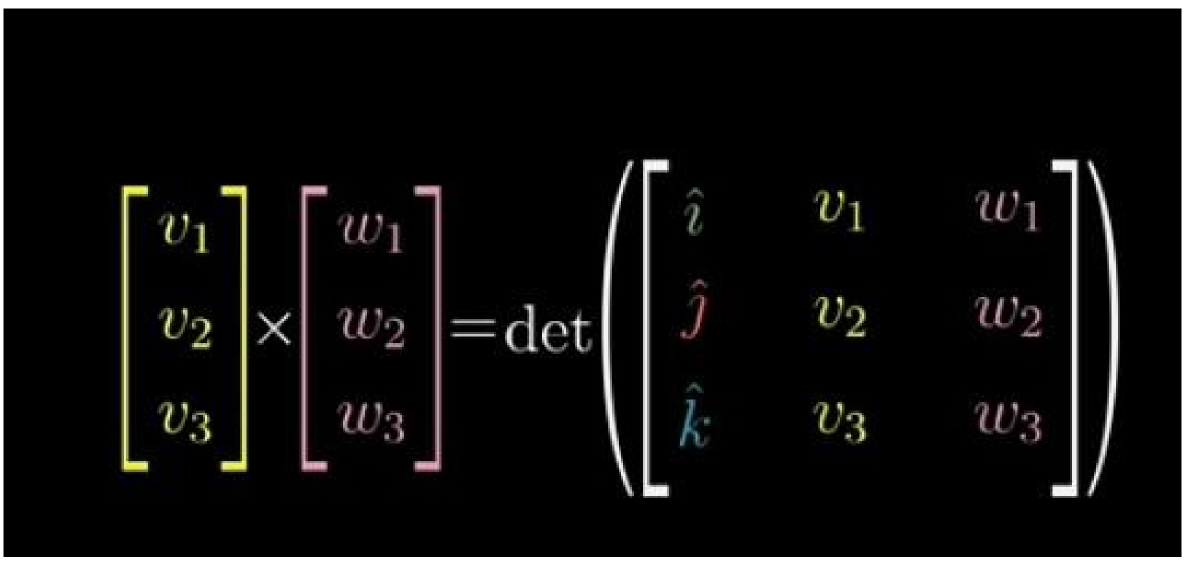


The cross product will be -ve if the vector v is to the left of w



Actual meaning of cross product is producing a third vector which is a combination of 2 vectors.
Cross product is a vector and not a number.
Direction of the cross product is found out using the right hand thumb rule.

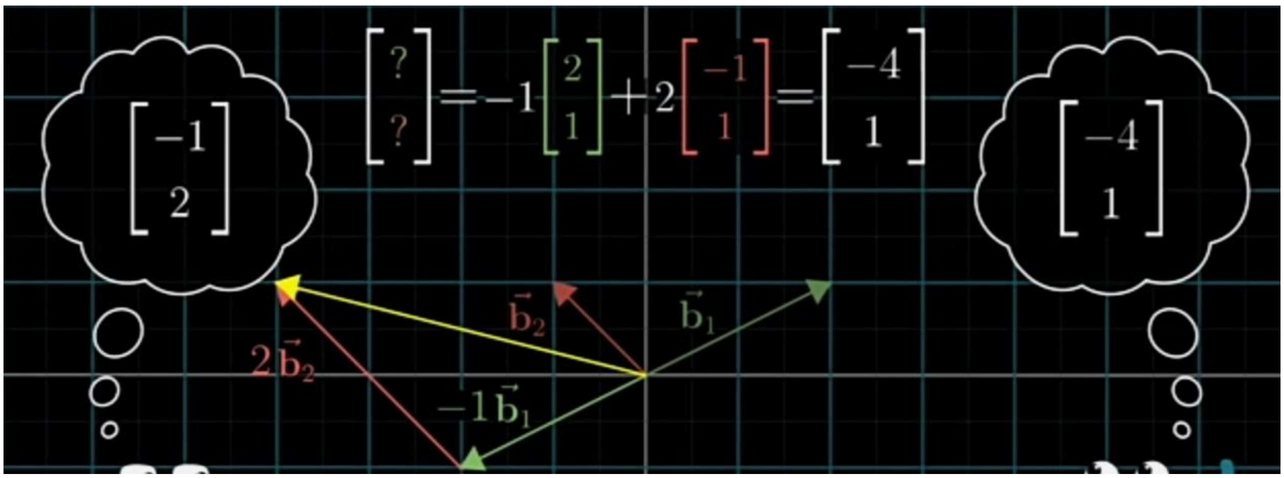
Computing cross product using the matrix of v and w vectors:



This determinant gives a vector in the third direction as a linear combination of the two vectors v and w.

Change of basis:
Translating between vectors and numbers is called the coordinate system and the two important vectors icap and jcap are the basis vectors of a standard coordinate system.

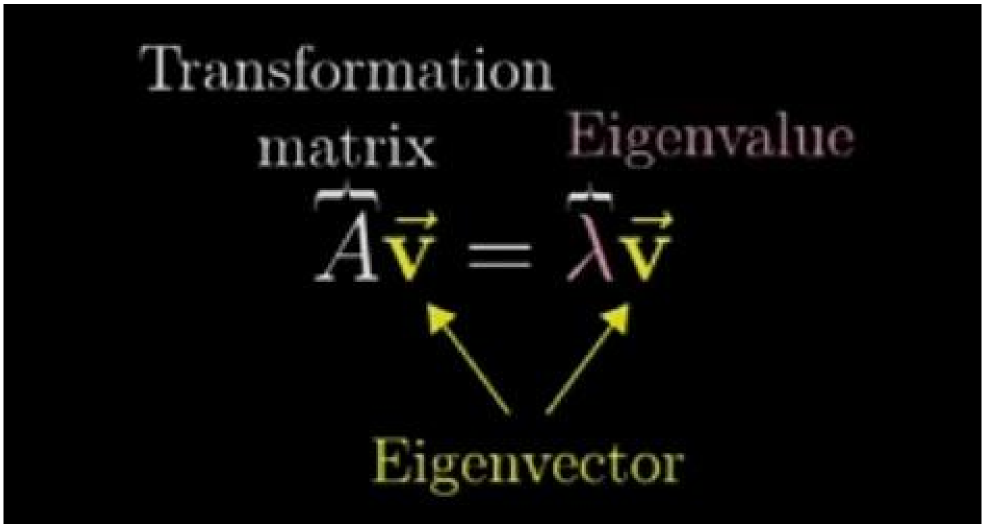
How to translate between coordinate systems?
If we make a different coordinate system, say, where the direction of the grid lines is different i.e. they are inclined at some angle and then the space between the grid lines may be different from the standard coordinate system, then we can translate the coordinates from that system to the standard or vice versa.
Suppose the following example where in some coordinate system, we have the coordinates as [-1 2] then the vectors would be -b1 and 2 b2 then in our coordinate system, we have b1 as [2 1] and b2 as [-1 1] then we can write -b1+2b2 and we will get the matrix as [-4 1].
This is how we can translate between coordinate systems.

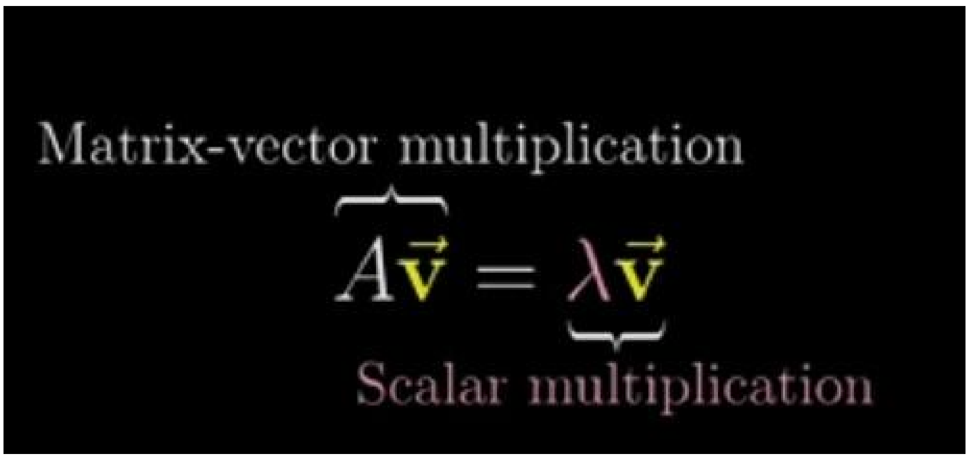


This is actually similar to vector matrix multiplication.
So basically we are applying matrix transformation here.
Before transformation, we are considering a linear combination of our basis vectors and the key feature is that the resultant matrix would be the same linear combination with new basis vectors.

How to translate a matrix?
First, we take the matrix from another language and multiply it with the matrix in our language found out using the change of basis method. This gives us the same vector but in our language. Then, we multiply the transformation matrix to the left. So, this gives the transformed vector in our language. Then, multiply the inverse of the basis matrix to the left. So, now we get the transformed vector in the other language.

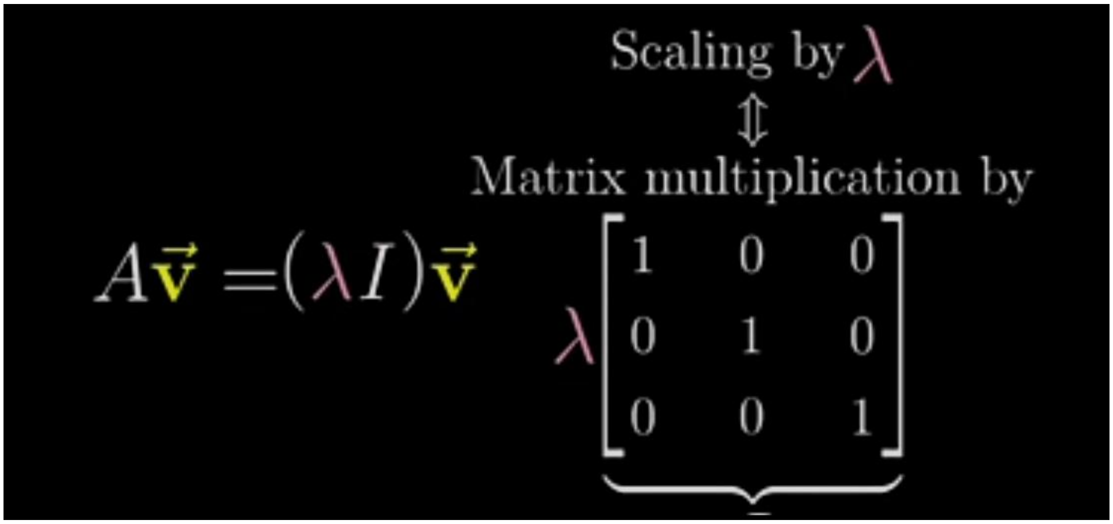
Eigen vectors
During a linear transformation, some vectors in the span get stretched or squished by some factor, these vectors are called the eigenvectors.
The factor by which the vector has been stretched is called the eigenvalue.
In 3D, the plane rotates, but in this case the eigenvalue is 1 as rotation does not cause scaling of vectors.

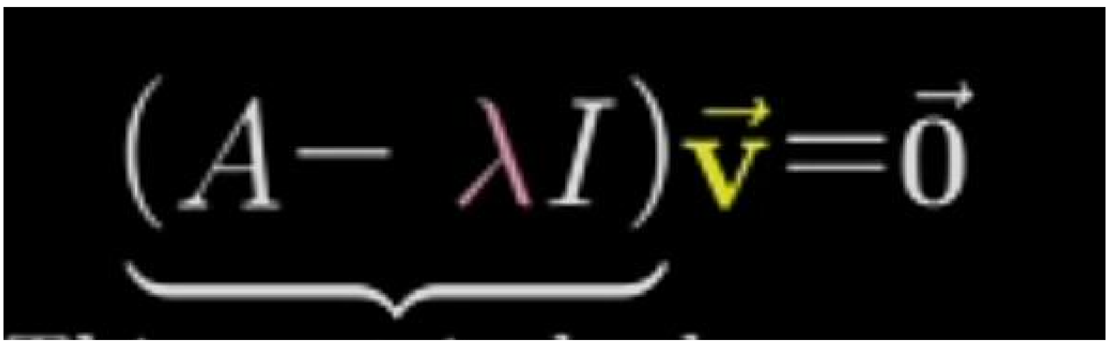




The above expression shows that the matrix vector product is equal to the result that we get by just scaling the eigenvector by some scalar value lambda. In this equation, lhs includes matrix vector multiplication while rhs includes scalar matrix multiplication.
So we rewrite the rhs as matrix vector multiplication using a matrix with has the effect of scaling a vector by a factor of lambda.
The columns of this matrix denote the basis vectors and each basis vector is multiplied by a factor of lambda. So this matrix contains zero everywhere except in the diagonal where the value is lambda.

We can write the equation in this way:





For the product of the matrix with a non-zero vector to be zero if the transformation associated with that matrix squishes space to a lower dimension. So, the determinant of the equation should be zero.
To find if the lambda is a eigen value we should subtract the lambda value from the diagonal values of the determinant and equate it to zero. So for only these values of lambda, it will be an eigen value. Now to find the eigen vector for that particular value of lambda we will substitute the value of lambda in the determinant and then solve for the vectors for which these diagonally altered values gives zero.
Actually a 2D vector doesn't have any eigen vectors.
When we apply shear then the vectors on the x axis are itself the eigenvectors with eigen value 1 as it does not stretch.

Control Bootcamp

Types of control:
Passive control - Passive vibration control does not require an outside source of power, but instead uses the mass of the structure to mitigate vibration. The main examples of passive vibration control systems are tuned mass dampers, viscous dampers.

Active control - Active structural control is an area of research in which the motion of a structure is controlled or modified by means of the action of a control system through some external energy supply.

- Open loop feedback control - Consider a block of system that has some input u and output Y. This basically inverts the dynamics to figure out what is the perfect input u to get the desired output y.
  Now, considering an inverted pendulum, to balance that, we can continuously pump the pendulum up and down at a high enough frequency so it will become stable naturally. So the base is u i.e. a high frequency sinusoidal wave and the output y is maybe the angle and the aim is to essentially keep the pendulum stable. But to achieve this we gotta be pumping it continuously otherwise it may become unstable.
  So we need another control.
- Closed loop feedback control - In this we take sensor measurements of what the system is actually doing and we have controllers and so feed it back to the input signal that can manipulate the system.

  U→ system →y
  ↑ ←control←↓

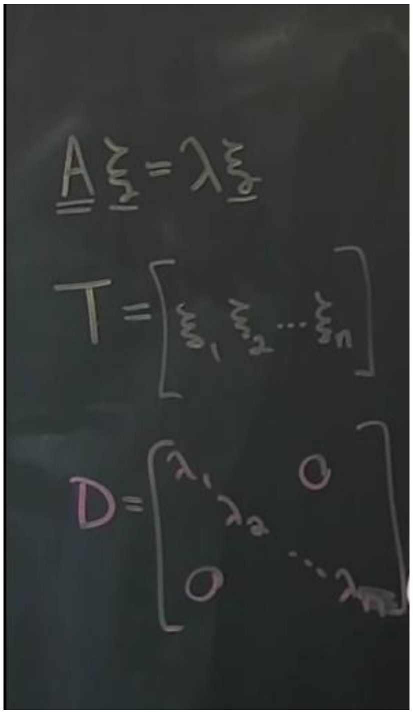
Why is feedback necessary?

- Uncertainty - If there's any disturbance then the system is going to make it such that our pre planned trajectory would be suboptimal.
  But in case of feedback, when there's uncertainty, we can feed it back and control it accordingly.
- Instability - In an open loop, we can never fundamentally change the behavior of the system itself.
- We can reject the disturbances incoming in the system - When there's a disturbance, it'll pass through the system dynamics then will be measured by the sensor and the controller can correct it when the feedback is received.
- Efficient control

We consider a state variable X which is a vector that describes all the quantities required in our system. In case of a pendulum it could be an angle or angular velocity.

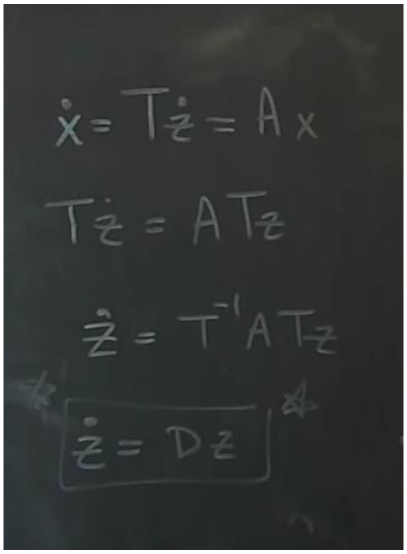
X• =Ax
So we have x(t) which has a solution x(t)= e^ At x(0)
If all eigenvalues of the system are positive then the system is stable.
In control systems, we write X• = Ax +Bu
U is like an actuator or a control knob.
It could be the position of the base or the voltage of the motor.
B tells us how the control knob affects the time rate change of our state.

X• =Ax where A is a matrix and X is a vector and x € R
In x(t)= e^ at x(0),
e^At = I + At + $A^2t^2/2!$ +$A^3t^3/3!$...
Computing this is very difficult.So, we use eigenvalues and eigenvectors to get a transformation from the x coordinates to eigenvector coordinates from which we can find out the expansion and then understand the dynamics as well.

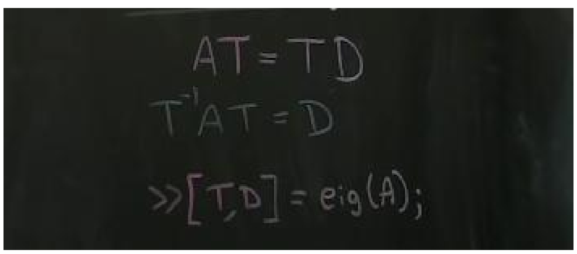


We are scaling the vector lambda times. And from this, we can write
AT=TD
But the Matrix D which has eigenvalues only along the diagonal may not be true at all times because sometimes we may have some generalized vectors.

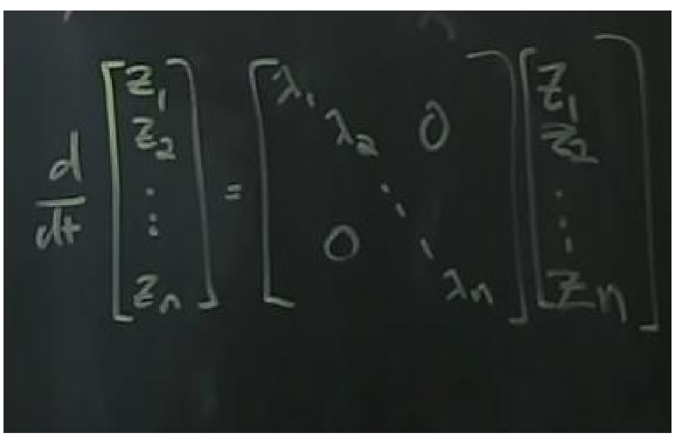
Writing x as Tz:



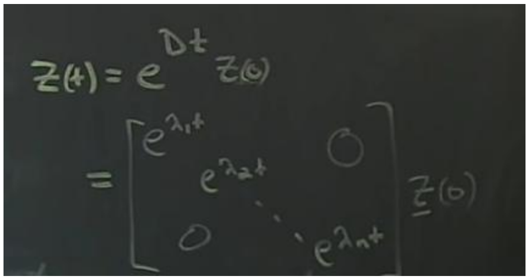
where z is the eigenvectors in the same direction as that of T.



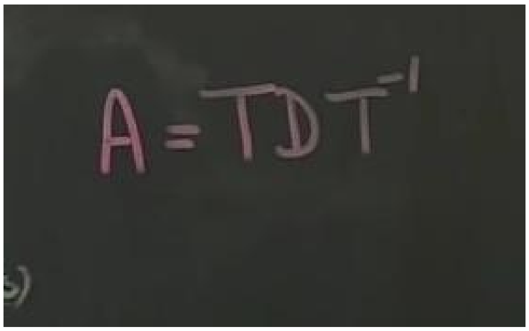
So, z• is always in terms of z.



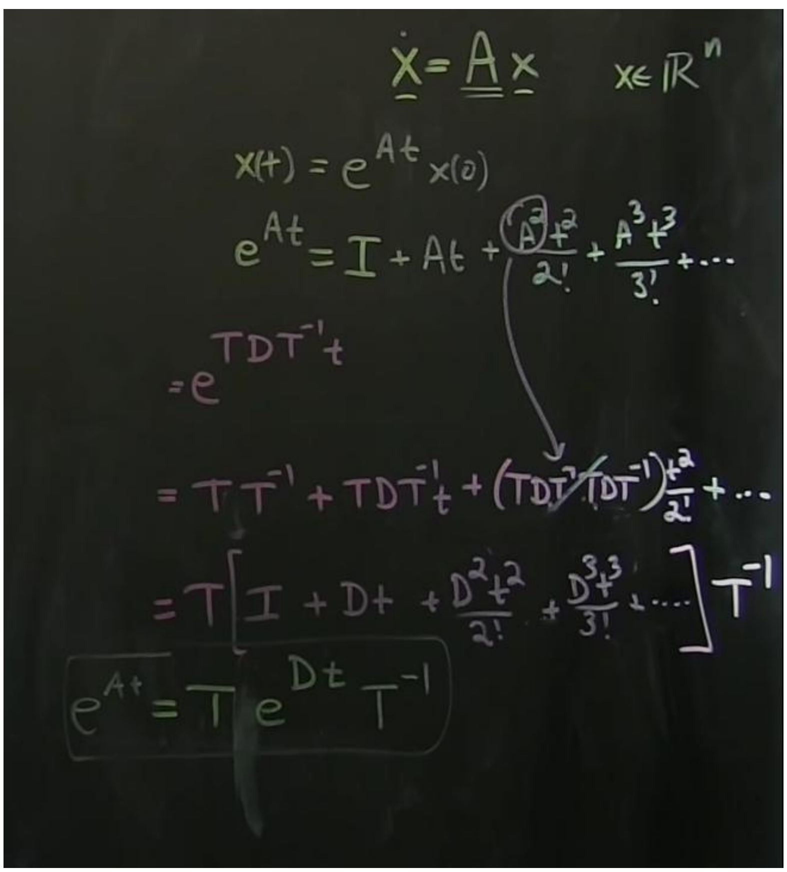
So this is a simpler way for calculating.

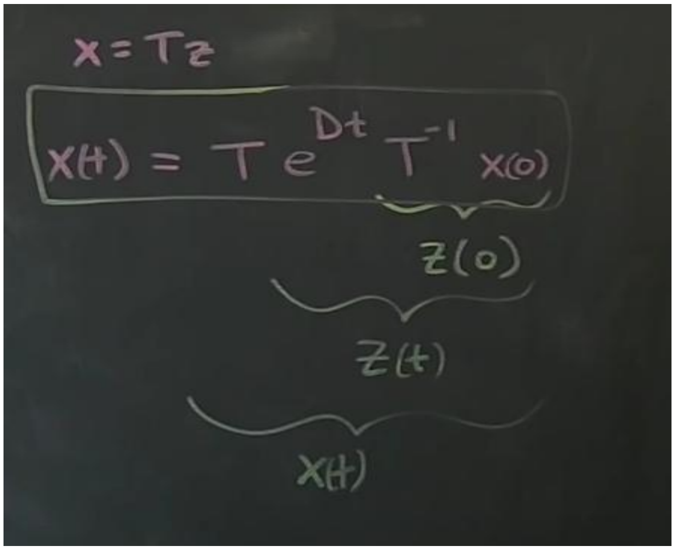


We can also write it like this

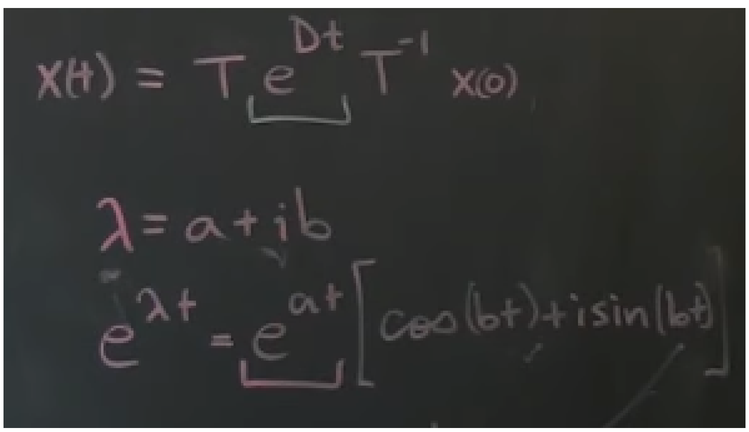


So then:



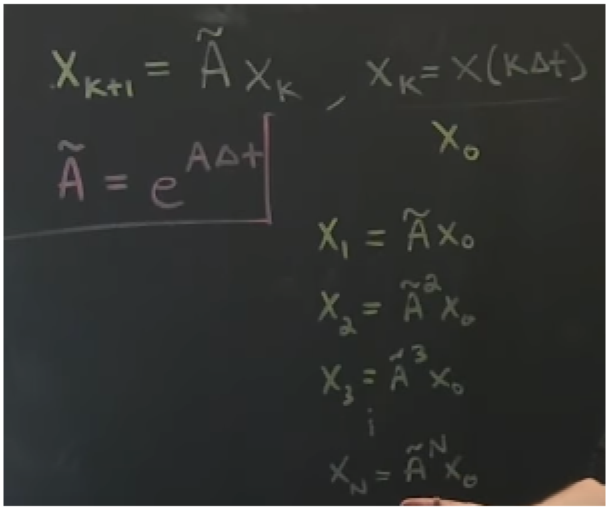


Stability:



In continuous value system,
If a>0, the graph is increasing.
If a<0, the graph is decreasing.
When lambda belongs to complex plane, and the real part a is -ve, then, it will be stable and if real part a is +ve then it is unstable.

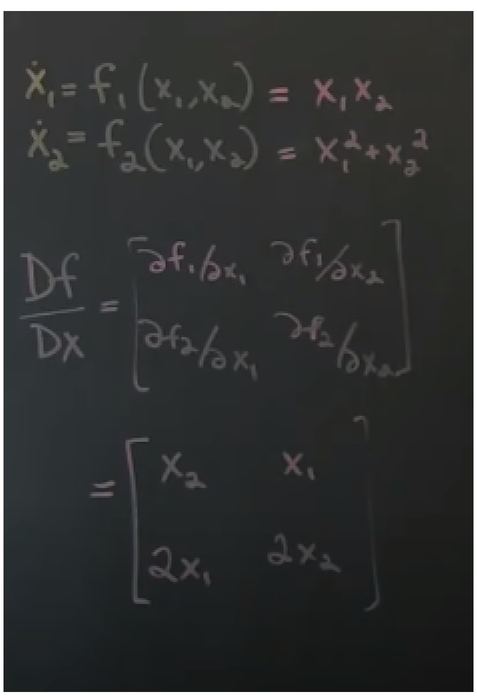
In a physical system, we do not have a continuous system x(t). Instead, we have a dynamical system x such that



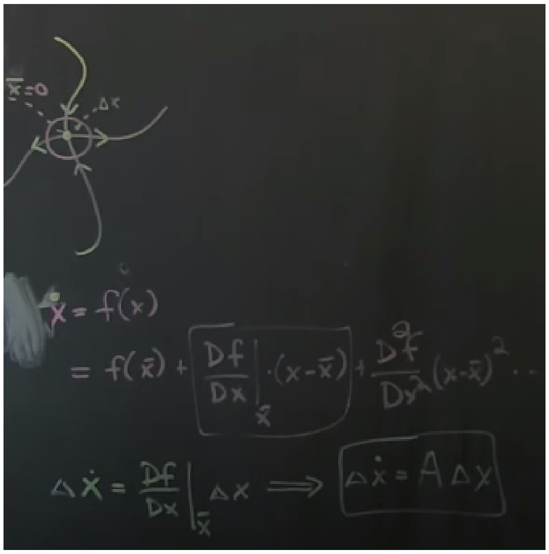
In discrete value system,
Any eigenvalue lambda can be written as a radius and an angle as lambda= Re^itheta
(lambda)^n = R^n e ^ i(ntheta), so in this case, the radius keeps on increasing and the angle keeps rotating.
If we have a unit circle, then the eigenvalues inside the unit circle will be stable and outside it will be unstable because if the real part becomes greater than 1, then the (lambda)^n term goes up to infinity.

To go from a non-linear system to linear system, for a pendulum,

- We need to get some fixed points. Xbar such that f(xbar)=0.
  So in case of a pendulum, the fixed points could be when it is straight up or down.
- Linears about xbar.

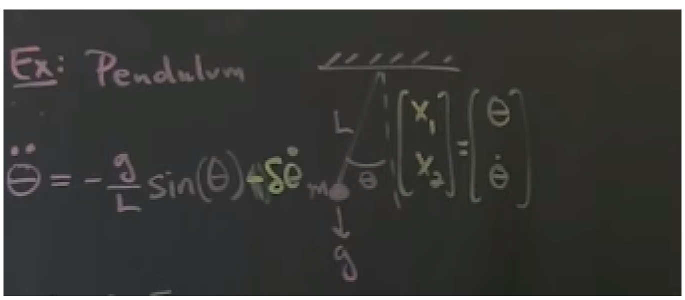
  

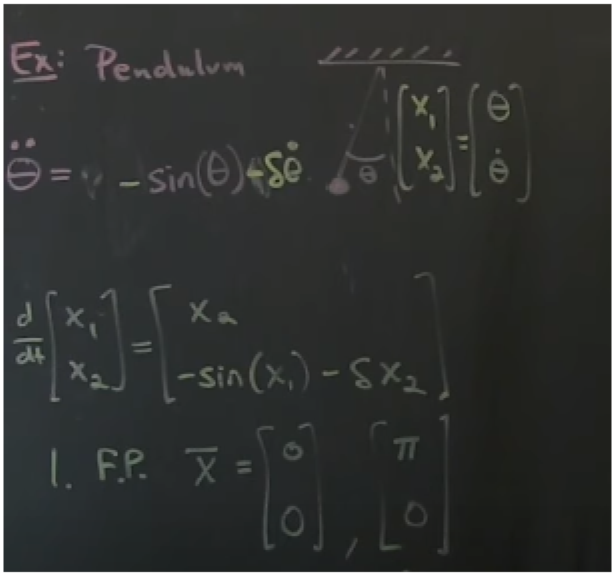
  Now, we have to evaluate this for xbar so we plug in the values of xbar in x1 and x2.



Whenever we are closer to the fixed point(when the pendulum is straight up), then the system is linear. Good and effective system can keep the pendulum stable in a better way.

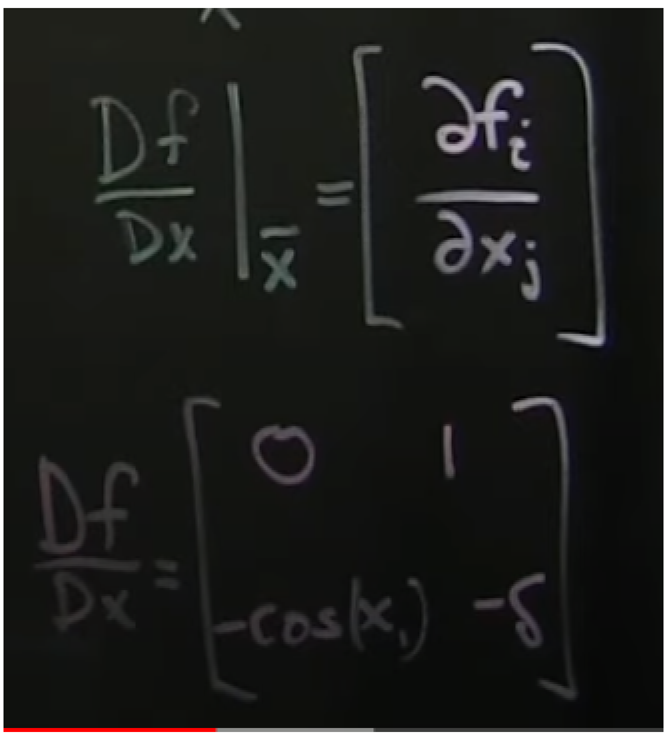
If the eigenvalues of linearization of fixed points are hyperbolic (non-zero real part), then, then the linearization done in that way will work.





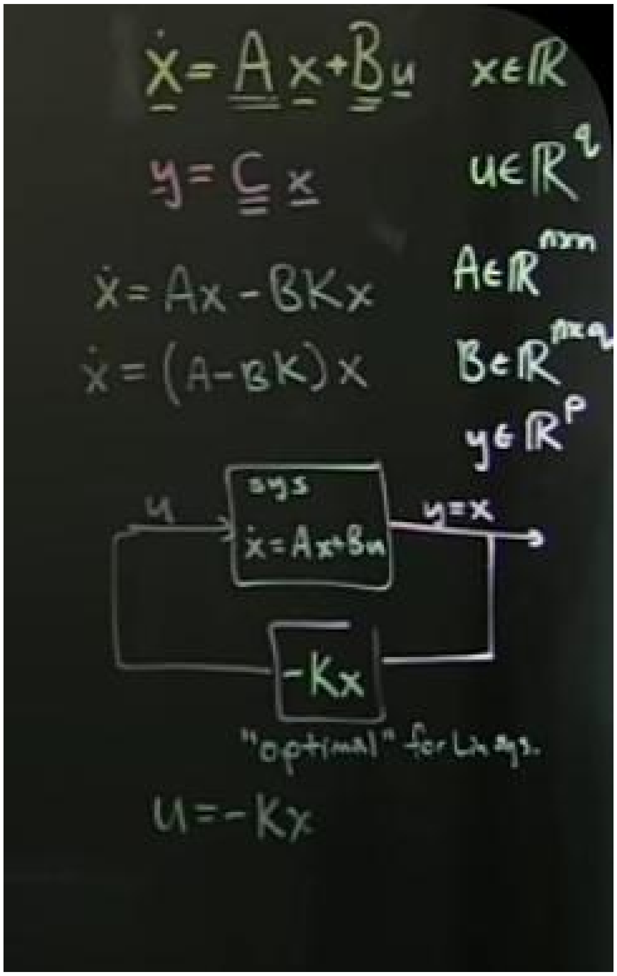
when g/l is 1

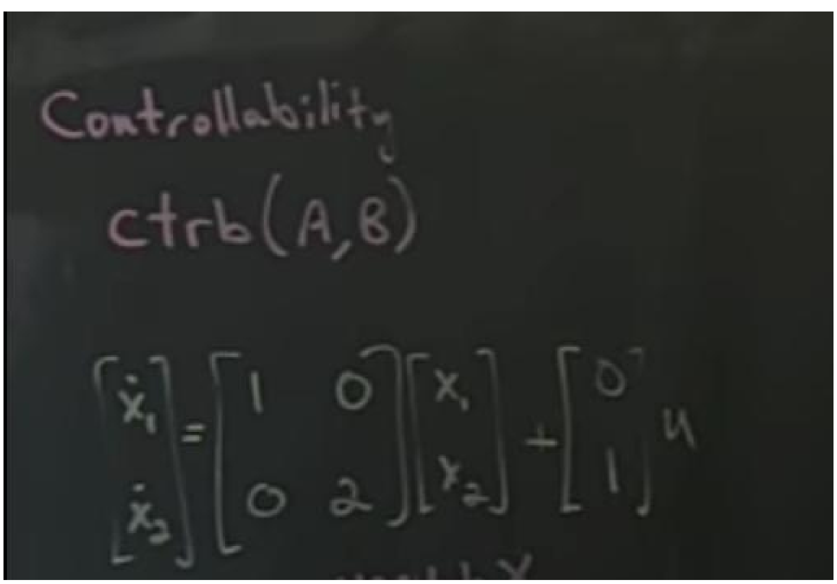
So the pendulum will have fixed pt when it is straight up or straight down so then theta will be zero or pi.



Controllability:

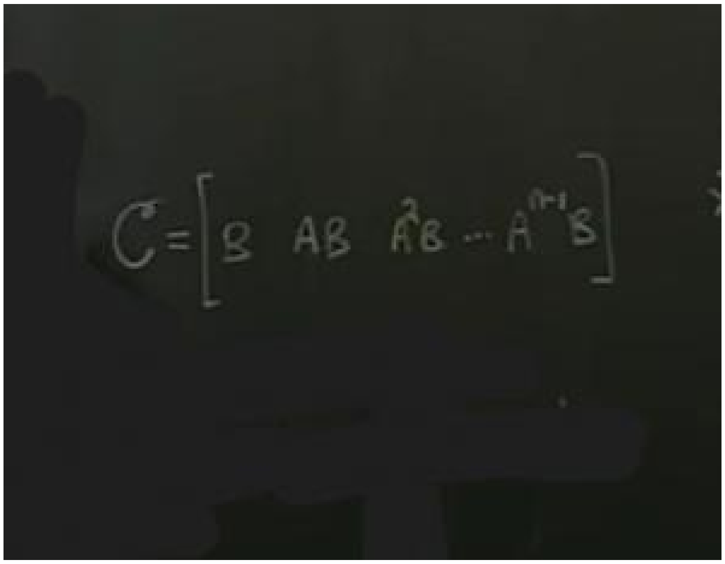
We have a linear system x• = Ax+Bu where x is a state vector and u is set of input signals or control knobs.





This is an uncontrollable system. We have to make it controllable.
One way to make it controllable is by increasing the number of actuators.

Another way is by coupling it with a matrix term like
[1 1
0 2]
Instead of the above one.
A system is controllable if this matrix has n linearly independent columns.

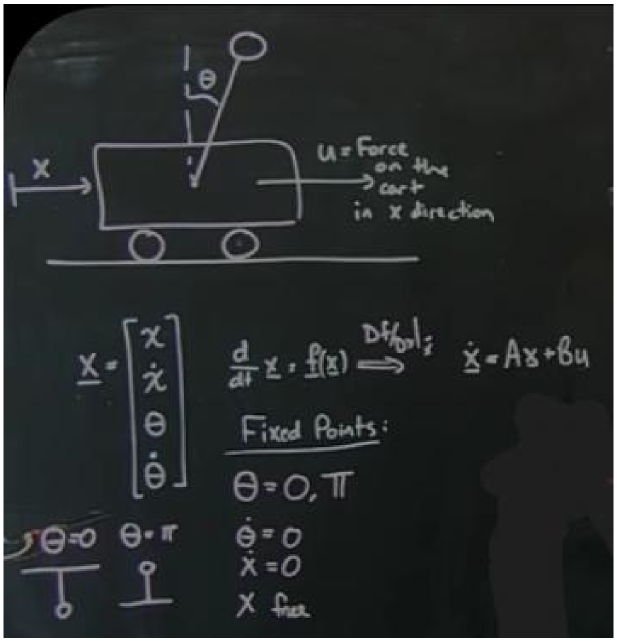


If the rank of the matrix C is n then the system is controllable.

Equivalences to controllability:

1. System is controllable. Then only we can place the poles arbitrarily.
2. Arbitrary eigenvalue or pole placement.
   such that if we have u =-kx then
   X• = (A-Bk)x
3. Reachability - If a system is controllable then for any value of u input can steer the system to any state x.

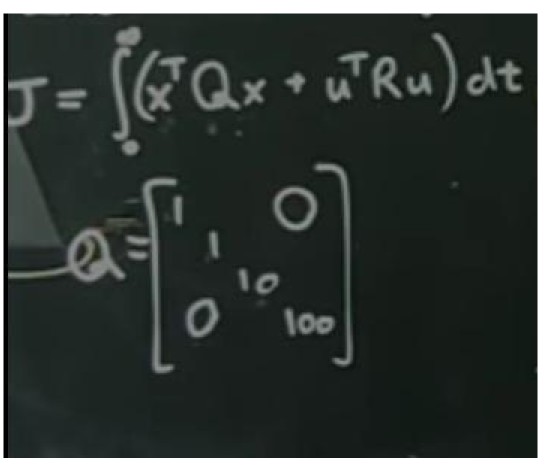
Inverted pendulum on a cart:



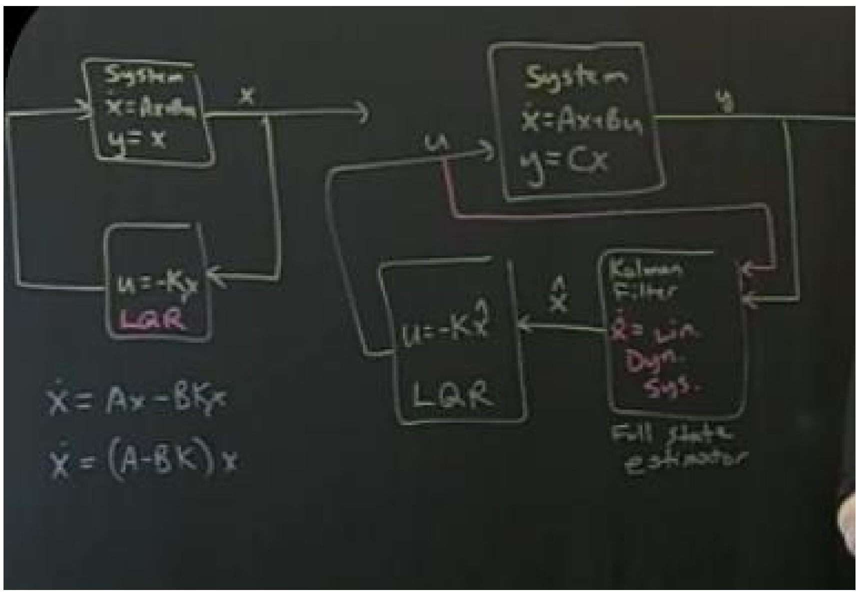
System with 2 degrees of freedom. A

For the pendulum to be straight down the theta has to be zero and for being straight up the theta has to be 90°.
In this, x doesn't matter and so x is a free variable.

When a system is controllable, we can define a full state feedback u=-kx so that we can place the eigenvalues of the closed loop system as desired.
By defining a matrix in matlab we can find the K to move the system to the eigenvalues and then we take the control law of the nonlinear system and show that we can stabilize the unstable upward pendulum system.
The eigenvectors that should be used can be found out using the Linear Quadratic Regulator (LQR).
We will build a cost function which is defined as follows:



where Q is a matrix which tells us about the penalty that could be developed if x is not present as desired. We need to make the matrix Q quite big so that the pendulum is easily stabilized. The R vector has to be small so that we can actuate it aggressively. When we have Q and R then we will have the K matrix (u=-Kx). This is the best control protocol that minimizes the cost function. And this is known as the Linear Quadratic Regulator control.
Linear full state feedback controller that minimizes the quadratic cost function and regulator means controller i.e it will stabilize the system.



Observability- Gives us an estimation of the full state of vector x with lesser measurements.

- A system is observable if the rank of the matrix C (y=Cx) is n.
- Can estimate x from any time series of y.
- We have the observability granion that tells us how much a system is observable.

Model predictive control

Model predictive control (MPC) is an optimal control technique in which the calculated control actions minimize a cost function for a constrained dynamical system over a finite, receding, horizon. At each time step, an MPC controller receives or estimates the current state of the plant.

- It is effective bcos it has some constraints
- Can convert a non linear system to linear.
- Optimisation
- Relies on fast hardware.