PAGE

(~~Kotlin~~)

(Robot diaries)

① 3000 mAH = (3000) (mA × H) = 3000 $\left(\frac{A}{1000}\right)$ × (H)

- this means if your circuit takes up for eg (1000 mA) then the battery Rated as 3000 mAH will Run ~~of~~ for 3 Hours.

$\left(\text{current} = \frac{\text{charge}}{\text{time}}\right)$ (charge = current × time = $\underbrace{\text{mAH}}_{\text{in general}}$)

② Motors (DC)

voltage → Rated voltage is the most efficient ~~output~~ Performance Voltage
less V = won't work / More V = Burn down

current → operating current = Avg. current drawn in normal working

stall current = the maximum current drawn.
like when you don't let it Rotate & still Power up.

$\text{Power}^{(\text{watt})}$ = voltage * current

power spike
if you want to change the direction of Rotation of your motor Running at full Force, then you need to Provide extra voltage to counter the "inductance" & "momentum".
this voltage will be 2* operating voltage
& current will be stall current
Acc. ~~to~~ to this ~~Rea~~ Voltage & Current we should make our circuit.

Torque → operating Torque → at normal voltage & normal current
stall Torque → at normal voltage & stall current
more torque = more acceleration

A safe to go Rule :-

if you use 2 motors for a wheeled Robot

(stall torque of each = weight of entire Robot × Radius of wheel)

20% above the Rated voltage is safe, Provides extra torque, might need heat sink.

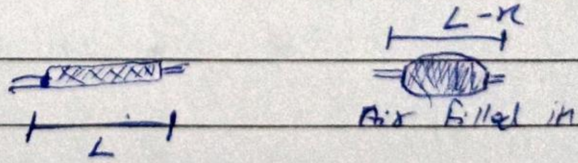
efficiency Per gear = 90% of original
for eg (2 gears = 90% × 90% = 81% efficiency

Solenoid (Actuators)

→ Remember stroke distance is sufficient for your application

Type → electromechanical (solenoid / electricity / Magnetics)
Hydraulic (water / oil)
Pneumatics (air)

Pneumatic air muscle →



→ Piezoelectric Effect.

DAIRY BiKe

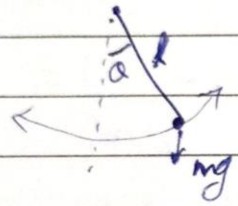
1.1

1) Any Physical sys is expressed by set of linear equations.
state variable are variable that change in the system.
Input variable " " " cause external changes.

A general state eq^n => $\dot{x} = Ax + Bu$

A = state jacobian Matrix = $\underset{\text{No of variable (state variable)}}{\longrightarrow}$ × ↓ No of eq^n — this determine dimension of Matrix
B = input jacobian Matrix = → No of input variable, ↓ No of eq^n

2) Eg :- For a simple Pendulum system



here state variable are (θ) & ($\omega = \dot{\theta}$)
Named as → x_1 & x_2

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{K}{m} x_2$$
} check yourself

3) Solving these eq^n for equilibrium Point.
=> Points where state of the sys. dont change.
=> all ~~fact~~ state variable are considered const.
=> diff. of all state variable = 0.

for eg :
$$\dot{x}_1 = -x_1 + 2x_1^3 + x_2$$
$$\dot{x}_2 = -x_1 - x_2$$

=> To find eq^n Point $\dot{x}_1 = 0$ & $\dot{x}_2 = 0$
=> Now for stability first find jacobian then eigen value
=> eigen value on left side of imaginary plane are stable, on right are unstable & zero (Real Part) are Partly stable.

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}, \quad |SI - J_1| = 0$$

DATE
PAGE

1.2

~~Another Method of deriving and solving eqn / Mathematical Modelling~~

~~state eq Equation =>~~ $\cancel{\dot{x}(t) = \underbrace{A}_{\text{Jacobian Matrix}} \overbrace{x(t)}^{\text{state variable}} + \underbrace{B}_{\text{Jacobian Matrix}} \overbrace{u(t)}^{\text{input variable}}}$

~~Steps~~ Mathematical Modeling of system

1) => Lagrange of a system => K.E − P.E = L

=> Eqn of Motion of system => $\dfrac{d}{dt}\left(\dfrac{\partial L}{\partial \dot{x}}\right) - \dfrac{\partial L}{\partial x} = 0$ — Euler-Lagrange Method

output will have an eqn with state variable

Partial diff. (all other zero except for given denominator)

=> For system with ext. force or torque :-

Eqn of Motion of system => $\dfrac{d}{dt}\left(\dfrac{\partial L}{\partial \dot{x}}\right) - \left(\dfrac{\partial L}{\partial x}\right) = T \text{ or } F$

output will have an eqn with input variable & state variable

2) state Equation => $\dot{x}(t) = Ax(t)$

or

$\dot{x}(t) = \underbrace{A}_{\text{Jacobian Matrix}} \overbrace{x(t)}^{\text{state variable}} + \underbrace{B}_{\text{Jacobian Matrix}} \overbrace{u(t)}^{\text{input variable}}$ $\quad J_B = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix}$

Eg :-

$\left.\begin{aligned} \dot{x}_1 &= x_1 x_2 - x_2 \\ \dot{x}_2 &= 2x_1 - x_2^2 \end{aligned}\right\}$ Not equal Hence do the Process below

$\{ \dot{x}(t) = Ax(t)$ since there is no input variable

Find Jacobian →

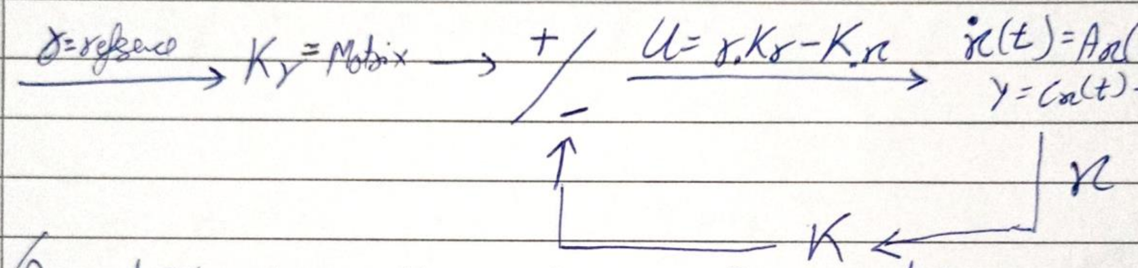
Now $\underset{\text{state eqn}}{\dot{x} = Ax}$ $\left(\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}, \; A = \text{Jacobian}, \; x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)$

This state eqn can be further written as state eqn of eqn point by putting eqm point in Jacobian.

stability is always checked for J_A or A and not for J_B or B

→ controllability & observability of system:-

3) controller design :-



(Remember, since, u = Input
and since Input force is zero at start
hence Refrence is = 0)

New state eqn = $\dot{x} = Ax + B(rK_r - Kx) = (A - BK)x + B_r K_r$

Now choose the K Matrix such that Resultant eigen value give a stable System.

2 method (Pole placement & LQR)

choose 2 Point on -ve x-axis of imaginary plane & use them as eigen value to find the unknown

Remember A & BK are of same order hence
if A = 2×2 then B = $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$

& $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$

More better method that take in account the Relative priority of state variable

q = diagonal Matrix with Relative value for respective state variable.

(all calculation are done by internal fn.)

.8 – 1m Radius