

fHandle = @(arg1,...) FunctionDefinition

MATLAB Commands	MATLAB Variables
<pre>>> fHandle = @(x) cos(x); >> x0 = 3; >> xmin = fminsearch(fHandle,x0) xmin = 3.1416 >> fHandle(xmin) ans = -1.0000</pre>	<pre>fHandle @(x)cos(x) x0 3 xmin 3.1416</pre>

1.

2. Solving ordinary diff equation is very diff. so we go with numerical approx solving

We use numerical based approximation for ODE equations

$$\frac{dP}{dt} = rP \left(1 - \frac{P}{K}\right), \quad P(0) = P_0$$

Folder

↓

$$P(t) = \frac{KP_0 e^{rt}}{K + P_0(e^{rt} - 1)}$$

$$\frac{dy}{dx} = e^{-(x^2+y^2)}, \quad y(0) = 1$$

ode45(derivFunction,interval,initialValue)

function_handle tSpan = [tStart,tEnd] y(tStart) = y0

- Function: $y' = f(t,y)$
- Interval of independent variable
- Initial value: $y(t_{start}) = y_0$

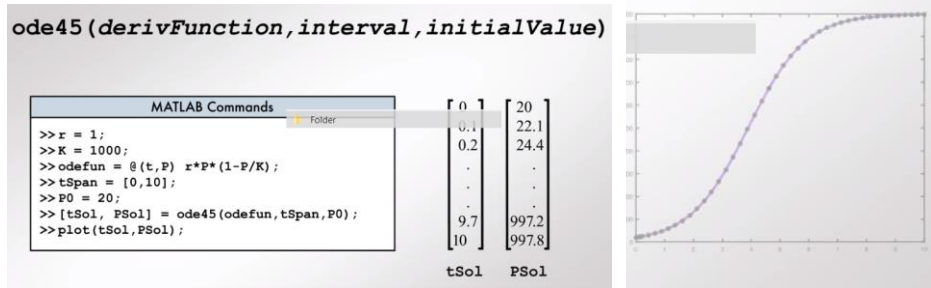
[tSol,ySol] = ode45(derivFunction,interval,initialValue)

$$tSol = \begin{bmatrix} tStart \\ \vdots \\ tEnd \end{bmatrix}$$

$$ySol = \begin{bmatrix} yStart \\ \vdots \\ yEnd \end{bmatrix}$$

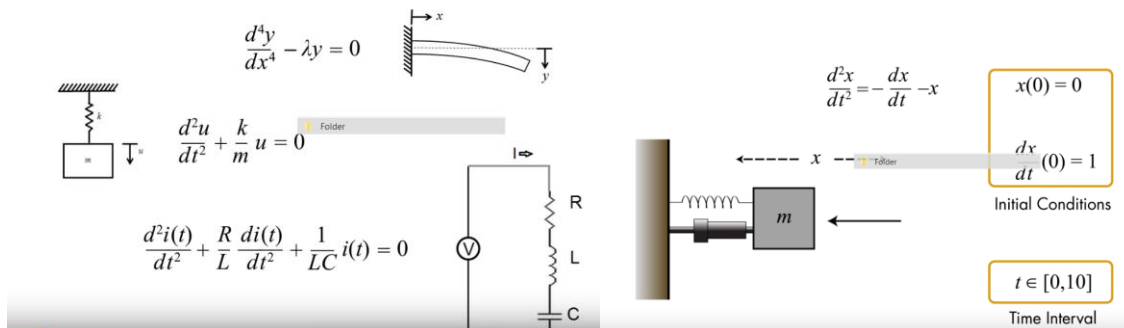
both tSol and ySol are vectors

3.eg

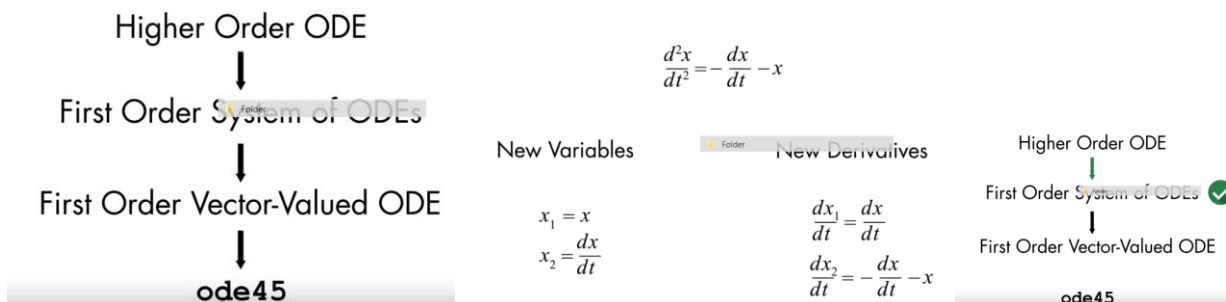


For corresponding tsol we have respective psol values

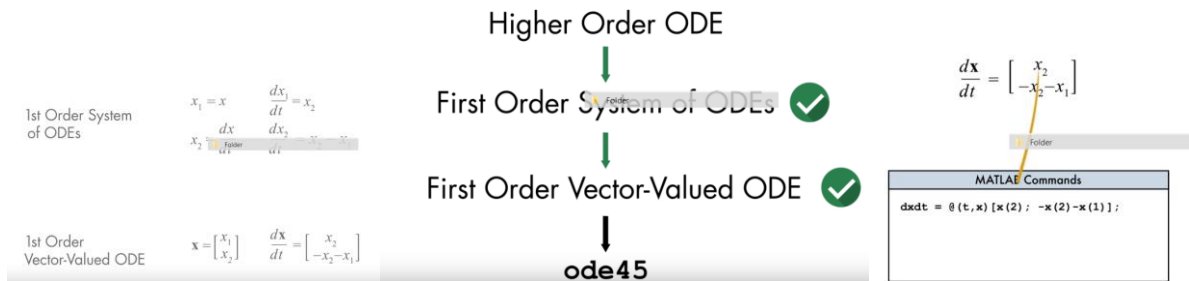
5.



Note: Since Matlab ODE45 tool box only can solve 1st order ODE, any higher order diff equations need to be converted to 1st order first and then given to MATLAB api



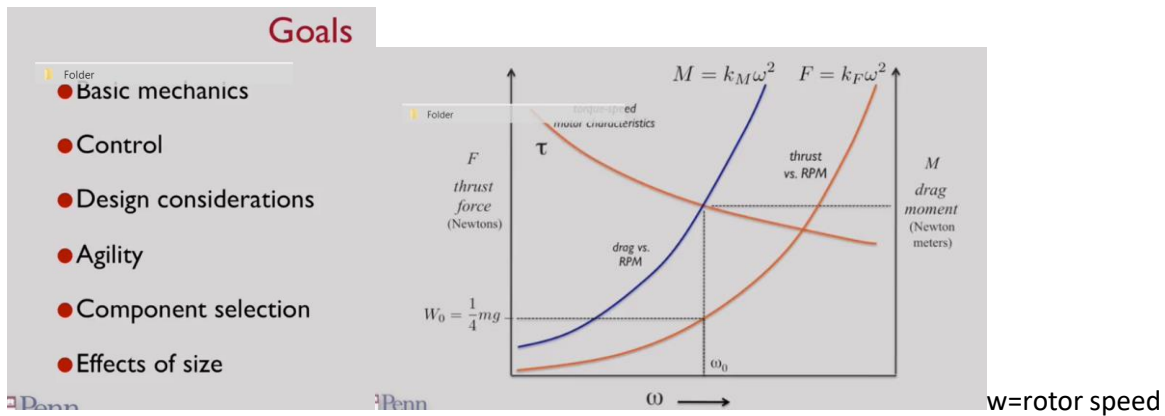
Now multiple variables need to be represented in to single variable



tSol	xSol
0	0 1.0000
0.0001	0.0001 0.9999
...	...
...	...
9.9953	0.0057 -0.0077
10.0000	0.0054 -0.0076
t	x dx/dt

////////////////////////////////////

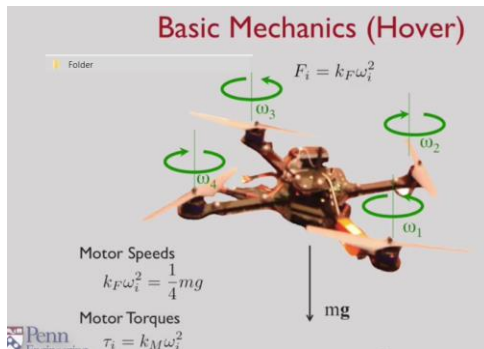
Basic Mechanics:



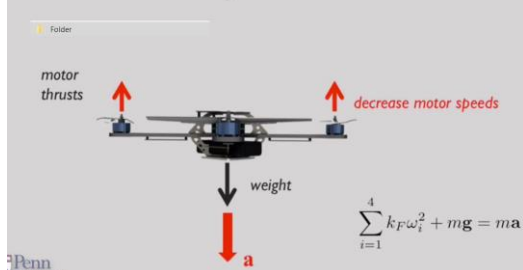
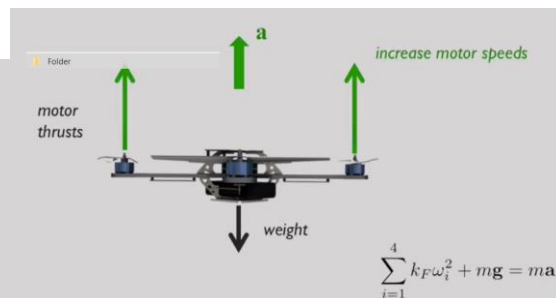
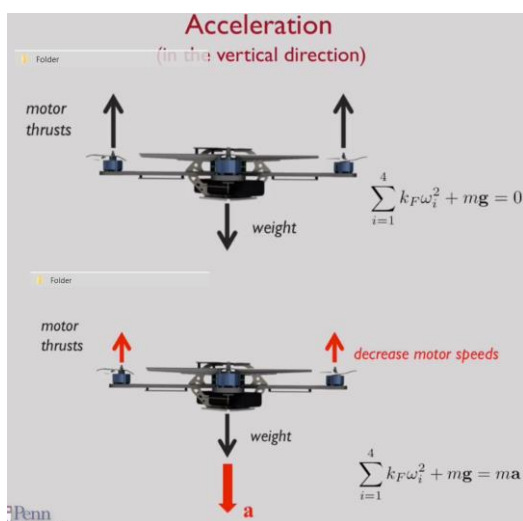
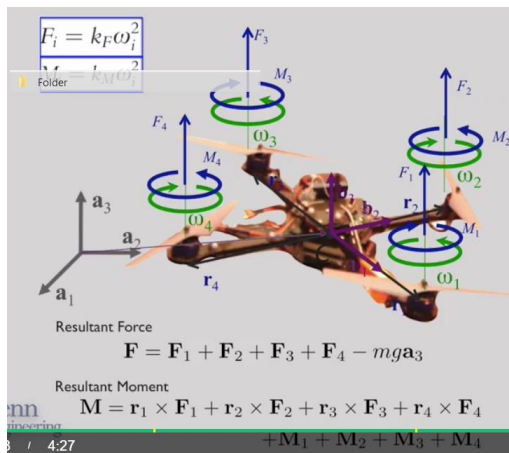
a. when rotor spins it has to overcome drag

b. since 4 rotors..each rotor has to support $w_0 = 1/4 w_g$ to be in equilibrium

c. A quadrotor is at hover and the speeds of all its motors double, it will **Accelerate Up**



k_F =force w=rpm M_i =Drag moment K_M =drag force



Dynamical Systems

Systems where the effects of actions do not occur immediately

State: a collection of variables that completely characterizes the motion of a system

Eg: A Thermostat in room. Temp in room will increase slowly but not instantly

Dynamical Systems

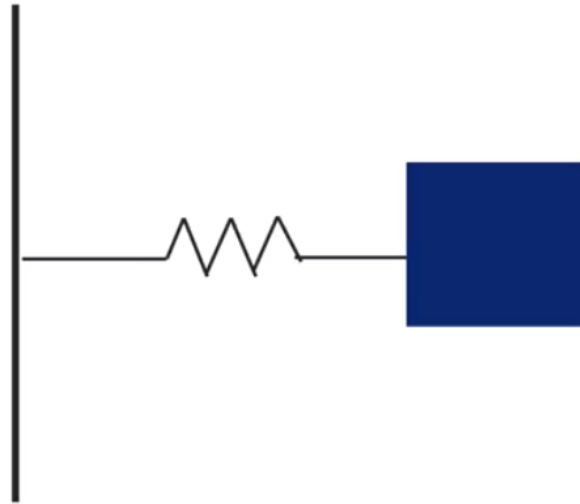
Evolution of these states over time is often given by a set of governing ordinary differential equations

- Order: highest derivative that appears in the equations

$$\boxed{\ddot{x}(t)} = u(t) \quad \text{Second-order system}$$

Since

Example 1: Mass-Spring System

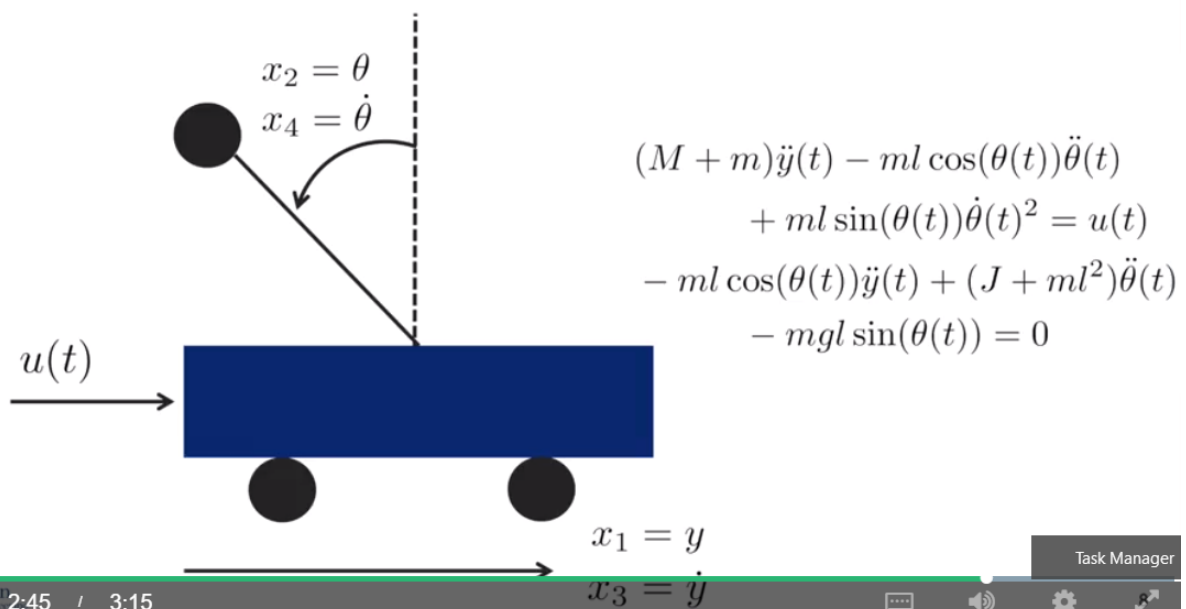


$$m\ddot{y}(t) + ky(t) = u(t)$$

Since highest derivative is 2nd order, the example 1 is 2nd order dynamic system

State of system = position and velocity of MASS

Example 2: Pendulum on a Cart



Here above system has 4 states = vel of cart ,vel of pendulum ,pos of cart,pos of pendulum

Example 3: Quadrotor



$$x_1 = x \qquad x_7 = \dot{x}$$

$$x_2 = y \qquad x_8 = \dot{y}$$

$$x_3 = z \qquad x_9 = \dot{z}$$

$$x_4 = \phi \qquad x_{10} = p$$

$$x_5 = \theta \qquad x_{11} = q$$

$$x_6 = \psi \qquad x_{12} = r$$

States -12

3 Lin vel ,3 ang vel ,3 pos,3 angular orientation

Rates of Convergence

How fast do we want this error to go to 0?

- The error *exponentially converges* to 0 if there exists constants α and β and time t_0 , such that for all $t \geq t_0$:

$$\|e(t)\| \leq \alpha e^{-\beta t}$$

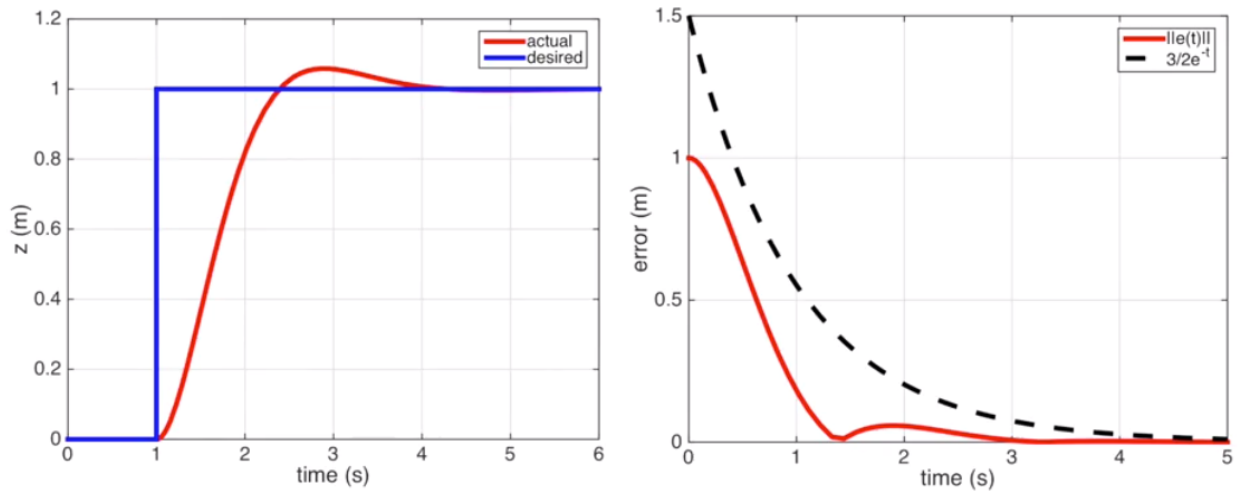
Feedback Control

Recall the control problem

Determine the appropriate input that will cause the error between the desired state and the actual state of a dynamical system to eventually reach 0.

$$e(t) = x^{des}(t) - x(t) \rightarrow 0 \text{ as } t \rightarrow \infty$$

Example I: PD Controller



Feedback Control

Here we will accomplish this using a PD (or PID) controller.

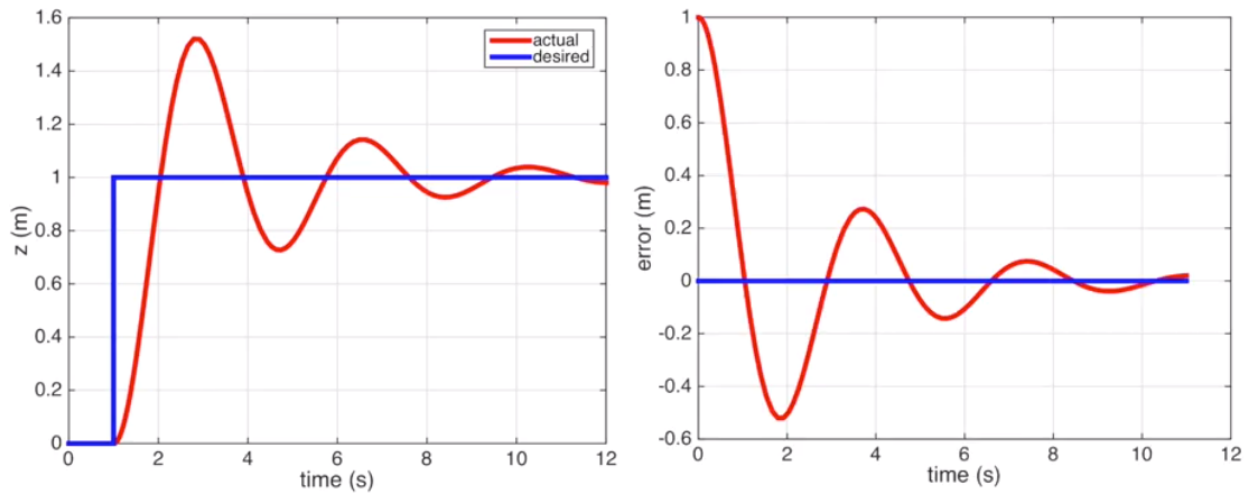
$$u(t) = \ddot{x}^{\text{des}}(t) + K_v \dot{e}(t) + K_p e(t)$$

Consider the controllers we used before to control the height of a quadrotor.

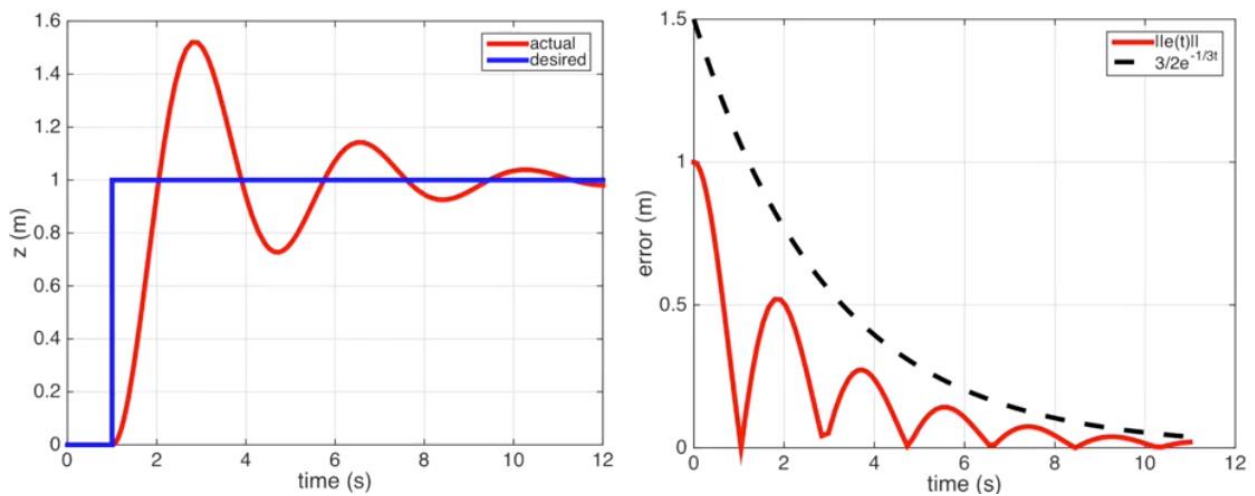
Left Fig : Reaching desired state

Right Fig: Error of feedback control getting converged to zero (Red=absolute error, black=exp error)

Example 2: High K_p



Example 2: High K_p



Right fig: Absolute error.

SO even though we see oscillation but convergence is happening exponentiallly.

Therefore in PD controller whatever is error function either oscillatory or other, Error fuction converges exponentiall