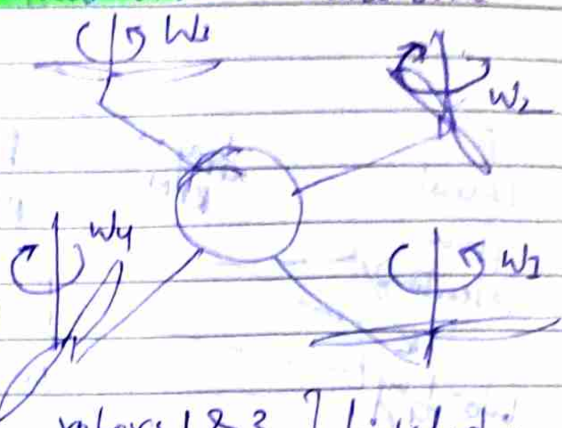
Quadrotors

Date __/__/____

Consists of 4 independently controlled rotors mounted on a rigid frame.



→ (6 DOF)

rotors 1 & 3
rotors 2 & 4 } pitched in opposite dir^n.

→ In any ~~...~~ system, ask how the following components work!

- ✓ State Estimation
- ✓ Control
- ✓ Mapping
- ✓ Planning

Estimate the pos & vel
(inc. rot & angular vel of the robot)

(Capability to map its environment;

(Command motors & produce desired action in order to navigate to desired state)

Planning safe trajectories)

→ (given a set of obstacles & given a destination, the vehicle must be able to compute a trajectory, a safe path to go from one point to another)

State Estimation — obtain reliable estimates of position & vel.

motion Capture Camera

How to navigate without GPS or external motion Capture cameras?

- USB Cameras
- 3D Camera
- Laser Rangefinder

KMel Robotics

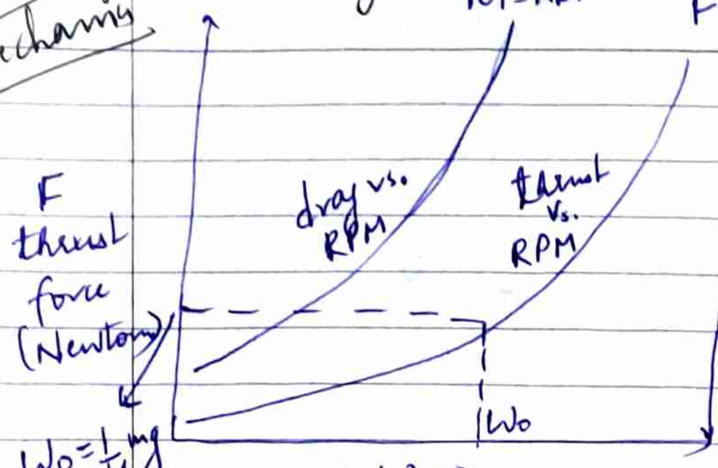
Basic Mechanics

Rotor Physics

Const. of proportionality (k_M)

$M = k_M \omega^2$ $F = k_F \omega^2$

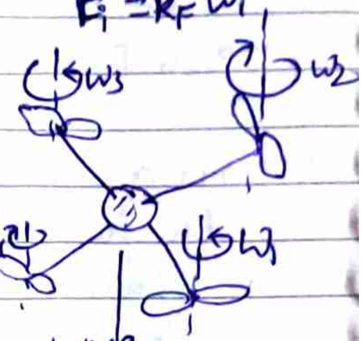
(Relationship is approximately quadratic)



$\omega_0 = \frac{1}{4} mg$

rotor speed (rad/s or RPM)

RPM or angular velocity of the motor



Motor speeds

$k_F \omega_i^2 = \frac{1}{4} mg$

mg

→ A quadrotor has 4 rotors that support the vehicle weight. So, each rotor spins and generates the thrust.

→ Relationship between the thrust force and the RPMs/angular velocity of motor is approximately quadratic.

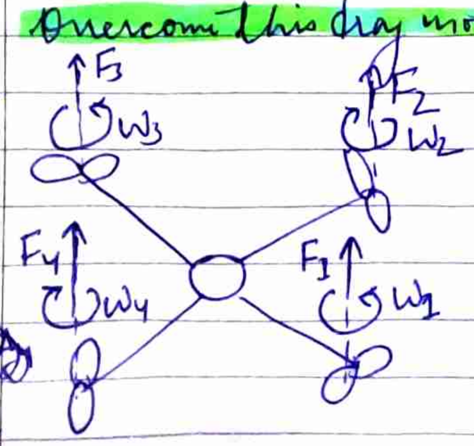
→ Every time a rotor spins, there is also a drag that the rotor has to overcome. ⇒ that drag moment is also quadratic.

→ So, every rotor has to support roughly 1/4 th of the weight in equilibrium which means by looking at the thrust force vs RPM curve, we can determine speed which will be required to produce 1/4 th of the weight → so that gives us ω_0 the operating speed.

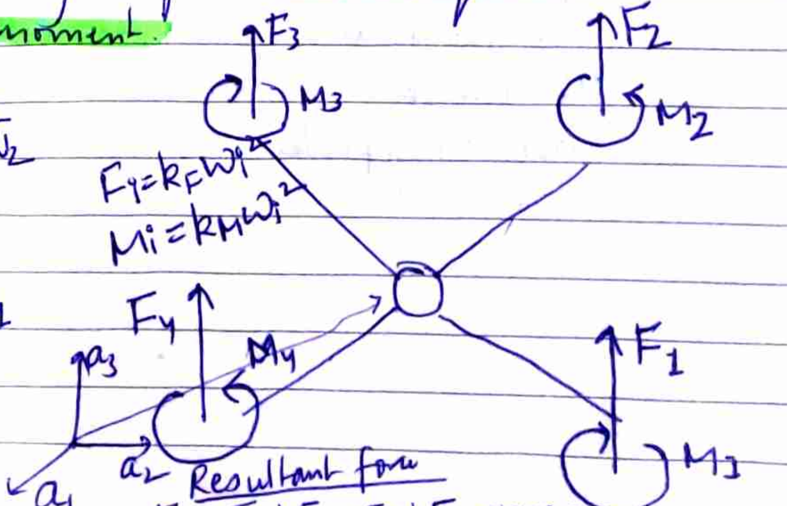
(that operating speed produces a drag moment & every rotor has to overcome the drag moment)

⇒ that's when motors come in, we have to size the motor so that they can produce the torque to overcome this drag moment.

$F_i = k_F \omega_i^2$



dirn of ω_i & M_i is opposite

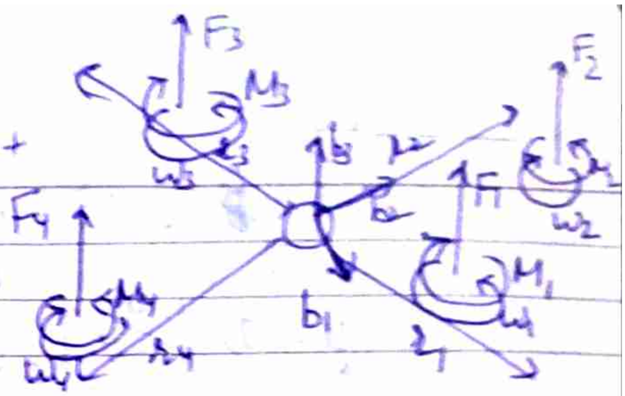


Resultant force

$F = F_1 + F_2 + F_3 + F_4 - mg\,a_3$

Resultant Moment

$M = r_1 \times F_1 + r_2 \times F_2 + r_3 \times F_3 + r_4 \times F_4 + M_1 + M_2 + M_3 + M_4$



⇒ the total moment is obtained by calculating the moments due to the forces ~~exerted~~ exerted by the rotors & the reaction due to the rotors spinning in counterclockwise or clockwise directions

⇒ these reaction are moments & they add to the net moment.

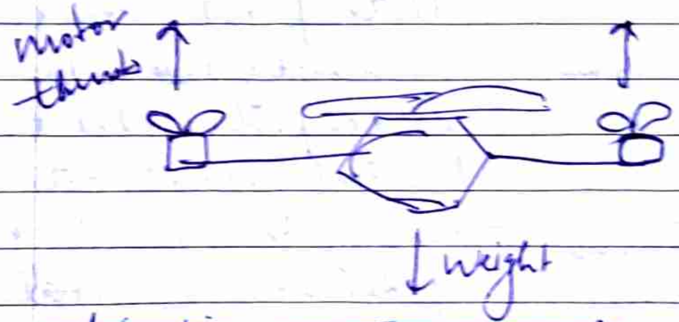
In equilibrium, Resultant F = 0

Resultant moment = 0

★★

But, what happens when these resultant force & moment are not 0? ⇒ acceleration

Acceleration (in the vertical direction)



$$\sum_{i=1}^{4} k_F w_i^2 + mg = 0$$

→ (Every motor thrust is the same & they will add up to support the weight)

→ But if we ↑ the motor speeds then robot accelerates up

→ If ↓ motor speed, robot will accelerate down.

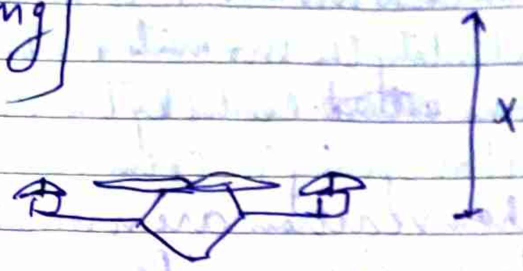
a combination of motor thrusts & the weight determines which way the robot accelerates.

Control of height

$$\sum_{i=1}^{4} K_F w_i^2 + mg = ma = m\frac{d^2x}{dt^2} = m\ddot{x}$$

$$u = \frac{1}{m}\left[\sum_{i=1}^{4} K_F w_i^2 + mg\right]$$

// $\frac{\text{sum of forces}}{\text{mass}}$



Second order dynamic system $u = \ddot{x}$

what input drives the robot to the desired position?

What are Dynamical Systems?

→ Systems where the effects of actions do not occur immediately.

→ eg. if we turn on the thermostat in a cold room, the temp in the room will not immediately rise to the set temp. It will take some time for the room to actually heat up.

→ Similarly, if we push the gas pedal in our car, it takes time for the car to speed up to the desired velocity.

→ Every dynamical system is defined by its state which is a collection of variables that completely characterizes the motion of a system

→ the most common states are the position & vel of physical components of the system. **

X — states of a system

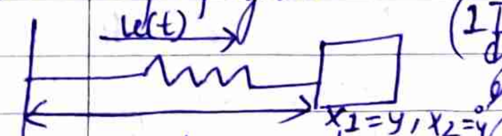
$X(t)$ → value of a system's states over time.

Dynamical System ** Evolution of these states over time is often given by a set of governing ODEs.

→ Order → highest derivative that appears in the eqn.

$\ddot{x}(t) = u(t)$ → 2nd order system

Ex ① Mass spring system



(1D dynamical system)

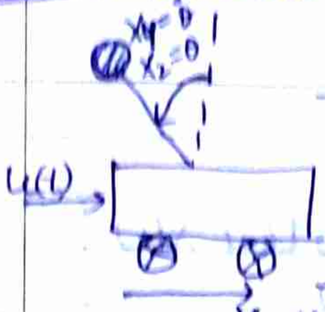
mass can only move backwards & forwards in the y-dir^n.

The mass's position is governed by following ODE → $m\ddot{y}(t) + Ky(t) = u(t)$ (2nd order system)

The state of the system is the position and velocity of the mass along the y-axis.

→ Input to this system is an additional force on the mass.

Ex-2 Pendulum on a Cart (2 dimensional dynamical system)



- → the cart is allowed to drive along the y dirn while the pendulum is simultaneously allowed to fall
- → The motion of this system to be modeled by the following set of coupled ODE. —

$x_1 = y$
$x_3 = \dot y$

2nd order system.

$$(M+m)\ddot{y}(t) - ml\cos(\theta(t))\ddot\theta(t) + ml\sin(\theta(t))\dot\theta(t)^2 = u(t)$$

$$-ml\cos(\theta(t))\ddot{y}(t) + (J+ml^2)\ddot\theta(t) - mgl\sin(\theta(t)) = 0$$

Four state in this system — positions of the cart & the pendulum, velocities of the cart & the pendulum

The input is an additional force on the cart itself.
Here, we assume that we cannot directly apply a force to the pendulum.

Ex-3 Quadrotor

→ To completely characterise a motion of the Quadrotor, we need to know its x, y, z position & angular orientation as well as its linear & angular velocities.

$x_1 = x$	$x_4 = \phi$	$x_7 = \dot x$	$x_{10} = p$
$x_2 = y$	$x_5 = \theta$	$x_8 = \dot y$	$x_{11} = q$
$x_3 = z$	$x_6 = \psi$	$x_9 = \dot z$	$x_{12} = r$

Control of a linear second-order system

Problem

State, Input $\quad x, u \in \mathbb{R}$

Plant model $\quad \ddot{x} = u$

Find a control input function $u(t)$ so that $x(t)$ follows the desired trajectory $x^{des}(t)$

General Approach — Define error,

$$e(t) = x^{des}(t) - x(t)$$

desired trajectory — actual trajectory

want $e(t)$ to converge exponentially to 0

Strategy —

Find $u(t)$ such that

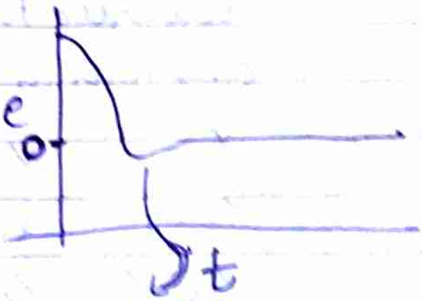
diff^n eq^n: $$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad , K_p, K_v > 0$$

(if we ensure that $K_p, K_v > 0$, then this error will go exponentially to 0)

Control eq^n: $$u(t) = \ddot{x}^{des}(t) + K_v \dot{e}(t) + K_p e(t)$$

Derivative gain — Proportional gain

feedforward term: We need some knowledge of how we want the traj to vary. so we are feeding forward the 2nd derivative of the desired trajectory



Undershoot

The error might undershoot, go from +ve → -ve but eventually we're guaranteed that it'll be 0.

Control for trajectory tracking in a simple second-order system

PD control

$$u(t) = \underbrace{\ddot{x}^{des}(t)}_{\text{feedforward term}} + \underbrace{K_v \dot{e}(t)}_{\text{Derivative term}} + \underbrace{K_p e(t)}_{\text{proportional term}}$$

→ The proportional term acts like a spring or a capitance.
The higher the proportional term is, the more springy the system becomes and more likely it is to overshoot.

→ The higher the derivative term, the more dense it becomes. So, this is like a viscous dashpot or a resistance in an electrical system.
By ↑ the derivative gain, the system essentially gets damped, and we can make it overdamped so that it never overshoots the desired value.

→ In exceptional cases, we might consider using a more sophisticated version of the proportional + derivative control.

PID control

$$\boxed{u(t) = \ddot{x}^{des}(t) + K_v \dot{e}(t) + K_p e(t) + \underbrace{K_i \int_0^t e(\tau)\, d\tau}_{\text{extra term}}}$$

In the presence of disturbances (eg. wind) or modeling errors (eg. unknown mass), it is often advantageous to use PID control.

extra term (proportional to the integral of the error) (we often do this when we don't know the model exactly, so for instance, we might not know the mass, or there might be some wind resistance that we need to overcome & we don't have a priori how much this wind resistance is)

So, the last term essentially allows us to compensate for unknown effects caused by either unknown quantities or unknown wind conditions or disturbances.

drawback of adding an extra ∫ term

Now, differential eqn becomes a 3 order differential eqn.

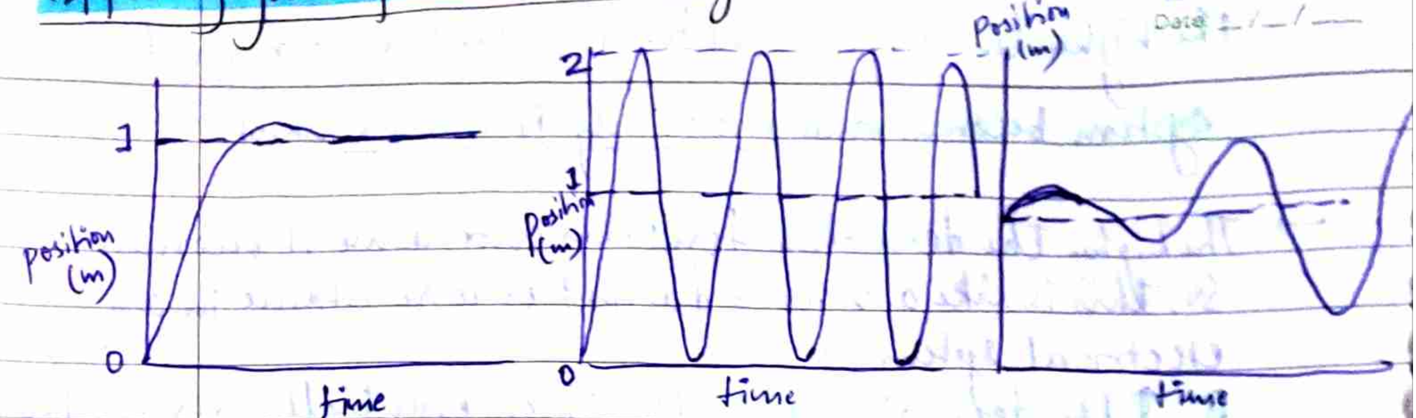
PID control generates a third-order closed loop system.

benefit

Integral control makes the steady-state error go to 0.
(the integral term will make the error go to 0 eventually)

Effects of Gains for a PD Control system

Date __/__/__

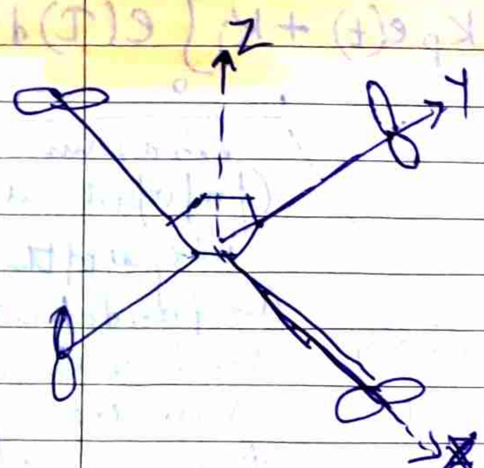


Stable | Marginally Stable | Unstable

$K_p, K_v > 0$ (If both the gains are +ve, we are guaranteed stability)

$K_p > 0, K_v = 0$ (while the system will not drift, it will oscillate about the desired value)

K_p or $K_v < 0$



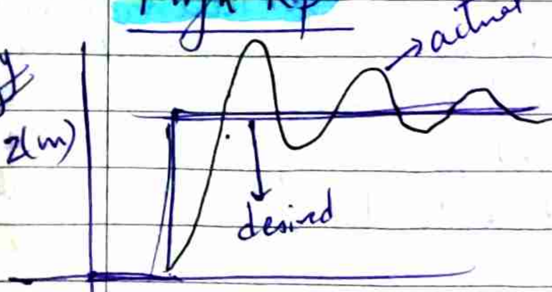
→ The robot is asked to hover & in this case someone displaces the robot from the nominal hover position.

→ The robot fights to overcome the disturbance

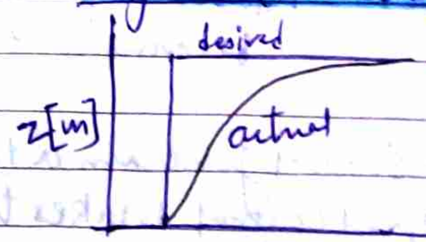
→ Using a combination of proportional & derivative gains, the robot is able to compensate for that disturbance & then settle back down into the hover position.

High Kp

Springy



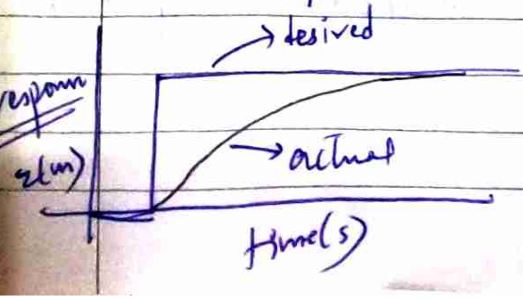
High Kv (overdamped)



→ the overshoot disappears but the system takes also a longer time in order to get to the desired position

Low Kp

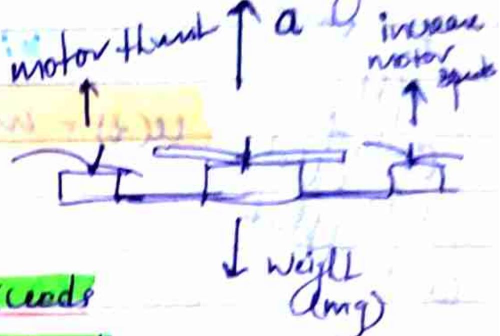
Soft response



Design Considerations

Previously, we assumed that the motors were capable of producing thrust the controller required.
So, in reality, if we look at this model, the motor thrusts are limited because the motors have a limited capacity.

Effect of motor thrust



$$\sum_{i=1}^{4} K_F \omega_i^2 + mg = ma$$

Clearly, the thrusts have to compensate for the weight and the thrust that exceeds the weight will produce the required accⁿ.
But, ~~that thrust produced by the motor is limited by~~ (maxᵐ) that the motor can produce is limited by the peak torque.

Let's Assume that this peak torque is known to us and that in turn determine the maximum thrust we can produce, T_{max}.

[Maxᵐ thrust $\Rightarrow T_{max}$]

This in turn determine the maximum acceleration

[Maxᵐ acceleration $\Rightarrow a_{max}$]

Control with Thrust limitation

Effect of maximum thrust on Input

$$u = \frac{1}{m}\left[\sum_{i=1}^{4} K_F \omega_i^2 + mg\right]$$ Input, defined in terms of the thrust.

$$= \frac{1}{m}\left[T\uparrow + mg\downarrow\right]$$ (T points in ↑ dirⁿ while weight in ↓ dirⁿ)

Assumption → We Know T_{max} (Maxᵐ thrust as determined by peak motor torque)

$$u_{max} = \frac{1}{m}\left[T_{max} + mg\right]$$

PD control

$$u(t) = \min\left(\underbrace{\ddot{x}^{des}(t) + K_v \dot{e}(t) + K_p e(t)}, u_{max}\right)$$

⇒ $u(t)$ is determine not just by the proportion of the derivation control law, but also the maximum thrust that can be applied.

the value of the control input that can be applied.

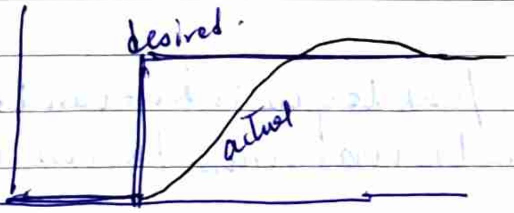
PID control

$$u(t) = \min\left(\ddot{x}^{des}(t) + K_v \dot{e}(t) + K_p e(t) + K_I \int_0^t e(\tau)d\tau,\ u_{max}\right)$$

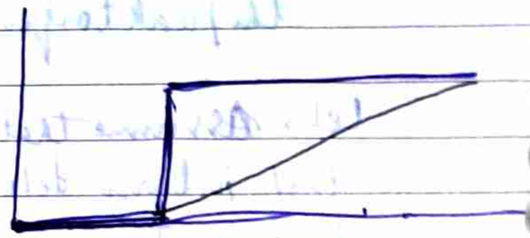
Effect of the Thrust/Weight Ratio

What happens if the payload of the robot is increased (with the same motors and propellers)?

The higher the thrust-to-weight ratio, the easier it is to control the drone in elaborate aerobatics. We can use 2:1 for gentle flying

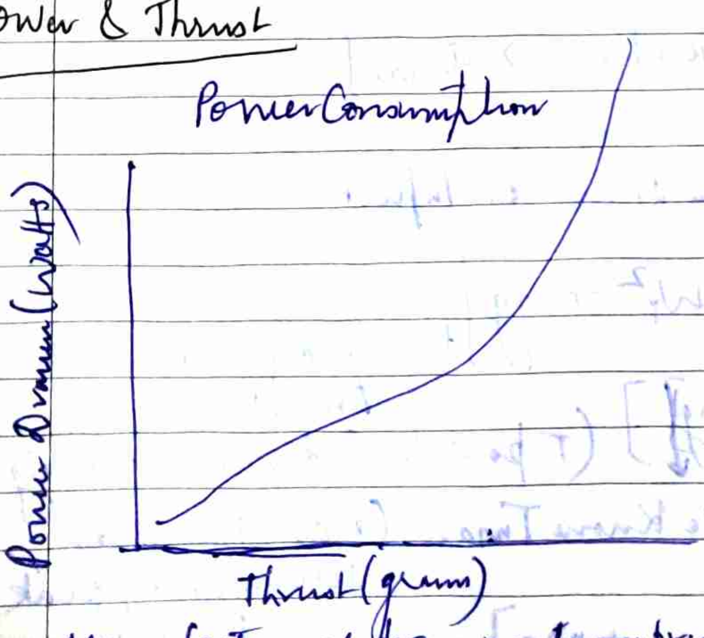


Thrust/weight = 2



Thrust/Weight = 1.2

Power & Thrust



Q

$T_{max} = 11.77N$ $(1.2kgf)$

$v = u + at$

$s = \frac{at^2}{2}$

$T - mg = ma$

$T = m(g+a)$

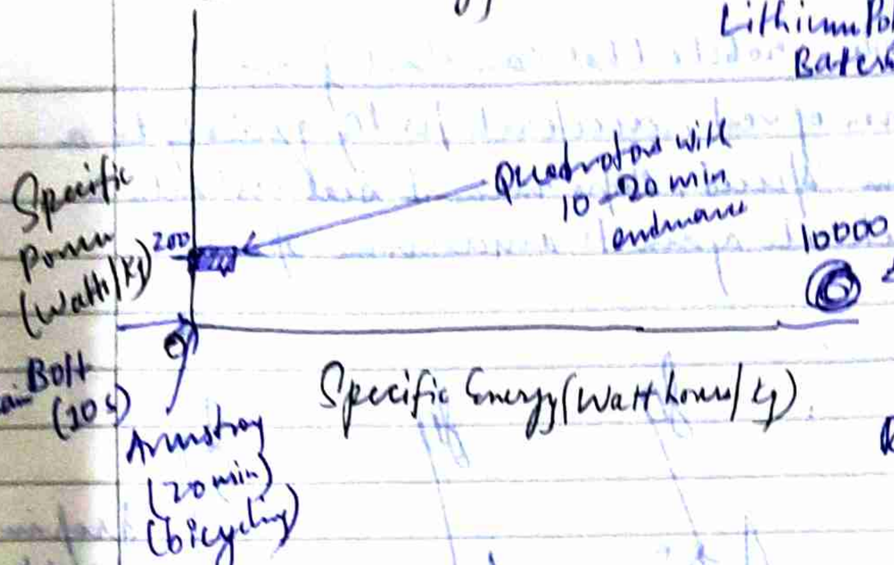
$m = \frac{T}{g+a}$

$= \frac{11.77}{9.8+}$

$1.2kgf$

Manufacturers of drone motors express the thrust in grams or ounces. This is the weight that one motor can lift off the ground. It doesn't mean, though, that a thrust equal to drone's weight is sufficient

Power & Energy



Lithium Polymer Batteries (produce around 200 watt hours per kilo)

Human's adipose tissue w/ fat carries about (10K) watt hours per kilo.

Robots consume about 200 watts per kill per hour. (Robots are ~~~~ 10 x more inefficient than possibly the most inefficient man on earth)

So, what do we do when we need a lot of power & we don't have batteries that carry a lot of power?
→ Reduce payload/weight.

Mass distribution in drone

Batteries ~ 33% mass
Motors ~ 25 % mass

Sensors & Power

Laser Sensor

270 gm
10 W for operation + (50-60) W for mobility
Range 30 m

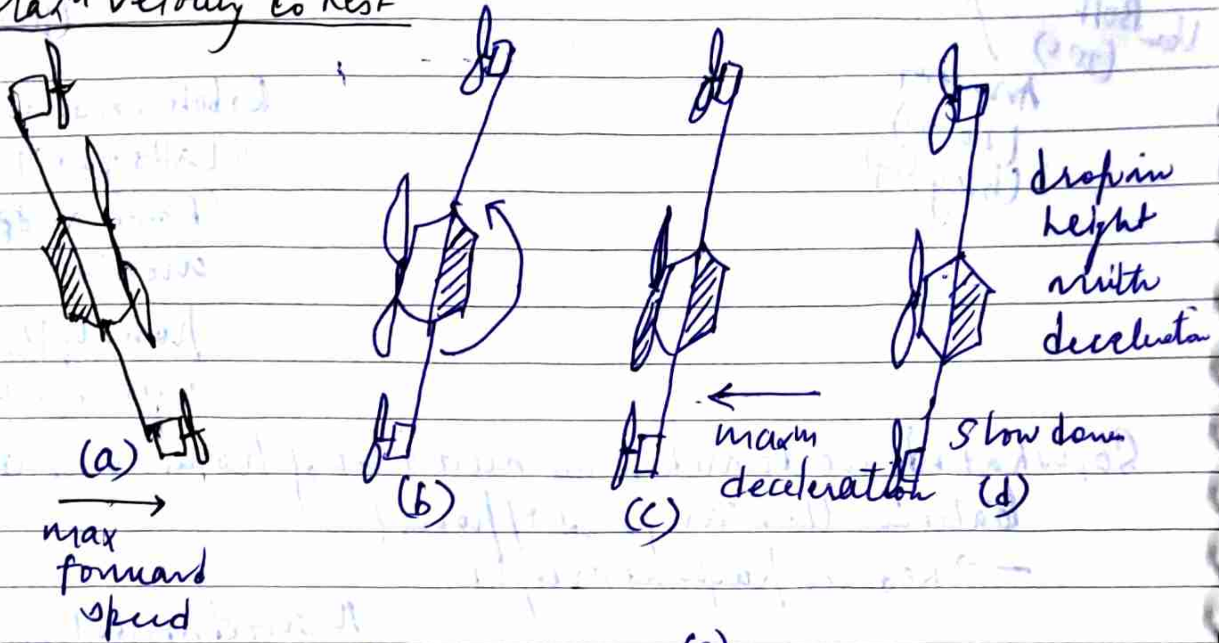
Cameras

80 gm (incl. frame, each cam 25g)
10.5 W for operation + 15 W for mobility
Range 10-15 m.

Agility & Maneuverability

↳ agile robots – robots that can start from a position of rest, accelerate pretty quickly to a maximum speed, stops when it sees obstacle & then accelerate again to a maximum speed.

Max^{m} velocity to Rest



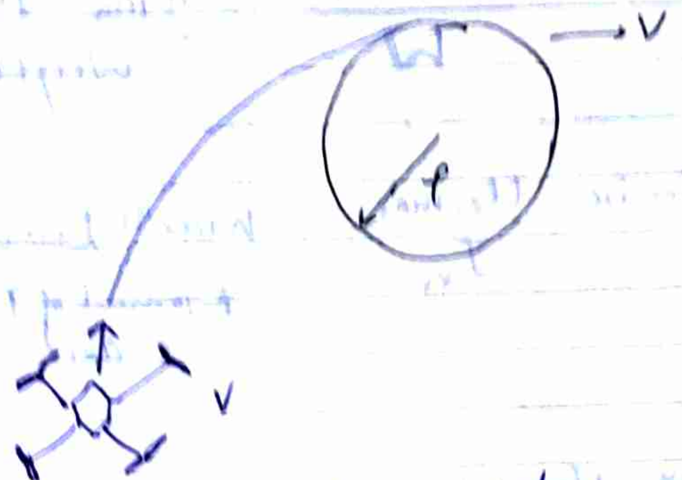
→ first, the robot is pitching forward (a) when it's going at max^{m} speed. when we decide to bring it to a position of rest, we must pitch it backward (b), reversing the direction of thrust so that we get deceleration (c).

→ This means that the robot must be pitching back at an aggressive angle. This will cause the robot to slow down (d).

→ But as a result, the thrust factor which now points in a direction other than the vertical dir^{n} will also cause the robot to loose height because the component of thrust in the vertical dir^{n} is now less than the weight.

Maximize Agility → minimize stopping distance

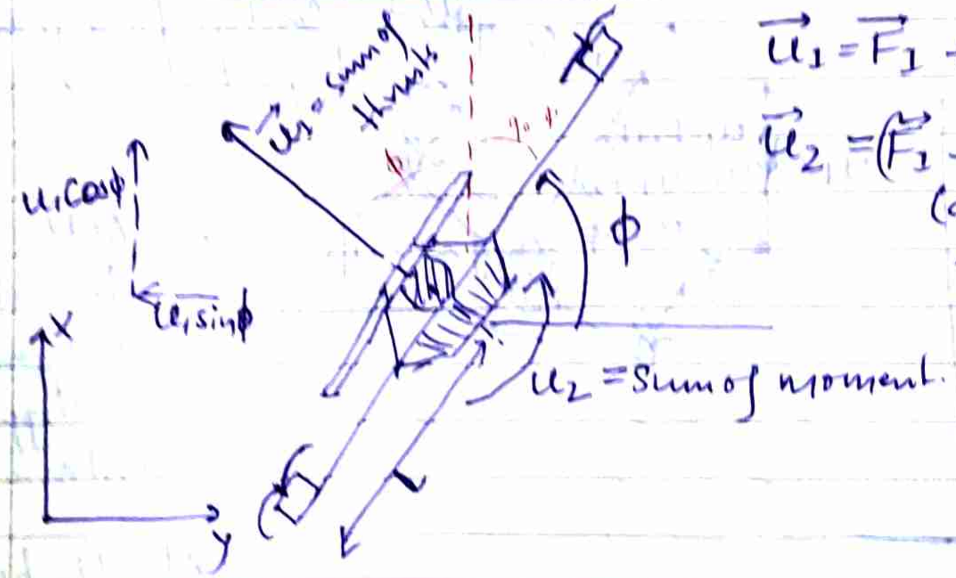
Turn Quickly without Slowing Down



Minimize the turning radius (ρ)

→ Stopping from max^m speed and turning at max^m speed, is actually sufficient to consider a fairly simple model of a quadrotor

Quadrotor in a Vertical plane



(propellers apply a thrust)

$\vec{u}_1 = \vec{F}_1 + \vec{F}_2$ — sum of 4 thrust (in case of a quadrotor)

$\vec{u}_2 = (\vec{F}_1 - \vec{F}_2)\,l$

(difference of the thrust contribute to the moments)

$$\begin{array}{l} \text{linear acceleration, } a \\ \\ \text{angular acc}^n\text{, } \alpha \end{array} \left\{ \begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{m}\sin\phi & 0 \\ \frac{1}{m}\cos\phi & 0 \\ 0 & \frac{1}{I_{xx}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \right.$$

Agility

Two key ideas

✓ Accelerate quickly → maximize linear acc^n, a_{max}

✓ Roll / pitch quickly → maximize angular acc^n, α_{max}

#learnthesmarterway

To maximize a, i.e., a_{max}

maximize $\frac{u_{1,max}}{W} = \frac{\text{max}^{m}\text{ thrust}}{\text{Weight}}$

To maximize α, i.e., α_{max}

maximize $\frac{u_{2,max}}{I_{xx}} = \frac{\text{max}^{m}\text{ turning moment}}{\text{Moment of Inertia along x-axis}}$

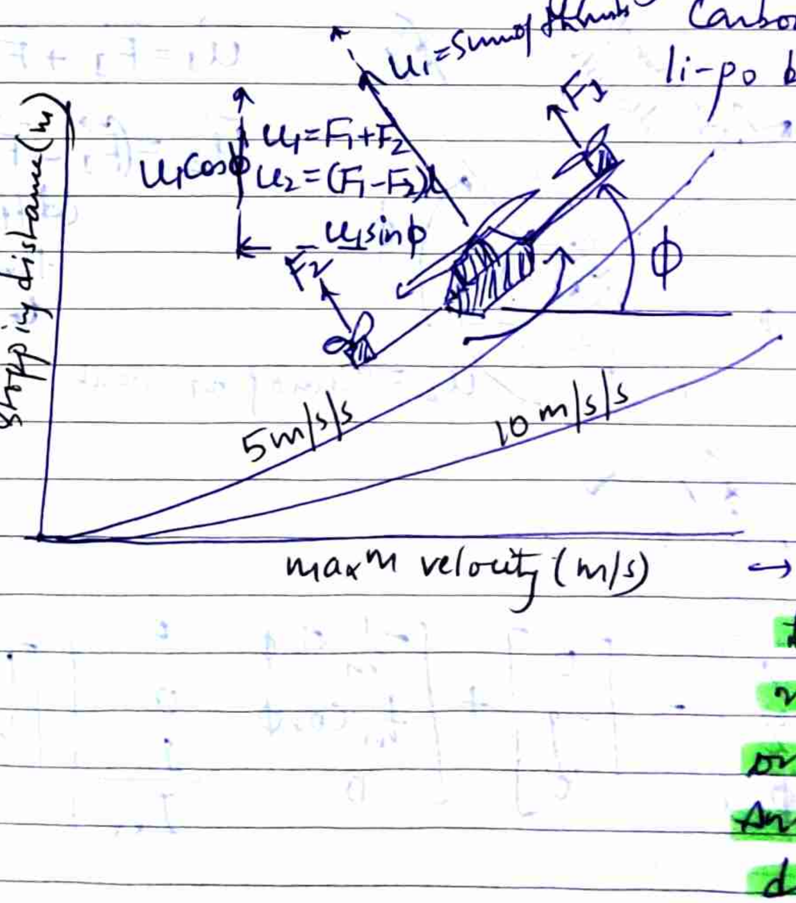
Stopping Distance

__Assumptions__ Thrust/weight ratio = 2

Assume robot can drop in height while turning

0 to 90 deg ~ 0.25 sec (1), 0.5 sec (2)

Conventional technology (eg. dc motors, Carbon fibre frame, li-po batteries)

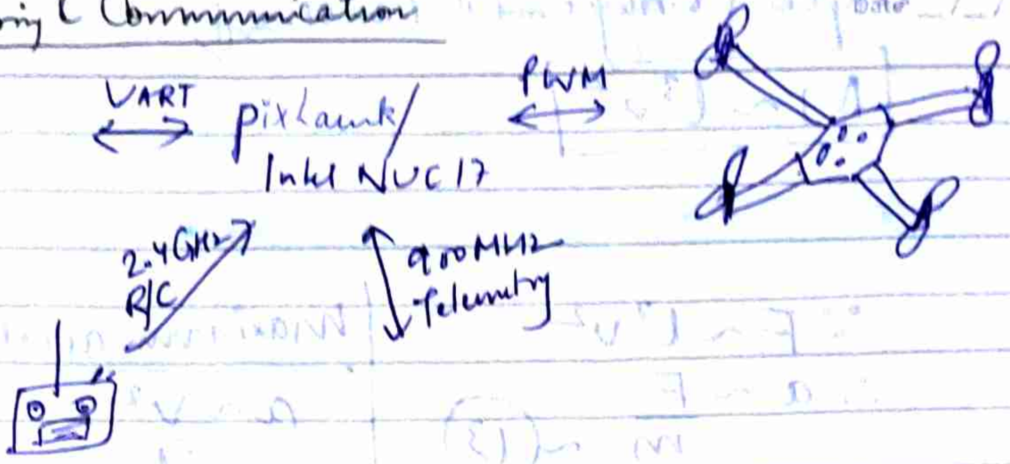


→ As the robot travels with a larger velocity, the stopping distance ↑

→ Also, the higher the ability of the robot to accelerate or decelerate, the smaller the stopping distance.

Component Selection

Processing & Communication



Laser scanner / Cameras

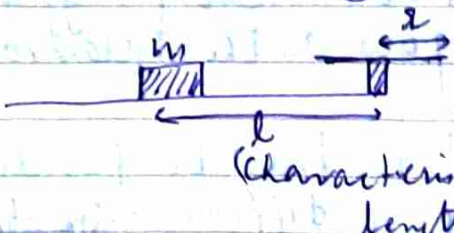
→ Longer the range, longer the stopping distance can be.
we can detect obstacles far away and so we get more time to come to a stop if we see an obstacle. This in turn allows us to go at a higher speed.

→ Also, Longer the range, the heavier our sensor might be & in turn ↑ the weight of the platform → decreases the thrust/weight ratio.

Effect of sizing the platform

Larger platform → bigger & heavier → thrust to weight ratio changes
Smaller platform → thrust to weight ratio might get better.

Agility with Scaling



✓ mass, inertia

$m \sim l^3$, $I \sim l^5$ (Moment of Inertia)

✓ thrust (total thrust applied by the rotor)

$F \sim \pi r^2 \times (\omega r)^2$, ω is rotor speed (rotor angular speed), r is rotor radius

$\sim r^2 v^2$ (bladetip speed) $\Rightarrow \boxed{F \sim l^2 v^2 \ (\because r \sim l)}$

✓ Moment (generated by the vehicle)

$M \sim F l$ (If we apply the thrust F on each motor, the moment that we can apply scales as force times length)

Assume that the rotor size scales as a characteristic length (Binaugeometric Constraint)

Date: __/__/__

$$\boxed{M \sim l^3 v^2}$$

Max^m acc^n & max^m angular acc^n

$\because F \sim l^2 v^2$

$\therefore a \sim \frac{F}{m} \sim l^3$

Also, $M \sim l^3 v^2$

$\alpha = \frac{M}{I} \sim l^5$

Maximum acceleration

$a \sim \frac{v^2}{l}$

$\alpha \sim \frac{v^2}{l^2}$

How does the blade tip speed, v scale as the characteristic stick length?

$\boxed{v \sim \sqrt{l}}$ characteristic length (Froude scaling)

blade tip speed

Froude scaling suggests that blade tip speed goes as the square root of length.

In contrast to Froude Scaling, in aerodynamics there is a different paradigm called Mach scaling. It suggests ~~that the blade tip speed goes as the square root of length~~ is roughly constant, independent of length.

$F \sim l^2 v^2 \sim l^2 l \sim l^3$

Froude scaling

$\boxed{v \sim \sqrt{l}}$

$F \sim l^3$

$a \sim 1, \alpha \sim 1/l$

Mach scaling

$\boxed{v \sim 1}$

$F = l^2$

$a \sim \frac{1}{l}, \alpha \sim 1/l^2$

$\alpha \uparrow$ as size of platform $\downarrow$

#learnthesmarterway

→ The smaller we make the vehicle, the larger the acceleration we can produce in the angular direction.

This allows greater agility.

Geometry and Mechanics

Transformation notation

$^{A}A_{B}$, $^{A}R_{B}$, $^{A}g_{B}$

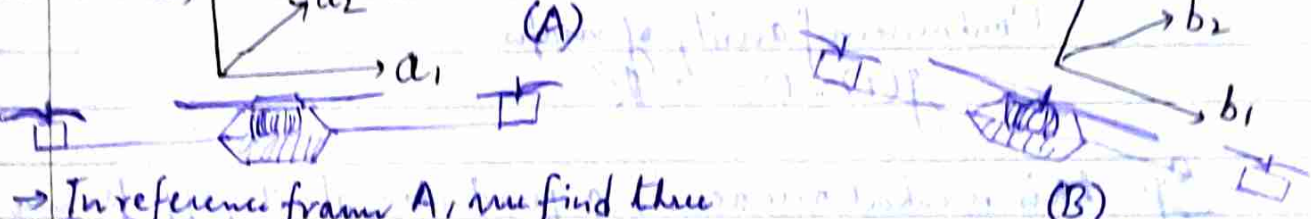
A_{ab}, R_{ab}

g_{ab}, h_{ab}

Quadrotor Kinematics

Rigid Body Transformations

Reference frame — we associate with each position and orientation a reference frame.



→ In reference frame A, we find three linearly independent basis vectors a_1, a_2 & a_3. Here, they are mutually $\perp^r$. While they don't have to be, it's convenient to choose them to be mutually ⊥ orthogonal.

→ b_1, b_2, b_3 is attached to frame B.

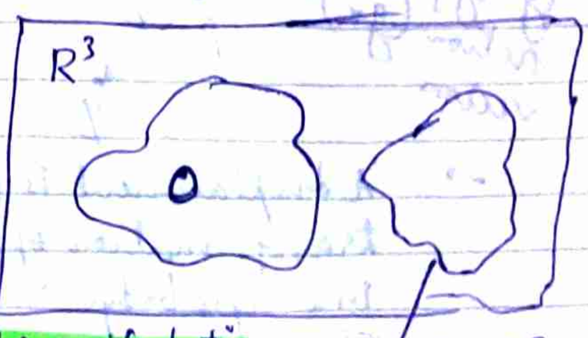
→ The key idea is that they must at least be linearly independent.

→ We can write any vector as a linear combination of the basis vectors in either frame

$V = V_1 a_1 + V_2 a_2 + V_3 a_3$ (in frame A) → Similarly for frame B

Rigid Body Displacement

Object $O \subset \mathbb{R}^3$ (subset of $\mathbb{R}^3$)



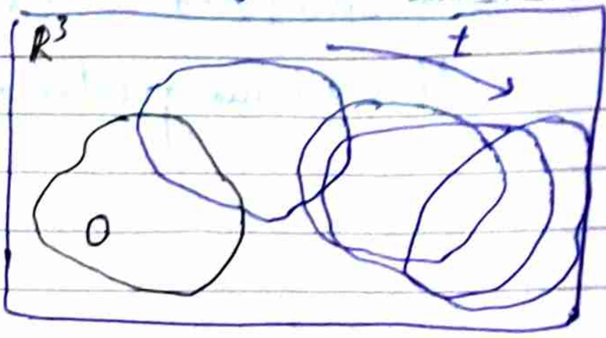
Same object O but with a diff^n pos & orient

→ A rigid body displacement is nothing but a map from this collection of points in the object to its physical manifestation in $\mathbb{R}^3$ (real space).

$g : O \longrightarrow \mathbb{R}^3$

→ As time changes, this object may occupy different positions and orientations.

→ Accordingly, we have different rigid body displacements.



What is rigid body motion?

→ As time ~~passes~~ changes, we have a continuous family of maps, so g (which is a displacement) is now parametrized by time.

→ As time changes, we have the collection of points in O, moving from one position & orientation to another & so on. This is a continuous set of displacements.

Continuous family of maps

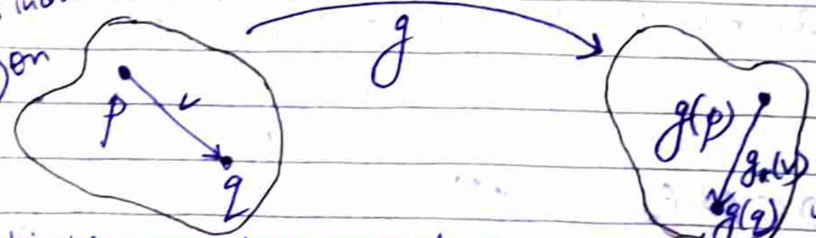
$$g(t): O \longrightarrow R^3$$

Eg. this is what we see in a quadrotor, starting off from a a horizontal position, moving to another position, accelerating, changing its orientation, changing the dir^n of thrust & then reversing the dir^n of the thrust by pitching back & then slowing down to the goal position.

Each of these snapshots is a displacement. This sequence of displacements represents a continuous family of displacements.

A single point on the rigid body p

Transformation (g) of points induces an action (g*) on vectors.



when this rigid body is displaced, the point p gets displaced to a new point, g(p)

→ A displacement is essentially a transformation of points. There are ∞ points in a rigid body.

→ If we have a second point q, the same displacement will take q & move it into a new point g(q).

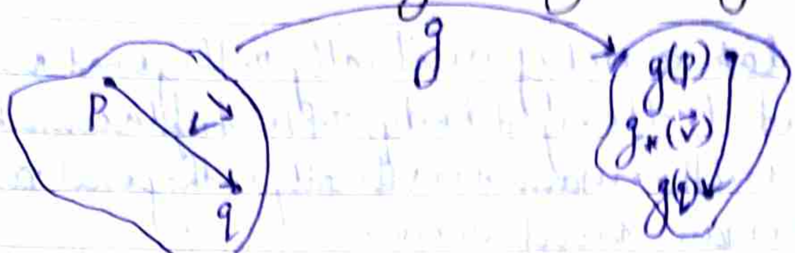
→ Every point defines a vector.

Since g moves p to $g(p)$ and $q \to g(q)$, it also moves the $\vec{v}$ to a new vector, $g_*(v)$.

So, displacement g induces a map on vectors

★ (displacement acts on points, but g_* acts on vectors)

What makes the map, g, a rigid body displacement?



Two properties the map must satisfy —

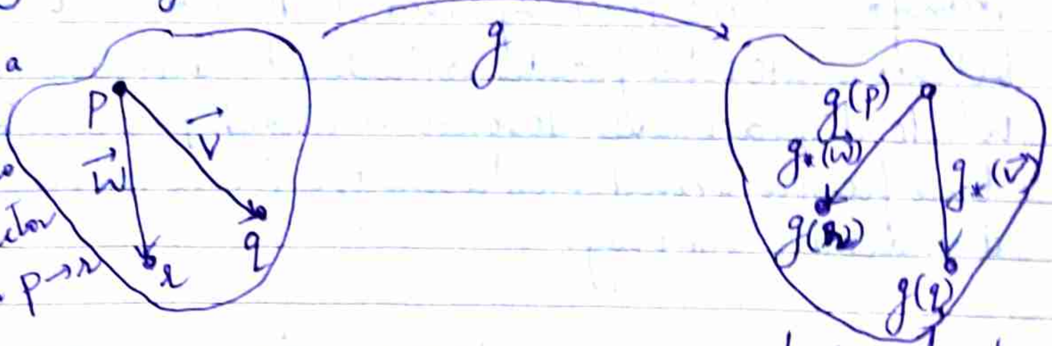
(1) the distance betn any pair of points remain unchanged in a rigid body displacement, i.e.,

(Lengths must be preserved) $||g(p) - g(q)|| = ||p-q||$ (defn of the word "rigid")

(after the displacement) (before the displacement)

(11) (Related to the cross products of vectors that are attached to the rigid body)

Let's choose a third point 'r' & generate a second vector going from $p \to r$



(Cross products are preserved) $g_*(\vec{v}) \times g_*(\vec{w}) = g_*(\vec{v} \times \vec{w})$

cross product remain the same whether we do it before the displacement or after the displacement, if the displacement is rigid.

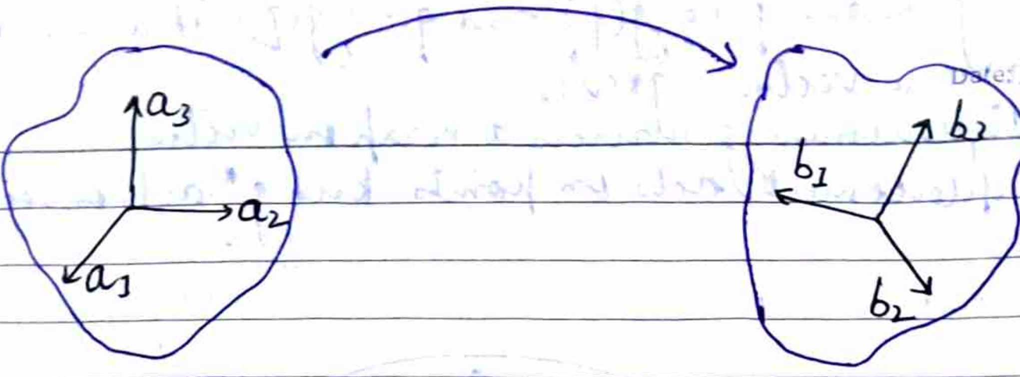
$\implies$ for a rigid body displacement,

→ orthogonal vectors are mapped to orthogonal vectors.

→ g_* preserves inner products

$$g_*(\vec{v}) \cdot g_*(\vec{w}) = \vec{v} \cdot \vec{w}$$

~~(before di~~ (After displacement) (before displacement)



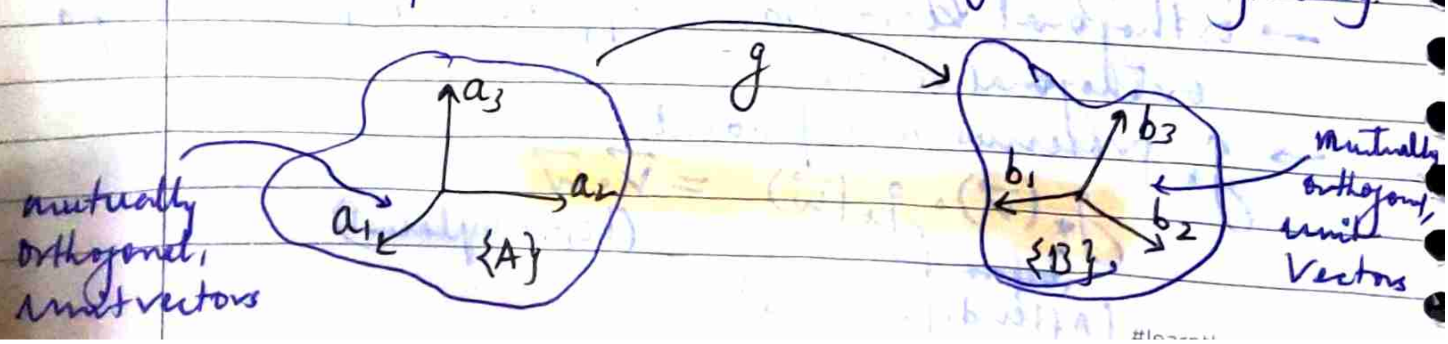
→ If we take a set of mutually orthogonal unit vectors attached to the rigid body, after displacement, those vectors will remain mutually orthogonal and they will remain unit vectors.

→ In summary, rigid body transformations or rigid body displacements satisfy two important properties. They preserve length and they preserve cross products.
(map preserves lengths) (cross products are preserved by the induced map)

Note: Rigid body displacements and rigid body transformation are used interchangeably. There is one important semantic difference.

Transformations generally used to describe relationship between reference frames attached to different rigid bodies while displacements describe relationships between two positions and orientation of a frame attached to a displaced rigid body.

Calculation :- (Assumption) we have mutually orthogonal unit vectors attached to every rigid body.
If it's a transformation, we are referring to two different rigid bodies. If it's a displacement, it's two distinct position and orientation of the same rigid body.



mutually orthogonal unit vectors in one frame as a linear Combination of mutually orthogonal unit vectors in the other frame

$$\hat{b}_1 = R_{11}\hat{a}_1 + R_{12}\hat{a}_2 + R_{13}\hat{a}_3$$
$$\hat{b}_2 = R_{21}\hat{a}_1 + R_{22}\hat{a}_2 + R_{23}\hat{a}_3$$
$$\hat{b}_3 = R_{31}\hat{a}_1 + R_{32}\hat{a}_2 + R_{33}\hat{a}_3$$

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

(rotation matrix)

properties of a Rotation Matrix

(I) Orthogonal $\Rightarrow RR' = I = R'R$

(II) Special Orthogonal $\Rightarrow |R| = +1$

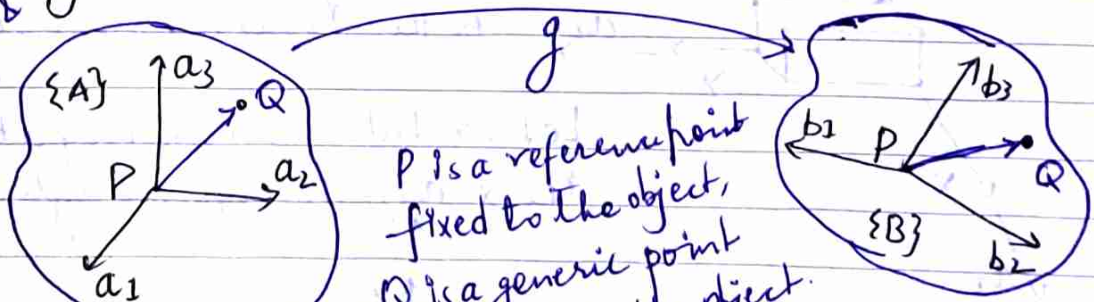
(III) Closed under multiplication

→ the product of any two rotation matrices is another rotation matrix.

(IV) The inverse of a rotation matrix is also a rotation matrix.

Structure of a Rotation Matrix

two distinct posi & orient. of the same rigid body

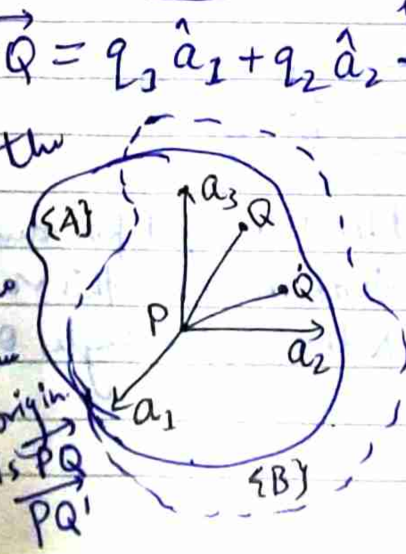


P is a reference point fixed to the object, Q is a generic point attached to the object.

Linearly Combination

$$\vec{PQ} = q_1\hat{a}_1 + q_2\hat{a}_2 + q_3\hat{a}_3 \qquad \vec{PQ} = q_1\hat{b}_1 + q_2\hat{b}_2 + q_3\hat{b}_3$$

translate the rigid body so that the two vectors have the same origin. one vector is $\vec{PQ}$ other is $\vec{PQ'}$



$$\vec{PQ} = q_1\hat{a}_1 + q_2\hat{a}_2 + q_3\hat{a}_3$$
$$\vec{PQ'} = q_1'\hat{a}_1 + q_2'\hat{a}_2 + q_3'\hat{a}_3$$

$$\begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

How to write (q_1', q_2', q_3') as a function of (q_1, q_2, q_3)

OR
$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix}$$

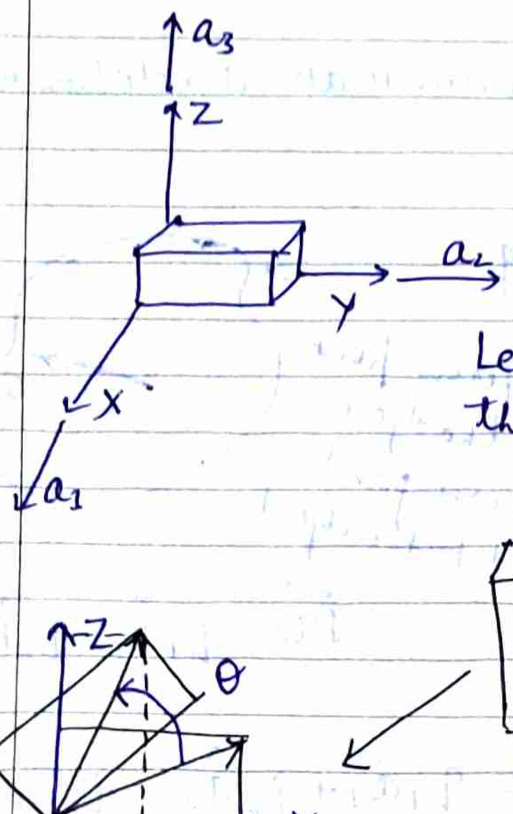
$\Downarrow$

Rotation matrix.

i.e.,
$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix}$$
(Rotation matrix)

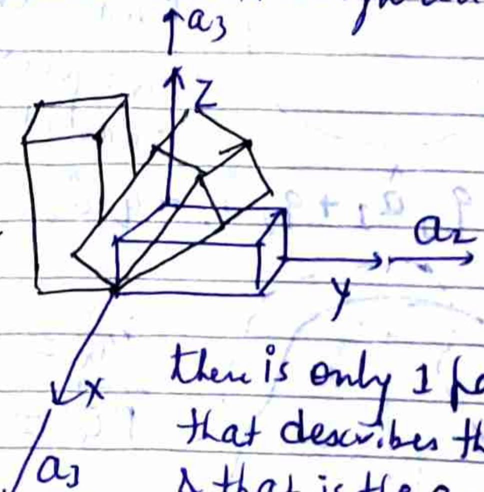
→ If we know how to write mutually orthogonal unit ~~...~~ vectors in one frame as a function of the other set of vectors (we can do this by calculating Rotation matrix). The same matrix tells us how to transform vectors in one frame to another frame.

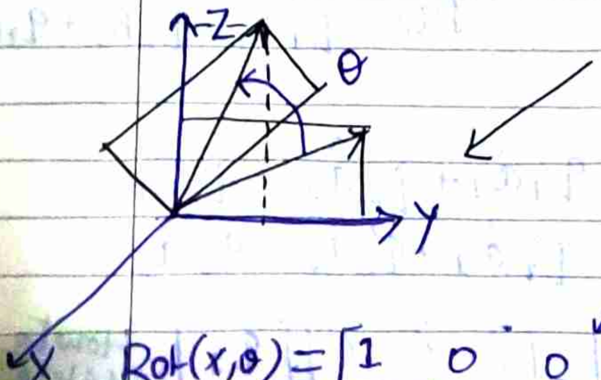
eg. Rotation – Rotation about the x-axis through θ.



Consider this rigid body, a rectangular ~~frame~~ prism, whose axes are aligned with X, Y & Z-axis or the a_1, a_2, a_3 unit vectors.

Let's rotate this rigid body about the x-axis, through an angle, θ.



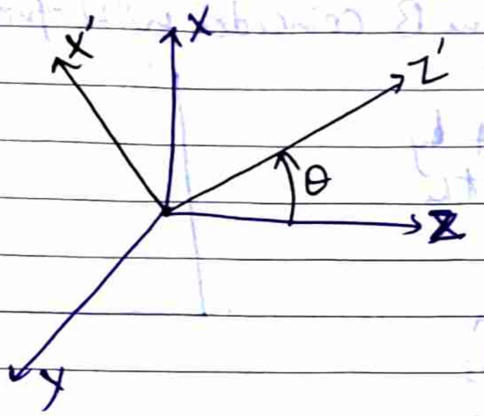


there is only 1 parameter that describes this rotation & that is the angle, θ.

$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

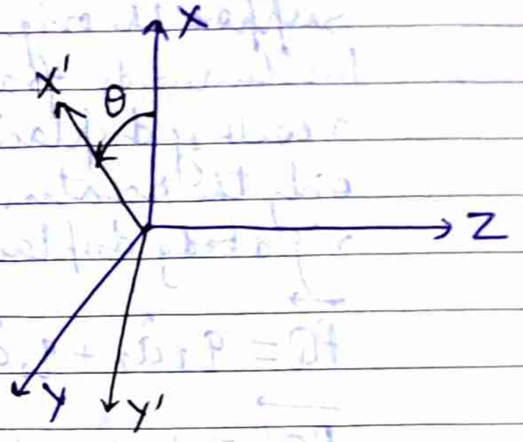
Rotation about the y-axis through θ

$$Rot(y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

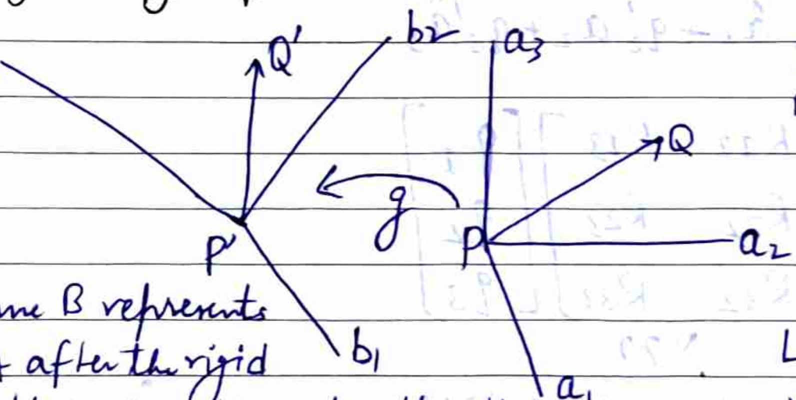


Rot. about Z-axis through θ

$$Rot(Z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Ex- Rigid Body Displacement



Let Frame A with mutually orthogonal unit vectors $\hat{a}_1, \hat{a}_2, \hat{a}_3$ with origin P attached to a rigid body.

Let Q be a point on this rigid body.

$\vec{PQ}$ → represents position of point Q relative to frame A.

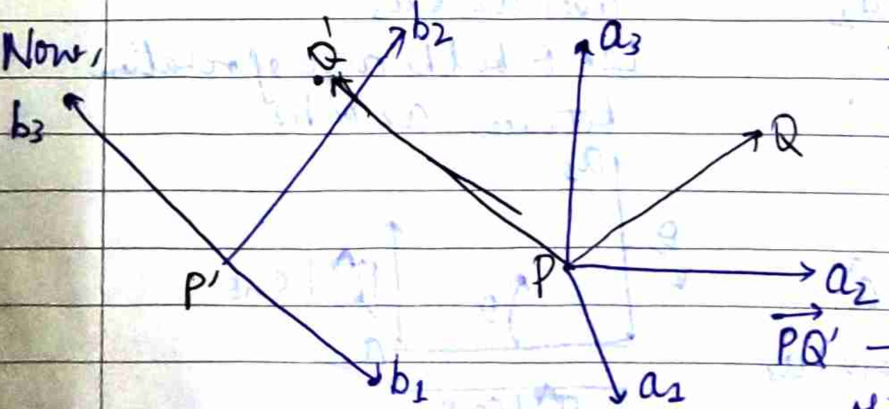
Suppose the body undergoes a rigid body displacement 'g'.

$$\vec{PQ} = q_1\hat{a}_1 + q_2\hat{a}_2 + q_3\hat{a}_3$$

Let Frame B represents Frame A after the rigid body displacement with mutually orthogonal unit vectors $\hat{b}_1, \hat{b}_2$ & $\hat{b}_3$ with origin at P'. point Q after the rigid body displacement is Q'.

$$\vec{P'Q'} = q_1\hat{b}_1 + q_2\hat{b}_2 + q_3\hat{b}_3$$

Now,



$\vec{PQ'}$ → gives us the position of the displaced point Q' relative to frame A.

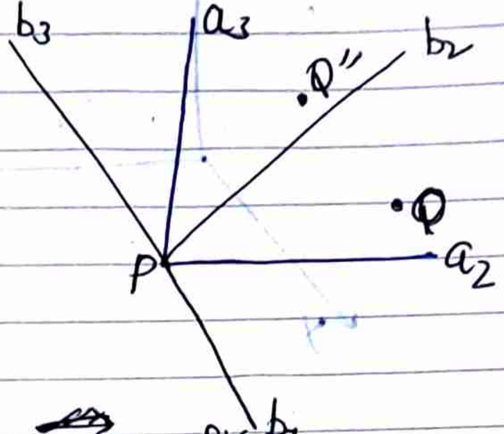
$$\begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix} = R \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} + d$$

Date: __/__/__

<u>Rotation between frame A and frame B</u>

Translate {B} so reference frame share an origin.

Suppose the origin of frame B coincides with frame A. In other words, frame B is a result of displacing frame A by only the rotation portion of the rigid body displacement.



$$\overrightarrow{PQ} = q_1\hat{a}_1 + q_2\hat{a}_2 + q_3\hat{a}_3$$

$$\overrightarrow{PQ''} = q_1\hat{b}_1 + q_2\hat{b}_2 + q_3\hat{b}_3$$
$$= q_1''\hat{a}_1 + q_2''\hat{a}_2 + q_3''\hat{a}_3$$

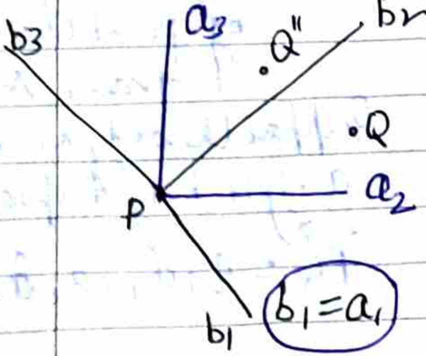
$\overrightarrow{PQ}$ is the p.v. of Q in {A}

PQ'' is the position vector of Q after the rotation in frame B / {B}

$$\begin{bmatrix} q_1'' \\ q_2'' \\ q_3'' \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

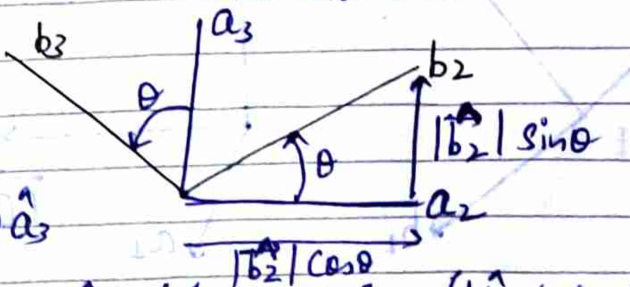
"??

we can easily see that, in this example, $\hat{a}_1$ & $\hat{b}_1$ are the same.



⇒ This means axes b_2 & b_3 are in the same plane as the axes a_2 & a_3.

Let θ be the angle of rotation between a_2 & b_2.



$$\hat{b}_3 = -(|\hat{b}_3|\sin\theta)\hat{a}_2 + (|\hat{b}_3|\cos\theta)\hat{a}_3$$
$$= -(\sin\theta)\hat{a}_2 + (\cos\theta)\hat{a}_3$$

$$\hat{b}_2 = (|\hat{b}_2|\cos\theta)\hat{a}_2 + (|\hat{b}_2|\sin\theta)\hat{a}_3$$
$$= (\cos\theta)\hat{a}_2 + (\sin\theta)\hat{a}_3$$

$|\hat{b}_1| = |\hat{b}_2| = |\hat{b}_3| = 1$ (unit vectors)

$$\vec{PQ}'' = q_1 \hat{b}_1 + q_2 \hat{b}_2 + q_3 \hat{b}_3$$
$$= q_1'' \hat{a}_1 + q_2'' \hat{a}_2 + q_3'' \hat{a}_3$$

$$\vec{PQ}'' = q_1(\hat{a}_1) + q_2\left[\hat{a}_2 \cos\theta + a_3 \sin\theta\right] + q_3\left[-a_2 \sin\theta + a_3 \cos\theta\right]$$

$$= q_1 \hat{a}_1 + (q_2 \cos\theta - q_3 \sin\theta)\hat{a}_2 + (q_2 \sin\theta + q_3 \cos\theta)\hat{a}_3$$

$$\vec{PQ}'' = q_1'' \hat{a}_1 + q_2'' \hat{a}_2 + q_3'' \hat{a}_3$$
$$\vec{PQ}'' = q_1 \hat{a}_1 + (q_2 \cos\theta - q_3 \sin\theta)\hat{a}_2 + (q_2 \sin\theta + q_3 \cos\theta)\hat{a}_3$$

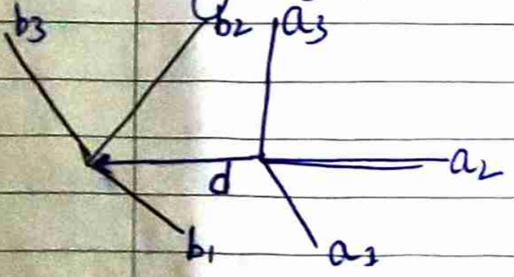
$$\therefore q_1'' = q_1$$
$$q_2'' = q_2 \cos\theta - q_3 \sin\theta$$
$$q_3'' = q_2 \sin\theta + q_3 \cos\theta$$

So,
$$\begin{bmatrix} q_1'' \\ q_2'' \\ q_3'' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$\therefore R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} = Rot(x, \theta)$$

Translation between frame A and frame B

Let d be the position vector from the origin of frame A to the origin of frame B, expressed in terms of frame A.



e.g. $\vec{d} = 1\hat{a}_1 - 3\hat{a}_2 + 1\hat{a}_3$

$$= \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}$$

→ we can characterize a rigid-body displacement with a rotation matrix and translation vector.

Let $\theta = \pi/4 \Rightarrow R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

$$\begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}$$

Suppose we know that the position vector $\vec{PQ}$ is given by component 1, 2, 1.

Components of $\vec{PQ'}$ in frame A/{A}

$$= \begin{bmatrix} 2 \\ -2.29 \\ 3.12 \end{bmatrix}$$

Qualitatively, we can see that point Q' relative to ~~point~~ {A} is in the positive $\hat{a}_1$, negative $\hat{a}_2$ & positive $\hat{a}_3$ direction.

→ If we know the components of $\vec{PQ'}$, we can invert the previous equation to find the components of $\vec{PQ}$.

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = R^T\left(\begin{bmatrix} q_1' \\ q_2' \\ q_3' \end{bmatrix} - \vec{d}\right)$$

→ The smaller we make the vehicle, the larger the acceleration we can produce in the angular direction

This allows greater agility.