19/5/22

Wants

→ slow rpm, 3 phase, bldc motor.

Have

→ high rpm, bemf sensing, positing arduino, performs well.

Dif

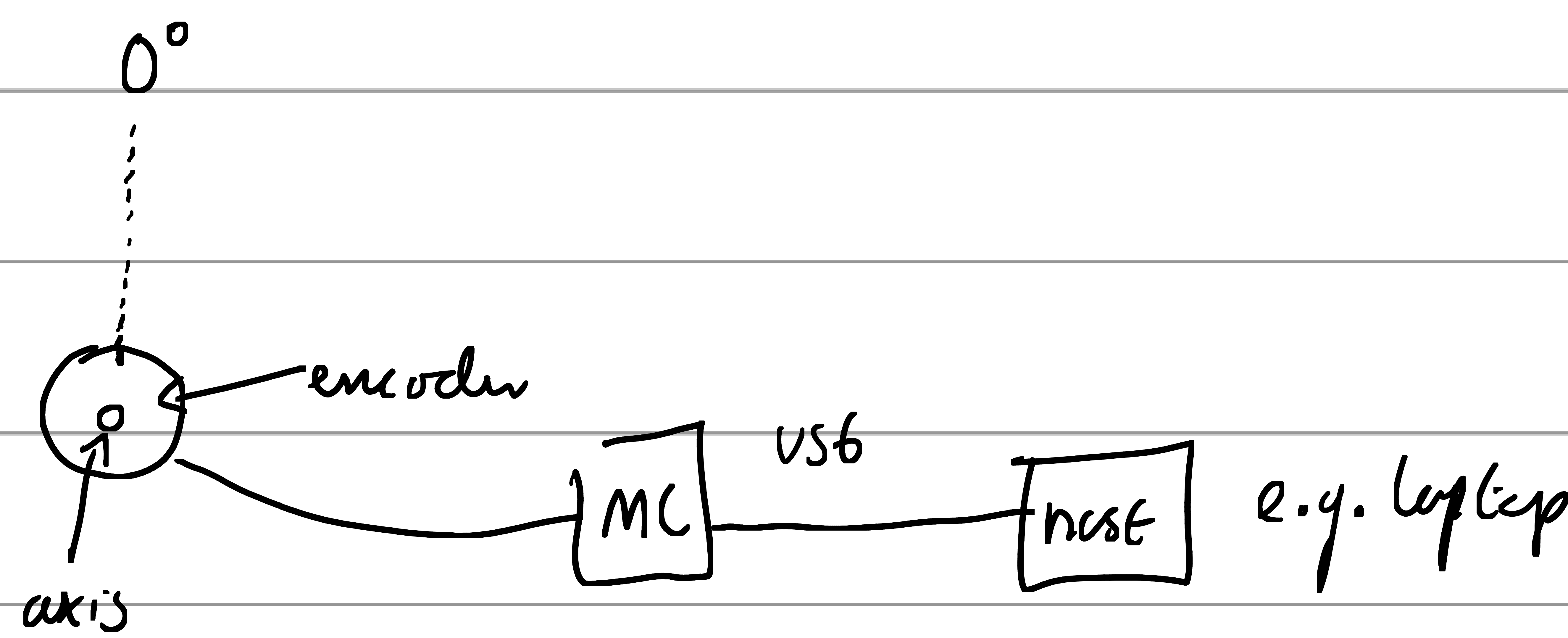
→ assume with low rpm that BEMF will be too noise.

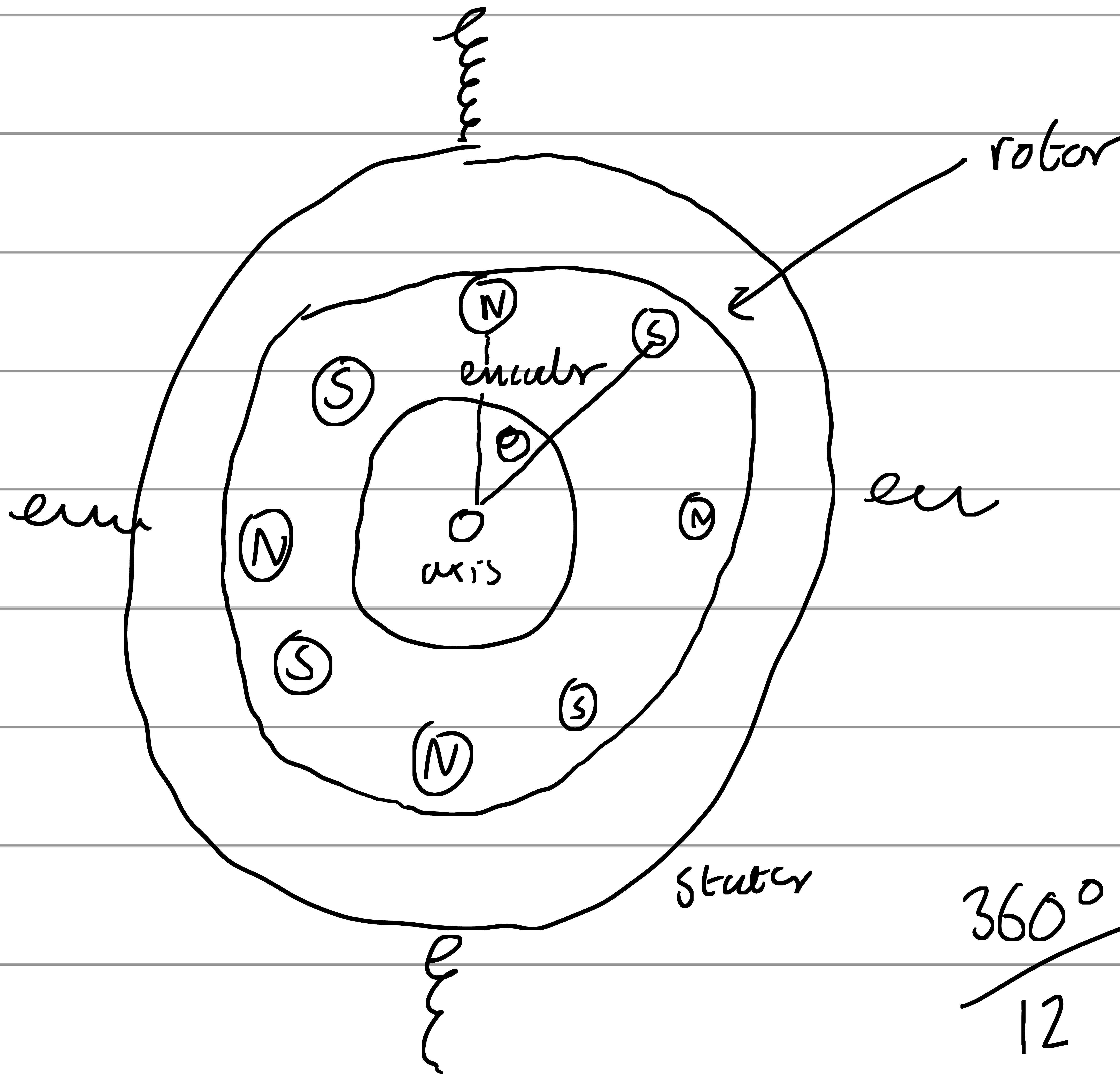
→ Arduino works bemf completely
→ Teensy 4.0 work bemf marginal, 100/255 delays

What we should consider.

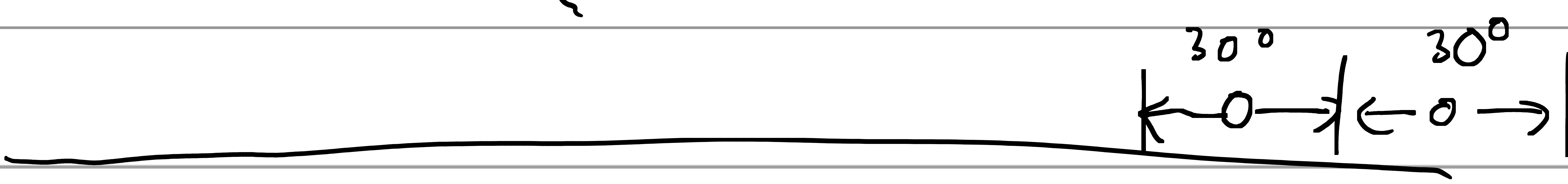
→ BEMF strategy will not work for low rpm, therefore we will not know position information reliably.

→ Rotary encoder (absolute)

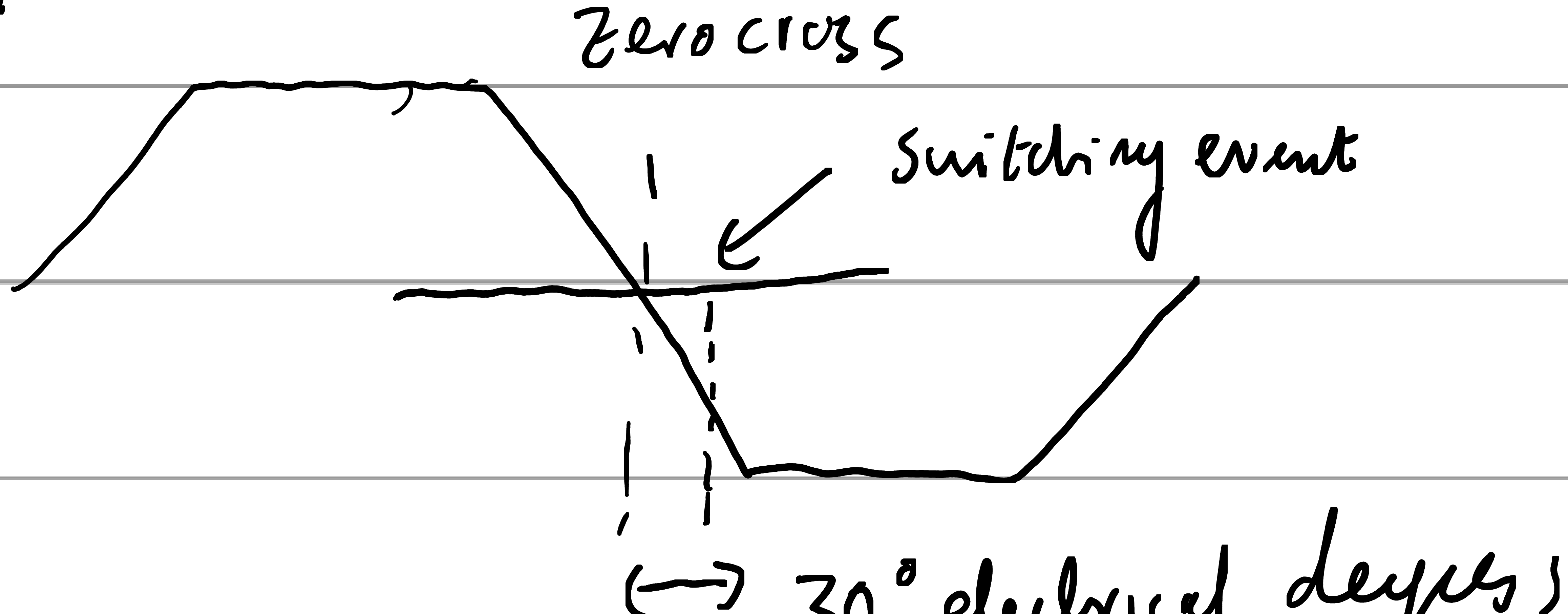




$$\frac{360^\circ}{12} = 36^\circ$$



Phase x:



BldC 6 electrical steps per electrical revolution
& you will have n - electrical cycles per mechanical cycle.

Mechanical steps per mechanical cycle $= \frac{N \text{ poles}}{2} \cdot 6$

4 poles, 3 phase $q=5$ → 12 slots

12 poles 3 phase → 36 slots

multiple of 3 slots

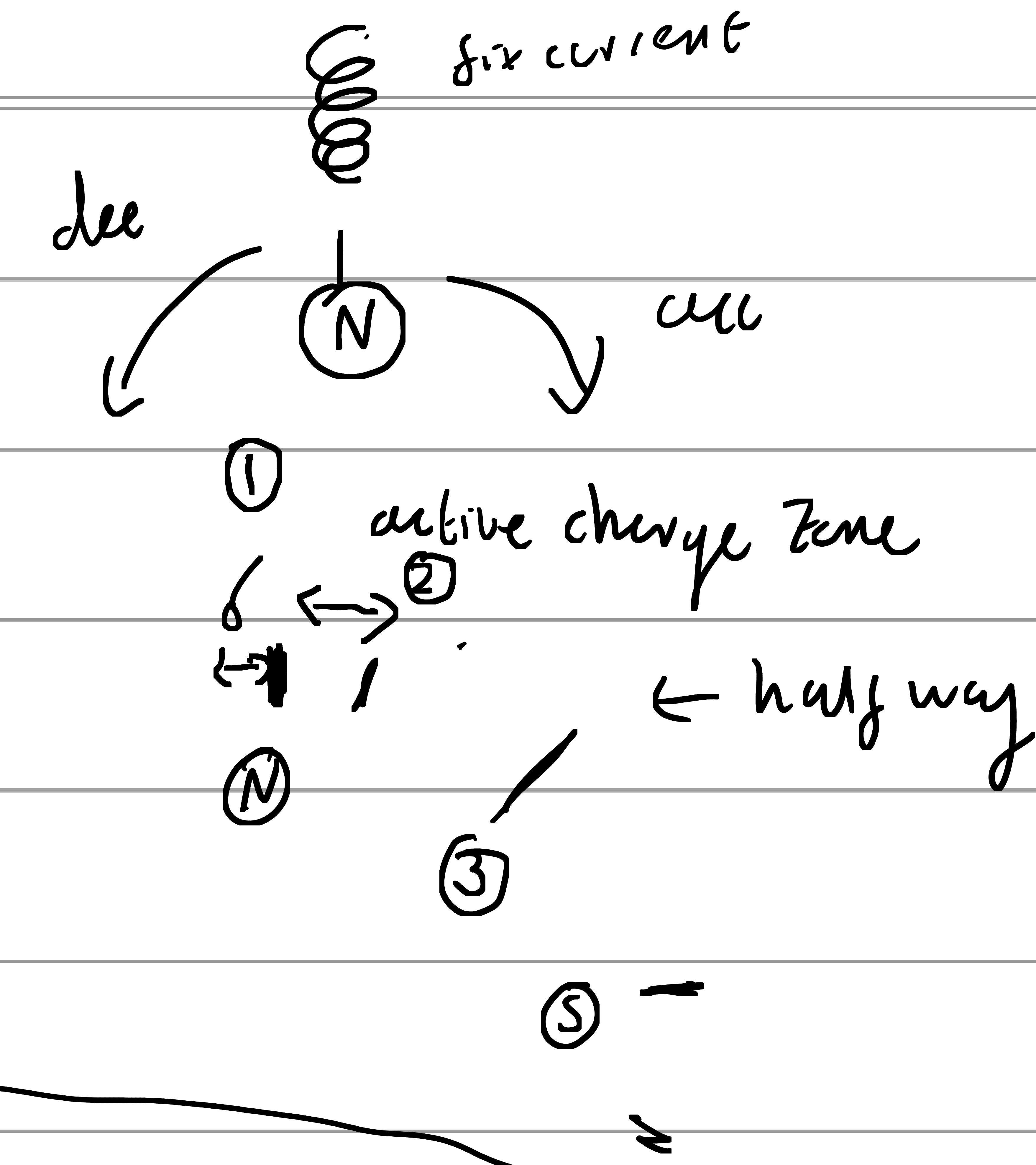
1) Slots = poles · phase

2) slots = stator count.

$= \frac{12}{2} \cdot 6 = 36$

$\boxed{6} \cdot \boxed{6}$

9 coils 12 poles



step 1 charge coils, Step 2 discharge coils, step 3 charge commutation step $A \to B$

proposing that we charge the relevant coil when the encoder has indicated precession beyond the central point of the pole (after max flux), then we have an interrupt each time the encoder senses a rotational displacement change, such that we disable the coil charge after some time well before we enter the next zone

swap phase + charge coil, wait for displacement discharge coil. [for each mechanical step.]

Duty cycle resolution, (impulse of a single pulse of a single electrical / mechanical step)

revert

Arduino $\boxed{256}$

Tensy 40

256, 512, 1024, ~~8~~ 2048

4096

8192

problems

switching	driving
eny	H bridge ⊗

Positional encoder removes this problem

might still be an issue

$360^\circ \rightarrow$ 36 steps

10° per pr step.

mechnical

10 samples per step — 1° angular resolution

maybe we want 100 sampls per step 0.1°

what we want from an rotary encoder

→ absolute position

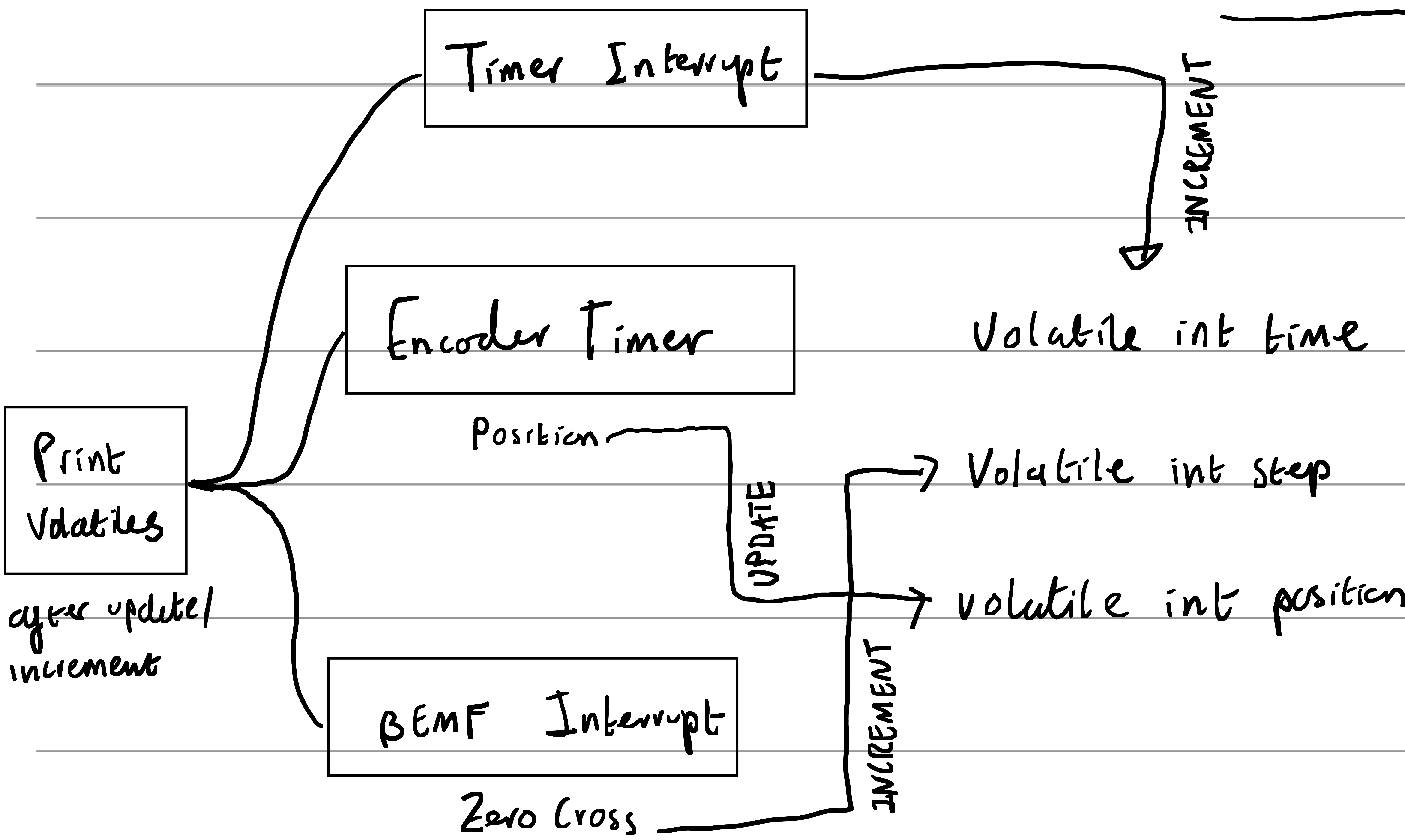
→ radial resolution $(0.1 \rightarrow 1^\circ)$

Nice to have

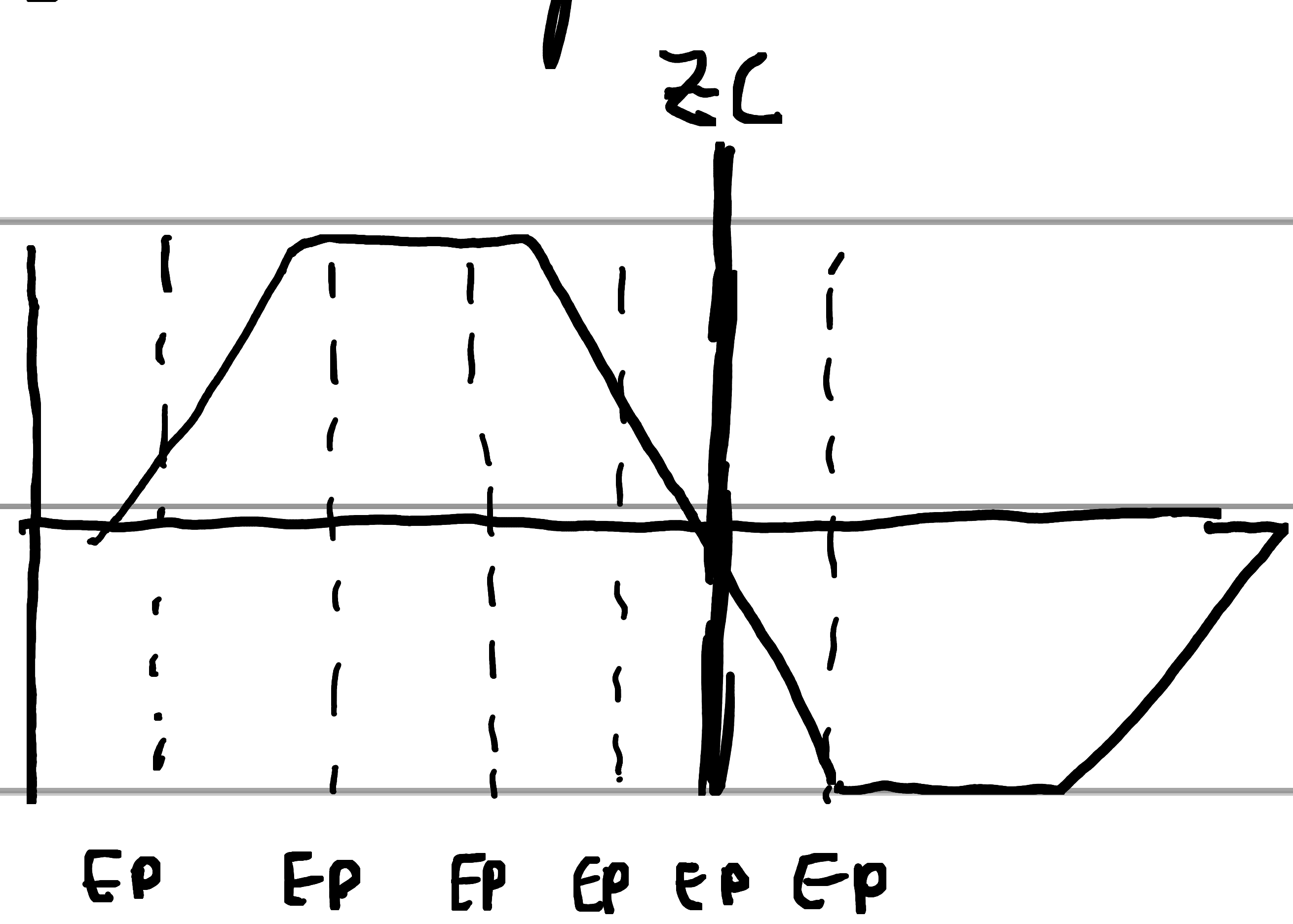
optical (no drag), no impurce the bearing, arduino + teensy40 support.

CALIBRATION

18/5/22



The the idea is to drive the assembly to a minimal speed such that we get clear bemf signals while all the time logging the bemf step, the rotational position via encoder values, and maybe the time.



EP - encoder pulse

ZC - Zero cross

We can bin the data via the encoder value the position

I therefore get an average I error for the Zero-cross angular positions. Ideal switching points are 30° (electrical) after the ZC positions.

One could use the ADC to confirm this.

The way this could work:

PWM Pulse @ 30 khz

↳ XBAR

↓

ADC Measure

↓

update measure

Timer Pulse

↳ Increment time

↳ Get encoder position

↑ up to 10 Mhz

21/8/22

Chosen the ADC method as the primary method.

Found 2 good magnetic encoders: Both 3.3 → 5v

→ AS5600 12-bit I^2C Not for high RPM

↓

4096

→ AS5147P 14-bit SPI 28krpm

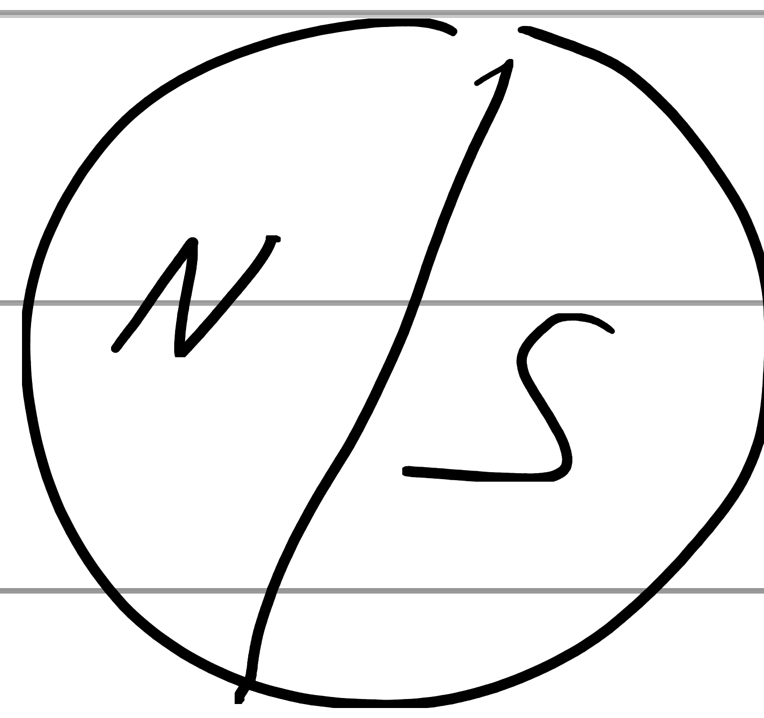
↓ 16,384

↑ This one is dedicated to motor control applications like bldc.

AS5147p

With 16,384 angular steps, 9 coils 12 poles thus 36 steps.

$\frac{16,384}{36}$ = 455.11 samples per zone.



2^{12} 4096

$\frac{360}{4096}$

$\boxed{0.088^\circ}$

$0.2 \rightarrow 0.7$

optical

Jerk model

We will be making position & time measurements

$P_i = [S_i, T_i]$ S - displacement angular

Euler method T - time interval

$$\omega_{i+1} = \frac{S_{i+1} - S_i}{T_{i+1} - T_i} \qquad W_{i+2} = \frac{S_{i+2} - S_{i+1}}{T_{i+2} - T_{i+1}}$$

$$\alpha_{i+2} = \dot{\omega}_{i+2} = \frac{W_{i+2} - W_{i+1}}{t_{i+2} - t_{i+1}}$$

$$= \frac{1}{T_{i+2} - T_{i+1}} \left(\frac{S_{i+2} - S_{i+1}}{T_{i+2} - T_{i+1}} - \frac{S_{i+1} - S_i}{T_{i+1} - T_i} \right)$$

$$= \frac{1}{a-b} \left(\frac{c-d}{a-b} - \frac{d-e}{b-f} \right)$$

$$\Rightarrow \frac{c-d}{(a-b)^2} \quad \frac{d-e}{(a-b)(f-b)} = \frac{S_{i+2} - S_{i+1}}{(T_{i+2} - T_{i+1})^2} \quad \frac{S_{i+1} - S_i}{(T_{i+2} - T_{i+1})(T_i - T_{i+1})}$$

what are we measuring θ_i, T_i $\theta_i(t)$

$$\omega_i = \dot{\theta}_i = \frac{d\theta_i}{dT_i}$$ angular velocity

$$\alpha_i = \dot{\omega}_i = \frac{d\omega_i}{dT_i} = \frac{d^2\theta_i}{dt^2}$$ angular acceleration

$$\dot{j}_i = \dot{\alpha}_i = \frac{d\alpha_i}{dT_i} = \frac{d^3\theta_i}{dt^3}$$ angular jerk.

Jerk model $X_i = [\theta, \dot{\theta}, \ddot{\theta}, \dddot{\theta}]$ in 1D

White noise $w(t)$ applying to jerk

$$X = AX_i + Bw(t)$$ a is a constant

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ \ddot{\theta}\\ \dddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & -a\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ \ddot{\theta}\\ \dddot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 1\end{bmatrix} w(t)$$

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ \ddot{\theta}\\ \dddot{\theta}\end{bmatrix} = \begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&-a\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ \ddot{\theta}\\ \dddot{\theta}\end{bmatrix} + \begin{bmatrix}0\\0\\0\\1\end{bmatrix} w(t)$$

$$\frac{d\theta}{dt} = \dot{\theta} \qquad \frac{d\dot{\theta}}{dt} = \ddot{\theta} \qquad \frac{d\ddot{\theta}}{dt} = \dddot{\theta}$$

$$\frac{d\dddot{\theta}}{dt} = -a\dddot{\theta} + w(t)$$

Measurement vector $Z(k+1) = H X(k+1) + V(k+1)$

H is the system measurement or observation matrix &

V is the measurement noise vector.

1D tracking

$$X(k+1) = F(k+1, k) X(k) + u(k)$$

$$F(k+1, k) = e^{A (t_{k+1}, t_k)}$$

matrix

$u(k)$ is the discrete white noise

$$F(t) = \begin{bmatrix} 1 & T & T^2/2 & p_1 \\ 0 & 1 & T & q_1 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & s_1 \end{bmatrix}$$

$$T = t_{k+1} - t_k$$

$$p_1 = (2 - 2aT + a^2T^2 - 2e^{-aT})/2a^3$$

$$q_1 = (e^{-at} - 1 + at)/a^2$$

$$r_1 = (1 - e^{-at})/a$$

$$s_1 \sim e^{-at} = 1 - at + \frac{a^2T^2}{2!} - \frac{a^3T^3}{3!} \cdots$$

where aT is small

$$F(T) \lim_{a \to 0} = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kalman algorithm

Prediction

1) Predict next frame

$$X_t = f(X_{t-1}, u_t) + q_1$$

2) Calculate prediction covariance matrix

$$Z_t = h(X_t, u_t) + q_2$$

Correction

3) calculate EKF gain

$$K_t = P_{t|t-1} H_t^T [H_T P_{t|t-1} \cdot H_t^T + R_t]^{-1}$$

4) Update the observation estimated value

$$X_t = X_{t|t-1} + K_t [Z_t - Z_{t|t-1}]$$

5) update error in covariance matrix

$$P_{t|t} = [Z_t - Z_{t|t-1} - K_t H_t] P_{t|t-1}$$

✓ ✓ ✓ ? Q? ✓ ✓

We need γ, h, Z, K_t, X_t, H, P

Initalisation, 3 measurements to estimate $\hat{\theta}$, $\hat{\dot{\theta}}$ & $\hat{\ddot{\theta}}$

$\hat{\dddot{\theta}}$ taken to be zero initally.

$$\hat{\theta}_3 = M_x(3) \qquad \hat{\dot{\theta}}_3 = \frac{M_x(3) - M_x(2)}{T}$$

$$\hat{\ddot{\theta}}_3 = \left[\frac{M_x(3) - 2M_x(2) + M_x(1)}{T^2}\right] \qquad \hat{\dddot{\theta}}_3 = 0$$

constant time interval T

Measurement noise

$M_x(1) = x(1) + v(1)$

$M_x(2) = x(2) + v(2)$

$M_x(3) = x(3) + v(3)$

Error in $\theta, \dot{\theta}, \ddot{\theta}, \dddot{\theta}$ $\quad \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$

$\varepsilon_1(3|3) = x(3) - \hat{x}(3|3) = x(3) - x(3) - v(3)$

$= -v(3)$

$\varepsilon_2(3|3) = \dot{x}(3) - \hat{x}(3|3) =$

Equations

$$X_j(k+1) = F_j X_j(k) + W_j(k)$$

$$Z_j(k+1) = H_j X_j(k+1) + V_j(k+1)$$

$$X_j = [\, x \ \dot{x} \ \ddot{x} \ \dddot{x} \,]^T$$

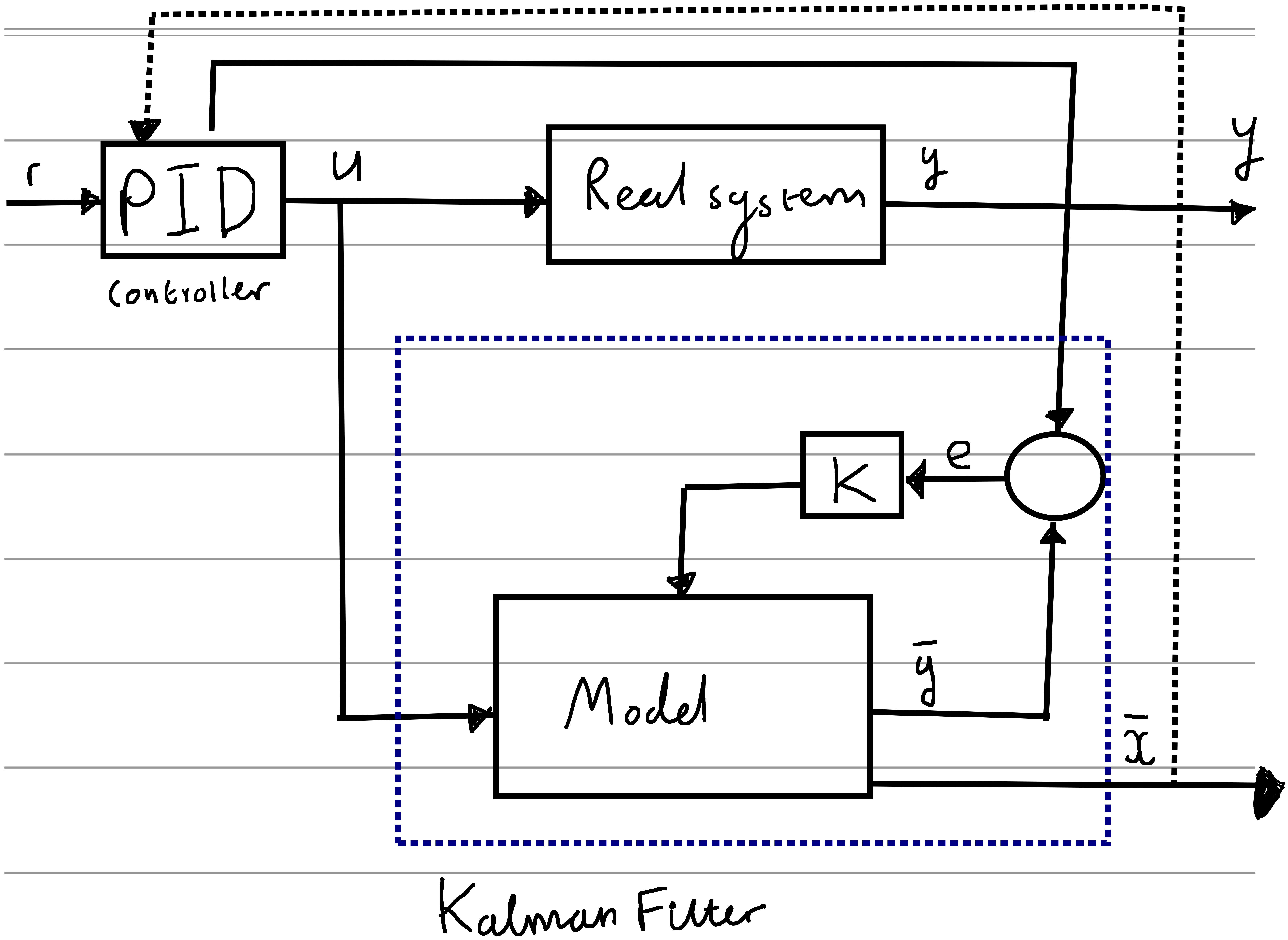
$$W_j = [\, u_{1j} \ u_{2j} \ u_{3j} \ u_{4j} \,]^T$$

$$H_j = [\, 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \,]$$

$$Z_j = [\, M_x \qquad]^T$$

$$V_j = [\, \text{noise } M_x \,]^T$$

$$F_j = \begin{bmatrix} 1 & T & T^2/2 & p_1 \\ 0 & 1 & T & q_1 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & s_1 \end{bmatrix}$$


r
PID
Controller
u
Real system
y
y
K
e
Model
$\bar{y}$
$\bar{x}$
Kalman Filter

Measurement errors

We are measuring angle θ in $2^{14} = 16,384$

θ degrees $0 \to 360$

radians $0 \to 2\pi$

we have angular resolution of $360/16,384 = 0.02^\circ$

so we get $\theta \pm 0.01^\circ$ $(\theta \pm \Delta\theta)$

$$\theta = (\underbrace{c}_{\text{encoder value}} \cdot 0.002^\circ) \pm 0.01^\circ$$

?

We are measuring time as ticks as an int32 with 31250 Hz

$$\frac{1}{31250\ \text{Hz}} = 3.2\times10^{-5}\,[s] = 32\times10^{-6}\,[s] = 32\,\mu s$$

so we get $t = (\underbrace{k}_{\text{ticks}} \cdot 32\,\mu s) \pm 16\,\mu s$

$(t \pm \Delta t)$

With an overflow time of $2^{16} \times (32\times10^{-6}) = 137,439\,[s] \sim 38\,[hrs]$

Speed error $\omega_i = \frac{d\theta_i}{dt_i}$ $\quad \Delta\omega_i = \omega_i\left(\frac{\overbrace{\Delta\theta}^{\text{CONSTANT}}}{\theta_i} + \frac{\overbrace{\Delta t}^{\text{CONSTANT}}}{t_i}\right)$

CONSTANT

Speed error $\omega_i = \frac{d\theta_i}{dt_i}$ $\Delta \omega_i = \omega_i \left(\frac{\Delta \theta}{\theta_i} + \frac{\Delta t}{t_i} \right)$

In reality these are digital measurements

so we have $2^{14} = 16,384 \pm 0.5$ Encoder steps angle

$S_E \pm \Delta S_E$

$$\frac{\Delta S_E}{S_E} = \frac{\cancel{0.5}\,1}{16,384 \times 2} = 3.05 \times 10^{-5}$$

$$\text{or}_{\text{Max}} \quad \frac{0.5}{1} = \frac{1}{2}$$

we have 2^{32} time max @ 31 250 Hz ~~k~~ 32 μS

$\pm 16 \mu S$

$$\frac{\Delta S_T}{S_T} = \frac{0.5}{2^{32}} = 1.16 \times 10^{-10}$$

$$\text{or}_{\text{Max}} \quad \frac{16 \times 10^{-6}}{32 \times 10^{-6}} = 0.5$$

25/5/22

Kalman plan

Jerk model try low α model 3rd order.

Measurement function → calculate velocity → jerk

Test sets → pendulum θ, t

→ Double pendulum lower half θ, t

Jerk

$$\dot{j}(t) = -\alpha \underbrace{j(t)}_{\text{jerk}} + \underbrace{w(t)}_{\text{white noise}}$$

Jerk correlation

$$r_j(\tau) = \underbrace{\sigma_j^2}_{\text{variance of target jerk}} \exp(-\alpha|\tau|)$$ ← small time

variance of w(t)

$$r_w(\tau) = 2\alpha\sigma_j^2\,\delta(\tau) \qquad Q = 2\alpha\sigma_j^2$$

State

$$X_{i+1} = AX_i + Bw_i(t)$$

$$X(k+1) = F(k+1, k)\,X(k) + u(k)$$

Measurement

$$Z(k+1) = \underbrace{H}_{\text{system measurement matrix}}\,X(k+1) + \overbrace{V(k+1)}^{\text{Noise vector}}$$

$$F(T) = \begin{matrix} 1 & T & T^2/2 & p_1 \\ 0 & 1 & T & q_1 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & s_1 \end{matrix}$$

$$T = t_{k+1} - t_k$$

low αT

$$\begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$H_j = [1\ 0\ 0\ 0]$ $\rightarrow$ 12

$\uparrow$

selecting just θ

$S_j = 0.1$

$q = 2\alpha\sigma^2 \cdot \left[\frac{1}{252}\right.$

$\cancel{Q} = \frac{q}{7}$

$\frac{1}{252}$ $\frac{1}{36}$

7

$0.01 = 2.7 \cdot \alpha \cdot \sigma^2$

$\cancel{\frac{0.01}{2.7} = \alpha\sigma^2}$

Initialise 3 measurements & jerk set to zero.

$$\hat{\theta} = M_\theta(3)$$

$$T$$

$$\hat{\dot{\theta}} = (M_\theta(3) - M_\theta(2))\ (T_3 - T_2)$$

$$\hat{\ddot{\theta}} = (M_\theta(3) - 2M_\theta(2) + 2M_\theta(1)) \,/$$

$$T_2 -$$

$$\hat{\dddot{\theta}} = 0$$

$$\begin{array}{ccc} M_\theta(1) & M_\theta(2) & M_\theta(3) \\ T_1 & T_2 & T_3 \end{array}$$

$$\frac{M_\theta(2) - M_\theta(1)}{T_2 - T_1}$$ dif

$$M_\theta(3) - 2M_\theta(2) + M_\theta(1)$$

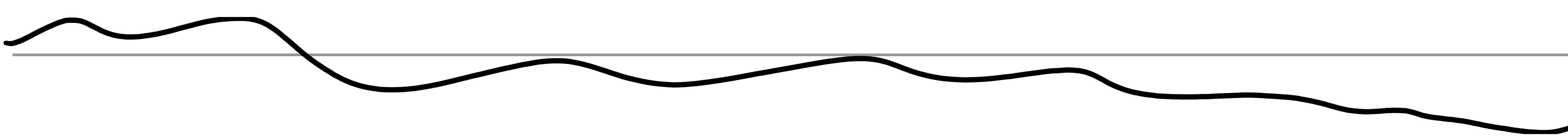
$$\frac{M_\theta(3) - M_\theta(2)}{T_3 - T_2}$$

$$\frac{M_\theta(3) - M_\theta(2) - (M_\theta(2) - M_\theta(1))}{T_3 - T_2 - (T_2 - T_1)}$$

timer

→ Time measurement + static position

Encoder → position + static time ↓ potentially no $dt = 0$



position $S(t)$ time a, b

speed = $\dfrac{\text{change pos}}{\text{change time}} = \dfrac{S(b) - S(a)}{b - a}$

position @ b $S(b)$

speed @ b = $\dfrac{S(b) - S(a)}{b - a}$

position @ b $S(b)$

speed @ b $= \dfrac{S(b)-S(a)}{b-a} = V(b)$

---

position $S(a)$ $S(b)$ $S(c)$ at time a, b, c

position @ c $= S(c)$

speed @ c $= V(c) = \dfrac{S(c)-S(b)}{c-\cancel{a}\,b}$

acceleration @ $a(c) = \dfrac{V(c)-V(b)}{c-b}$

$$\Rightarrow \frac{\dfrac{S(c)-S(b)}{c-a} - \dfrac{S(b)-S(a)}{b-a}}{c-b}$$

$$\frac{\dfrac{\overset{A}{S(c)} - \overset{B}{S(b)}}{c-a} - \dfrac{\overset{B}{S(b)} - \overset{C}{S(a)}}{b-a}}{c-b}$$ Simplify

$$\frac{1}{c-b}\;\frac{(b-a)(S(c)-S(b)) - (c-a)(S(b)-S(a))}{(c-a)(b-a)}$$

$$= \frac{1}{c-b}\;\frac{bS(c) - bS(b) - aS(c) + aS(b) - cS(b) + cS(a) + aS(b) - aS(a)}{(c-a)(b-a)}$$

$$= \frac{1}{c-b}\,\frac{1}{(c-a)(b-a)} \times \Big((b-a)S(c) + \overset{S(b)}{(-b+a-c+a)} + (c-a)S(a)\Big)$$

$$= \frac{1}{(c-b)(c-a)(b-a)} \times \Big(S(a)(c-a) + S(b)(2a-b-c) + S(c)(b-a)\Big)$$

$-A + 2B - C$ $\qquad$ $+\,b(S(c) - S(b))$

or $\;\ldots \times \; a\big(S(b) - S(c) + S(b) - S(a)\big) + c\big(S(a) - S(b)\big)$

Trusting wolfram

$$\frac{\dfrac{\overset{A}{S(c)} - \overset{B}{S(b)}}{c-a} - \dfrac{\overset{B}{S(b)} - \overset{C}{S(a)}}{b-a}}{c-b}$$

solution

$$\frac{a\,(S(c) - 2\,S(b) + S(a)) + b\,(S(b) - S(c)) + c\,(S(b) - S(a))}{(a-b)\,(a-c)\,(b-c)}$$

$$= a(c)$$

$\theta, \dot\theta, \ddot\theta$ at a, b, c

Final

$$\hat\theta(c) = \theta(c)$$

$$\hat{\dot\theta}(c) = w(c) = \frac{\theta(c) - \theta(b)}{c-b}$$

$$\hat{\ddot\theta}(c) = a(c) = \frac{a\,(\theta(a) - 2\theta(b) + \theta(c)) + b\,(\theta(b) - \theta(\cancel{a})\,c) + c\,(\theta(b) - \theta(a))}{(a-b)\,(a-c)\,(b-c)}$$

$$\hat{\dddot\theta}(c) \sim 0$$

$\theta, \dot{\theta}, \ddot{\theta}, \dddot{\theta}$ from time $a, \underbrace{b, c, d}_{\text{needed for } a(t)}$

$a \to b$
$b \to c$
$c \to d$

$$\hat{\theta}(d) = \theta(d)$$

$$\hat{\dot{\theta}}(d) = \frac{\theta(d) - \theta(c)}{d-c}$$

$$\hat{\ddot{\theta}}(d) = \frac{b(\theta(b) - 2\theta(c) + \theta(d)) + c(\theta(c) - \theta(b)) + d(\theta(c) - \theta(b))}{(b-c)(b-d)(c-d)} \quad *$$

$$\hat{\dddot{\theta}}(d) = \dot{j}(d) = \frac{a(d) - a(c)}{d-c} \quad \Delta$$

$$\hat{\ddot{\theta}}(c) = \overset{*}{a}(c) = \frac{a(\theta(a) - 2\theta(b) + \theta(c)) + b(\theta(b) - \theta(a)) + c(\theta(b) - \theta(a))}{(a-b)(a-c)(b-c)}$$

combine $*$'s with Δ expression.

$a = a$ $d = d$
$c = c$
$b = b$

Relable $A = \theta(a)$ $C = \theta(c)$
$B = \theta(b)$ $D = \theta(d)$

Wolfram

$$\hat{\dddot{\theta}} = \frac{1}{(d-c)(b-c)}\left(\frac{-a(A - 2B + C) + Ac - B(b+c) + bC}{(a-b)(a-c)} + \frac{-b(B - 2C + D) + Bd - C(c+d) + cD}{(b-d)(d-c)}\right)$$

Simplified:

$$\hat{\theta}(d) = \theta(d)$$

$$\hat{\dot{\theta}}(d) = \frac{\theta(d) - \theta(c)}{d-c} = \omega(d)$$

$$\hat{\ddot{\theta}}(d) = \frac{\omega(d) - \omega(c)}{d-c} = \alpha(d)$$

$$\hat{\dddot{\theta}}(d) = \frac{\alpha(d) - \alpha(c)}{d-c} = J(d)$$

Dependencies First Level

$\theta(d) \rightarrow$ None

$\omega(d) \rightarrow \theta(c)$, $\theta(d)$

$\alpha(d) \rightarrow \omega(c)$, $\omega(d)$

$J(d) \rightarrow \alpha(c)$, $\alpha(d)$

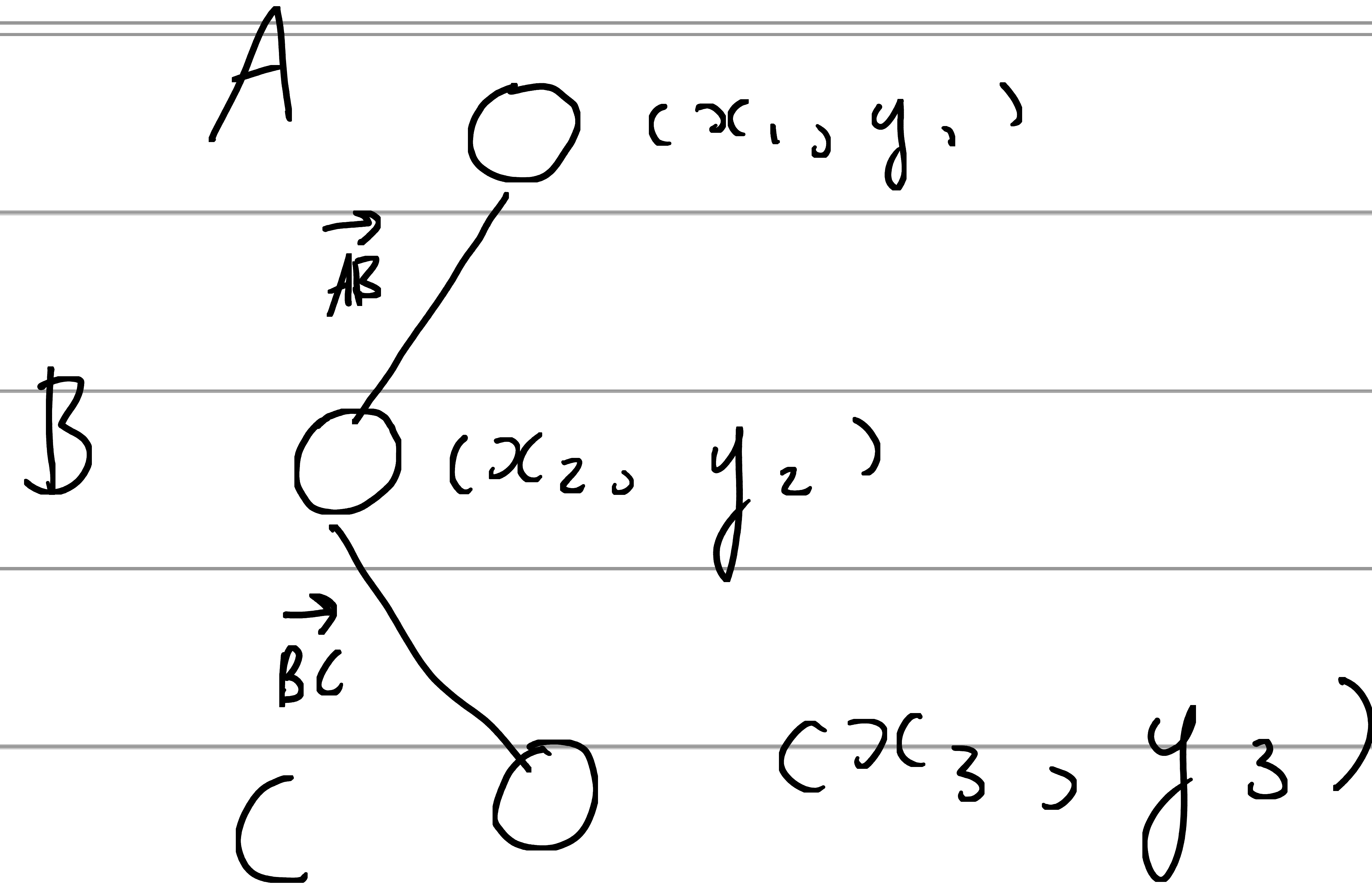
so we need $\alpha(c)$, $\alpha(d)$, d & c

$\omega(c)$, $\omega(d)$, d & c

$\theta(c)$, $\theta(d)$, d & c

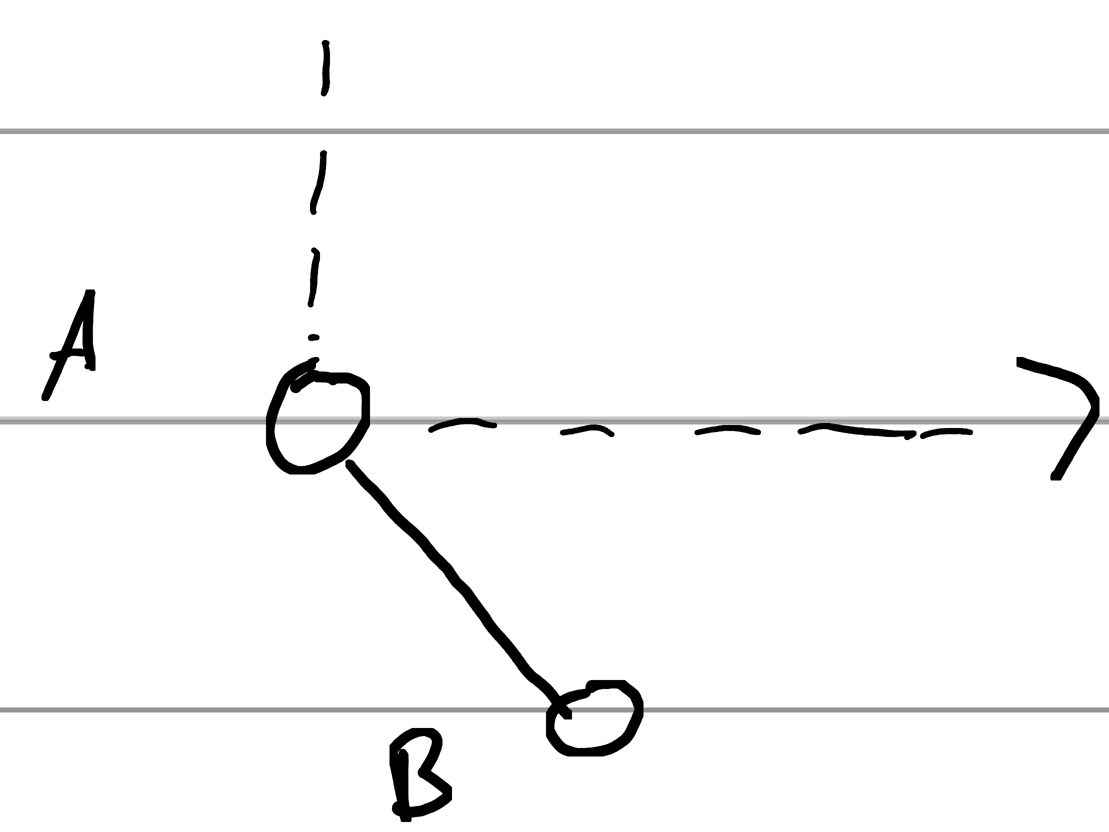
Incremental seems more sane as have to account

for $0 \to 360 \to 0$ discontinuity.



$$a \cdot b = |a||b|\cos\theta$$

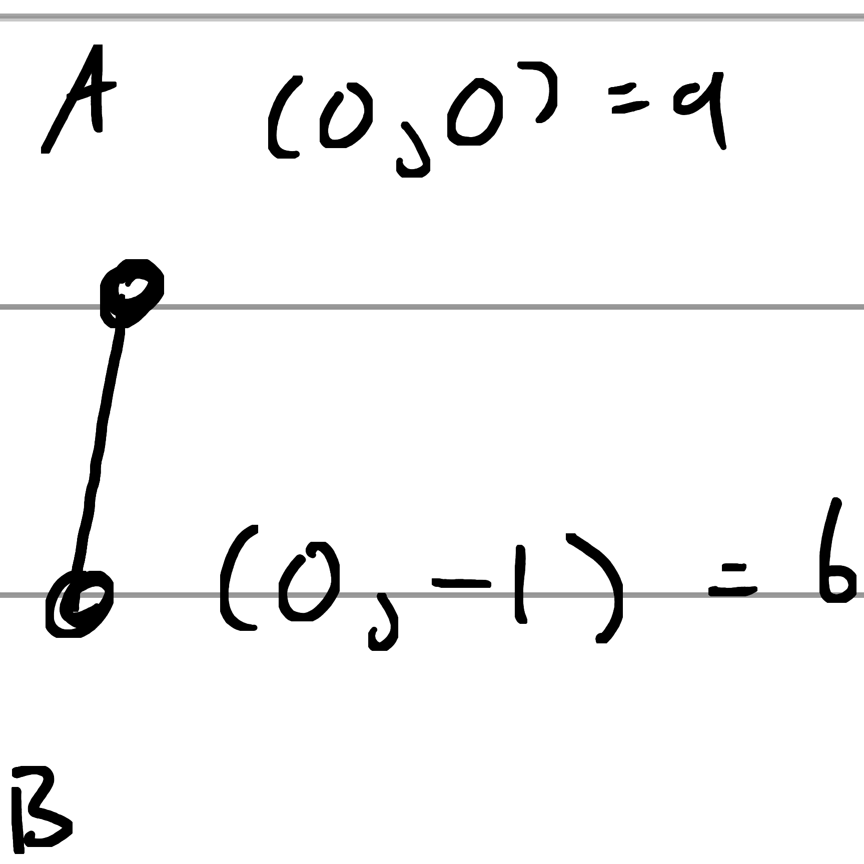
$\overrightarrow{AB}$



$$\frac{x}{y} = \frac{r\cos\alpha}{r\sin\alpha}$$

$$\frac{y}{x} = \tan\alpha$$

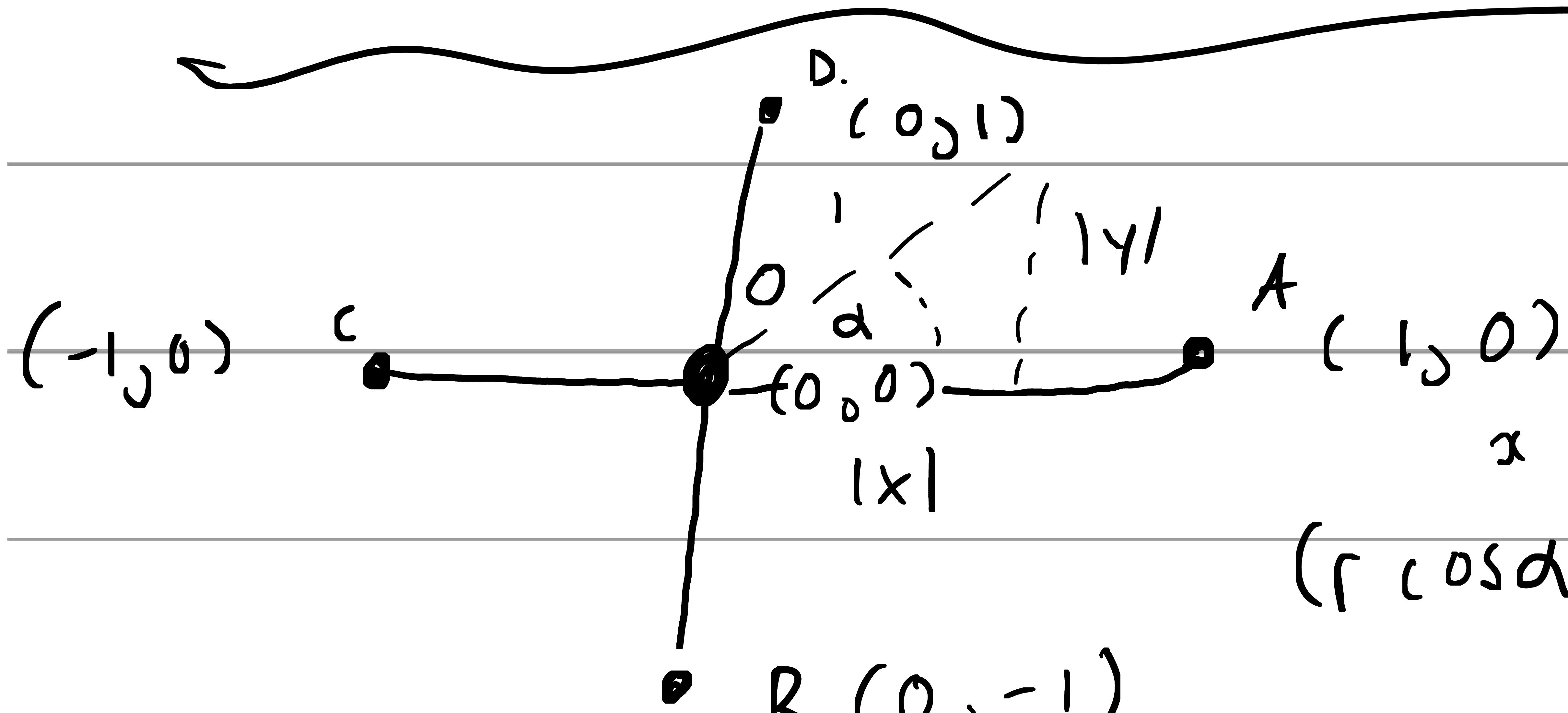
example



$a \cdot b = 0$

$|a| = 0$

$|b| = 1$

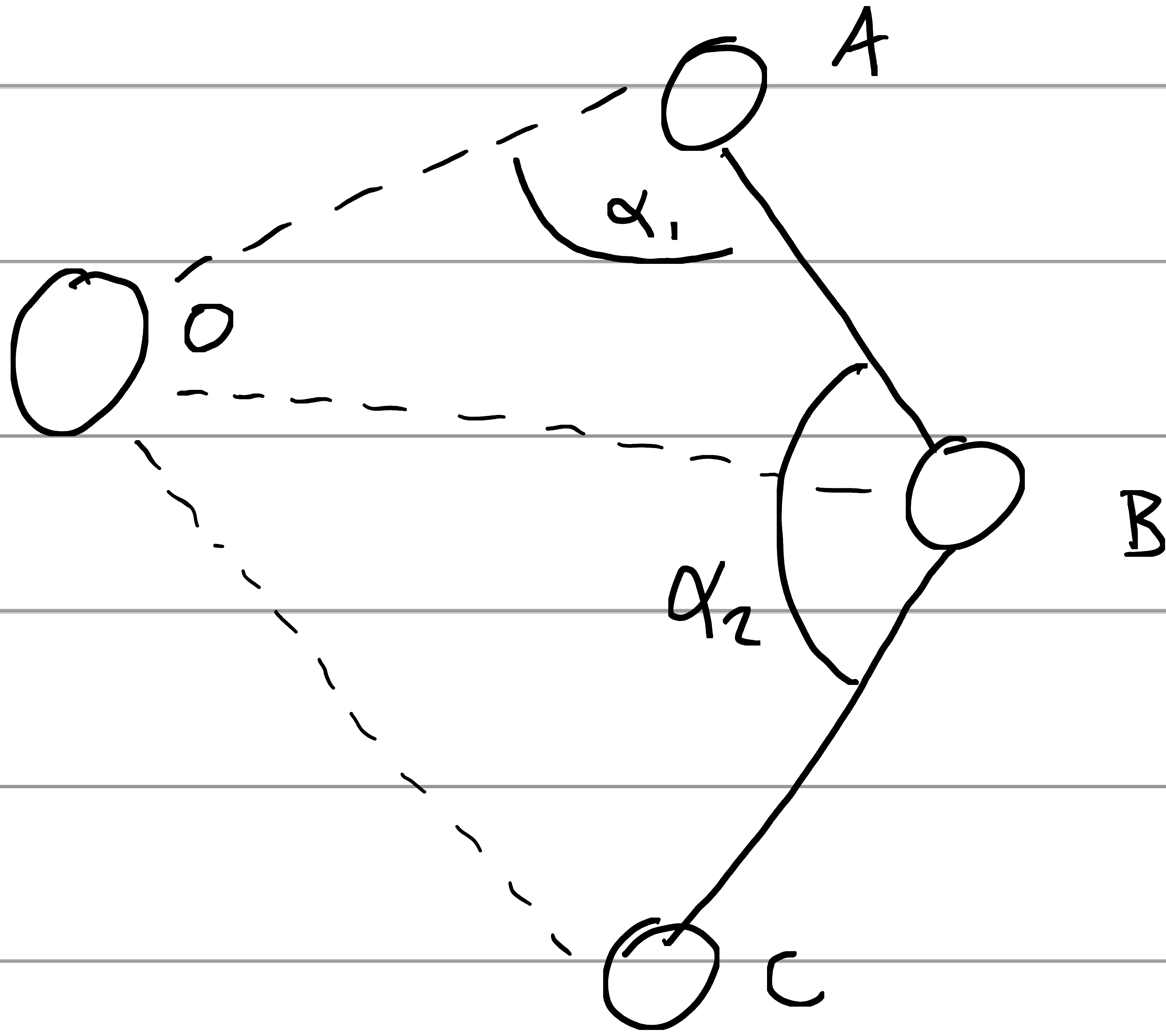


$x = r\cos\alpha$

$y = r\sin\alpha$

$x \quad y$

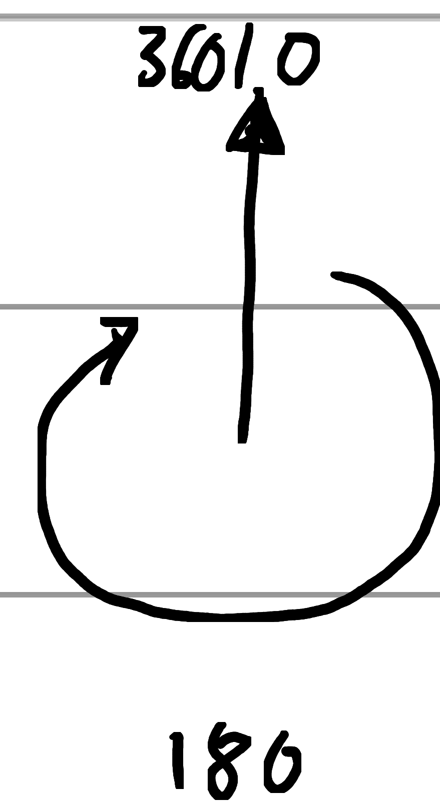
$(r\cos\alpha, r\sin\alpha)$

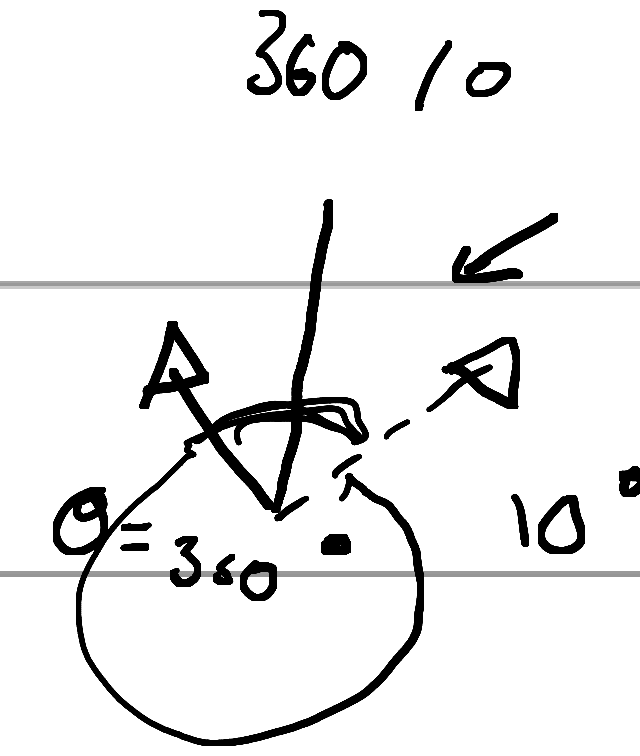


$$\vec{OA} \cdot \vec{OB} = |\vec{OA}||\vec{OB}|\cos\alpha_1$$

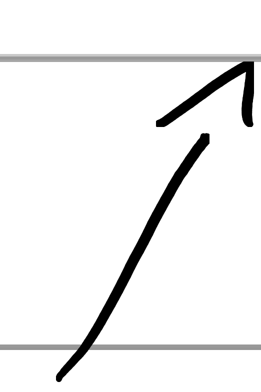
$$\vec{AB} \cdot \vec{AC} = |\vec{AB}||\vec{AC}|$$
$$\cos\alpha_2$$

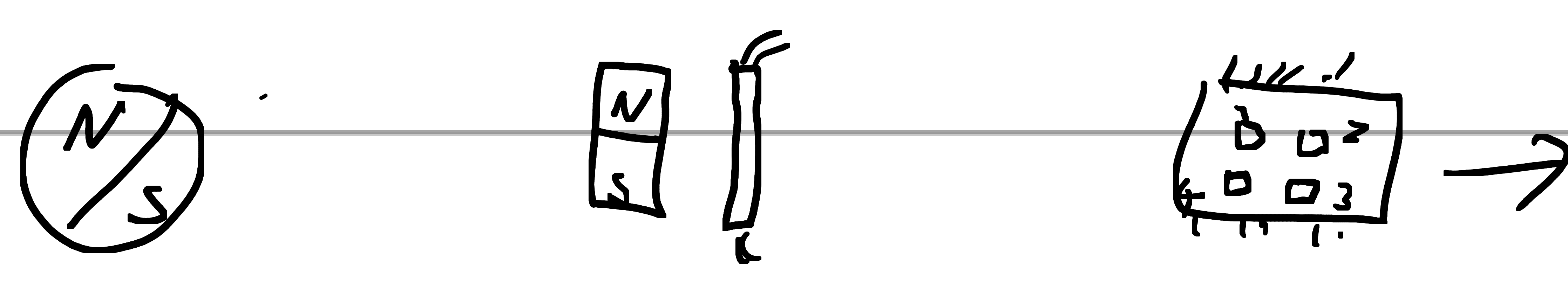
Example O(0,0) A C





1°





$0 \to 2^{64}$

$\sim 16{,}000$

$\downarrow$

0

27/5/22

→ Pendulum angle code + modular funs done. ✓

→ charting of state estimates graph 60hz ✓

→ Need realtime hw timer for dt python
dt ~ 0.00228
done ... need scripts for venv ✓

Next

Repo 1

→ Encoder SPI interface + Jerk estimate

(Need sensor + basic rotation assembly, manual rotation)

Repo 2

→ ADC + Encoder

Repo 3

→ CMP + Encoder

→ serial plot check

→ generate switching map (Need coils + sensor on axis)

→ driving ESC + Encoder first version

→ Fixed rpm test

→ Fixed rpm measure asymmetry
→ Compensate for asymmetry
→ Record data + review 1

→ Kalman PID virtual

→ Review 2

→ Kalman hardware

→ Review 3

→ Motor Review

→ Hardware inputs, speed, pitch, etc → capacitive → digital pot
→ OLED Display. → app lol.

28/5/22

→ Did draft of plotter & created test script

Todo rpm scripts → venv setup of ! env → pip freeze / install → Activate → run bokeh

29/5/22

ADC pwm frequency, recover old work, New hardware timer integer counter, sqwave to map the zero crossing forwards/backwards. Longest continuous run with virtual neutral above 0.

Positions, kalman C++ version. Arduino matrix library.

$\underbrace{\text{ADC + time measurement}}_{\text{@pwm hz}}$, + ZC on host

30/5/22

Kalman

$$P = \begin{bmatrix} \sigma_x^2 & \sigma_x^2/T & \sigma_x^2/T^2 & 0 \\ \sigma_x^2/T & 2\sigma_x^2/T^2 & \frac{3\sigma_x^2}{T^3} & \frac{5}{6}\sigma_j^2 T^2 \\ \sigma_x^2/T^2 & 3\sigma_x^2/T^3 & 6\sigma_x^2/T^4 & \sigma_j^2 T \\ 0 & \frac{5}{6}\sigma_j^2 T^2 & \sigma_j^2 T & \sigma_j^2 \end{bmatrix}$$

Measurements

$$\begin{bmatrix} x, y \\ x, y \end{bmatrix}$$

P 6×6 H 2×6

$$S = HPH^T + R$$

2×6 6×6 6×2 $\to 2\times2$

$P = 4 \times 4 \qquad H = 1 \times 4$

$S = HPH^T + R$

HPH^T selects $P[0,0]$

PH^T selects $P[0]$ row

$1\times4 \quad 4\times4 \quad 4\times1 \qquad 1\times1$

$$K = \overbrace{P\,H^T}^{4\times1} \times \overset{1\times1}{S^{-1}}$$

$4\times4 \quad 4\times1$

31/5/22

→ Finished prototype of kalman code in its branch.

seems to work well with control of the jerk & higher derivative noise compared to the Euler method.

Thinking about how to port it for teensy

→ arm-math has floating point matrix math.

Stages of Kalman

Project state forward (involves X) → double

Project error forward can be floats Matrix!

Calculate Kalman gain can be floats No matrix

Calculate innovation (involves X) → double

update covar matrix can be float Matrix

Is float → double worth it?

calculating S only needs to extract one matrix value, We care about theta not overflowing /

going out of sync

Custom code for X propagation with F (double)

Matrix code for P error propagation / update (f32 Matrix)

custom code for innovation / residual & final state (double)

Leave this for later, look at the Kalman extension needed for PID value optimisation before trying to optimise this! As it could affect the state vector

Power control

duty cycle pwm drives rpm against a load to an equilibrium.

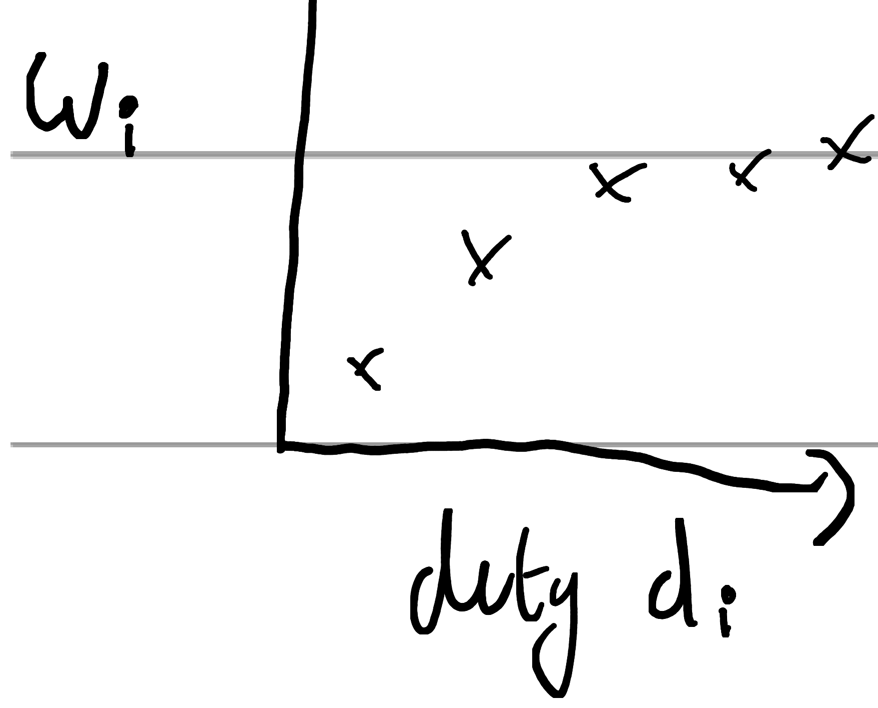
$$\tau_M = m\left(\alpha + \int_{T_1}^{T_2} J\,dt\right)$$ motor torque

@ Equilibrium

$$0 = \tau_M - \underset{\sim}{\tau_r}$$ motor resistance / load

approximately a constant ... probably $\tau_r(\omega)$

So we know we will give it some low duty & measure the velocity, do this a number of times to find correlation.



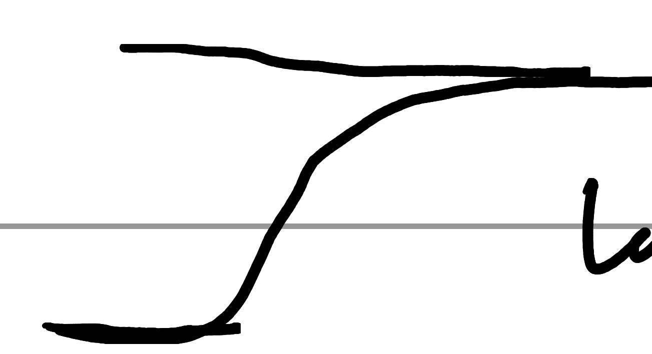
1/6/22

Tuning:

1) Set all gains to zero
2) Increase gains until speed fluctuates around set point
3) Set PID to minimise oscillation
4) Raise I until you reach setpoint & oscillation dampens.

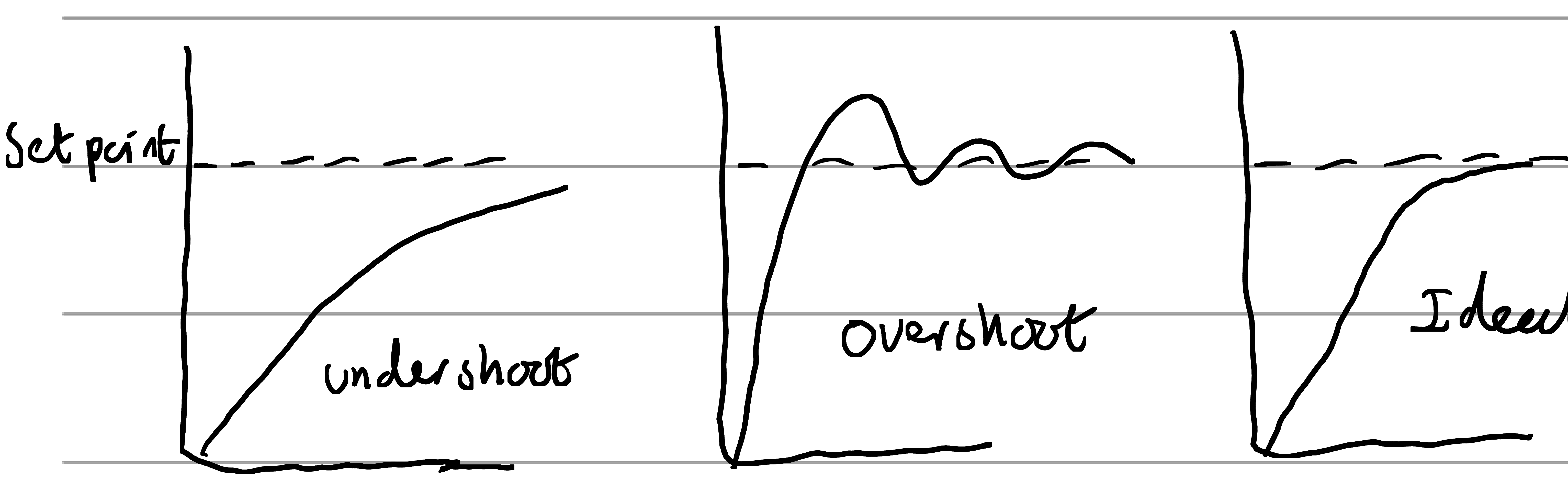
BCO

$f(x) = \frac{L}{1+Ae^{Bx}}$



logistic

What can a PID controller do?



Factors

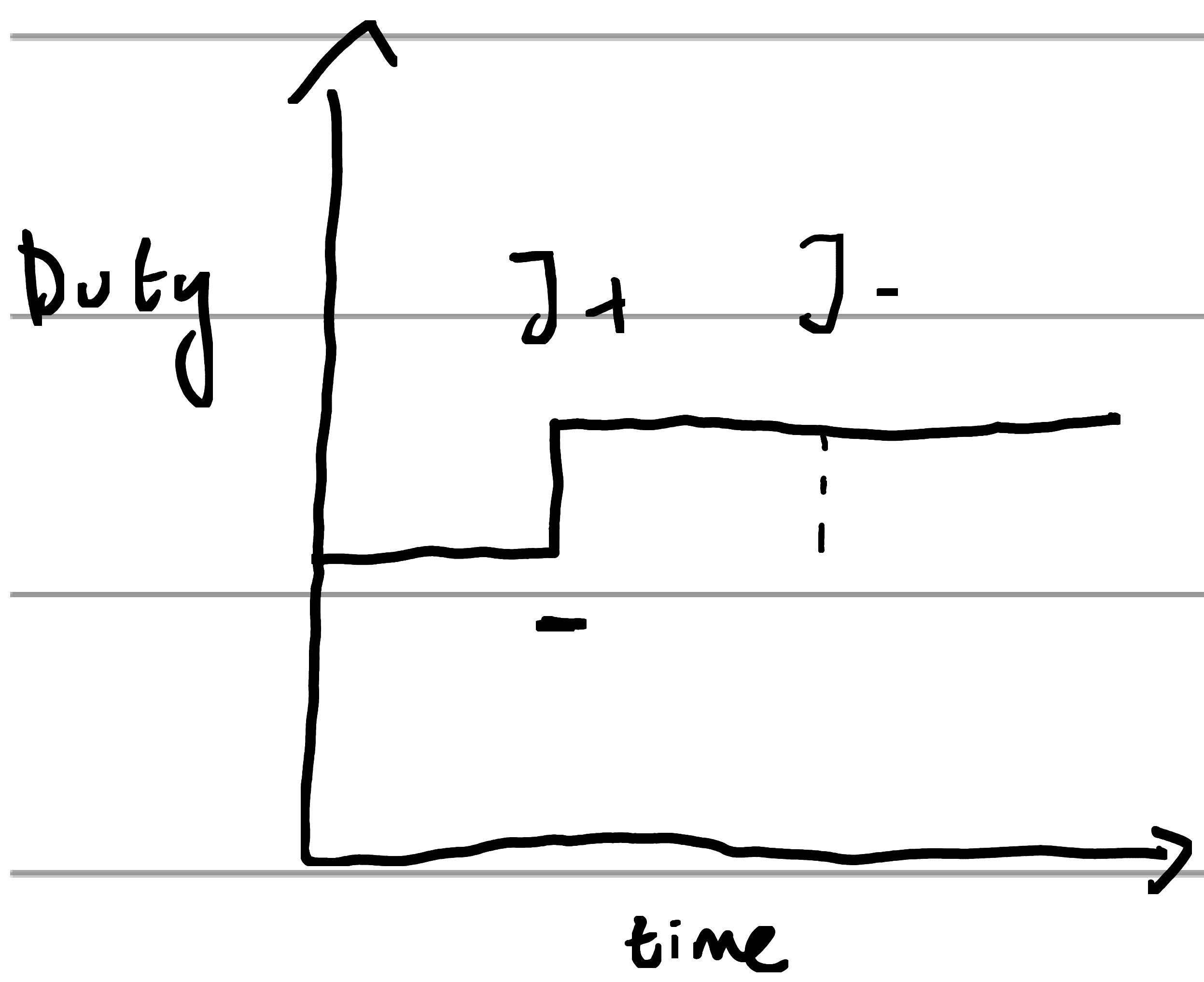
→ Process gain, how much amplitude changes & becomes stable after applying a step change.

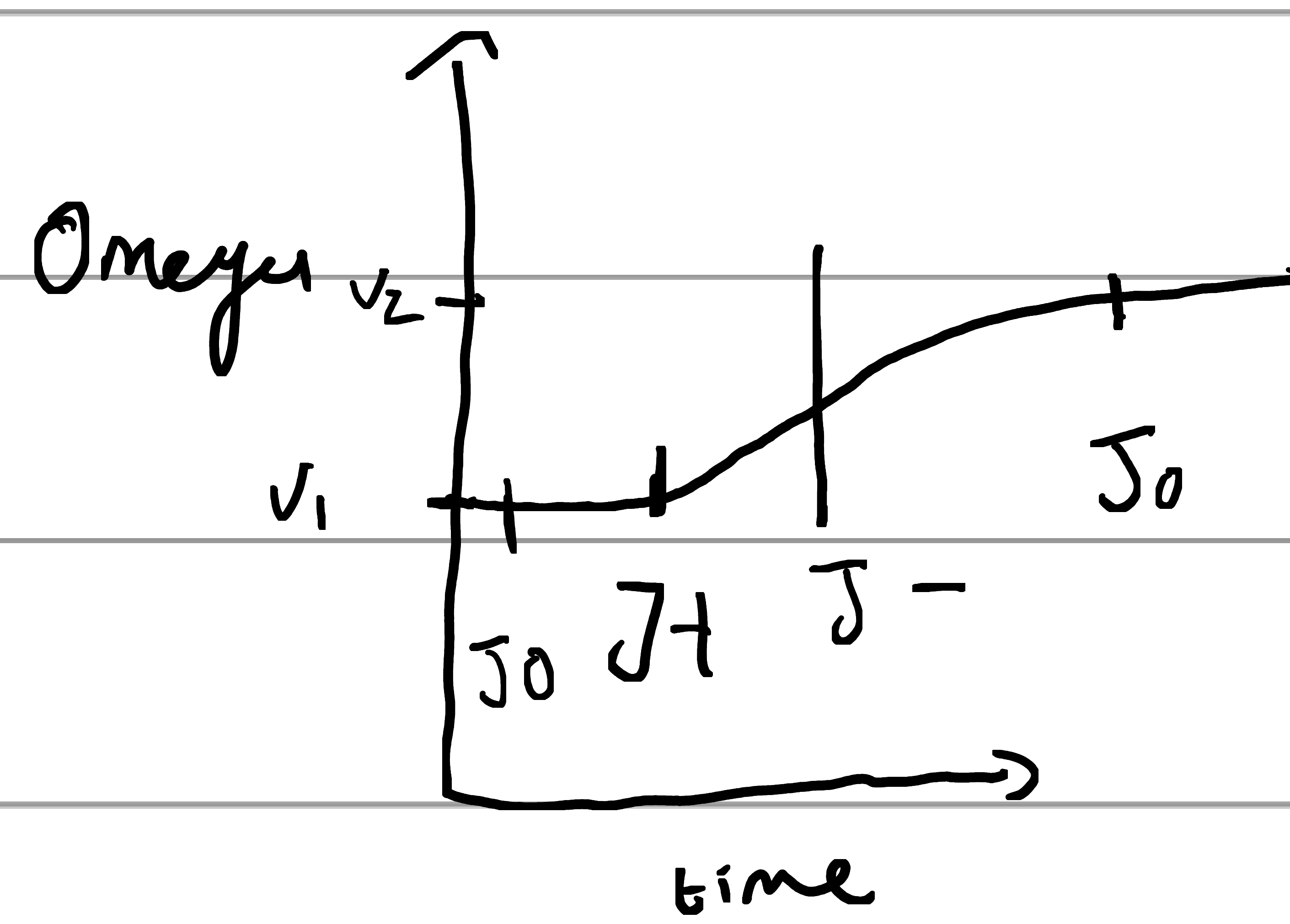
→ Process response time, time to change from one steady state to another after a step change

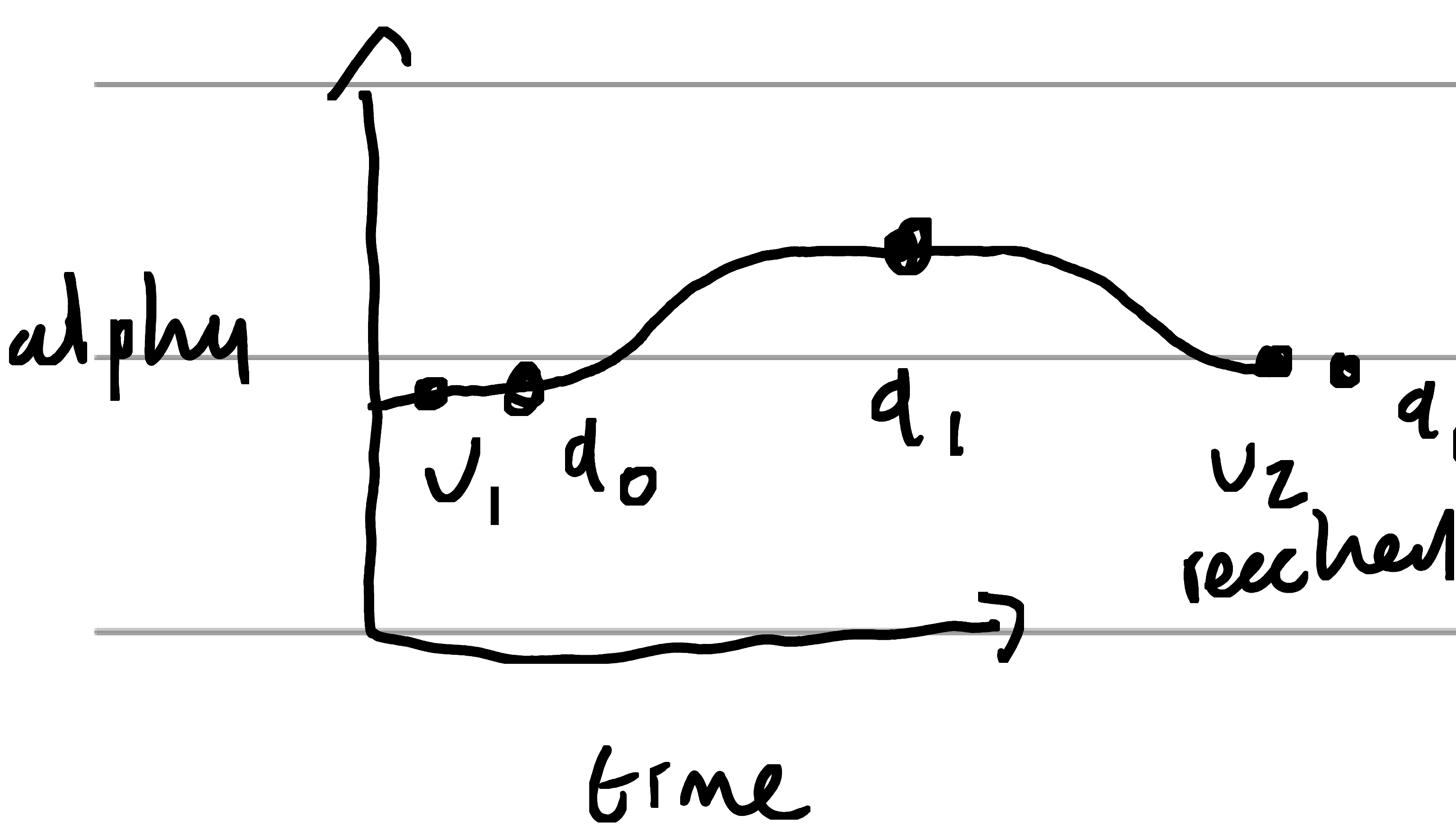
→ Dead time, period between step being applied (latency)

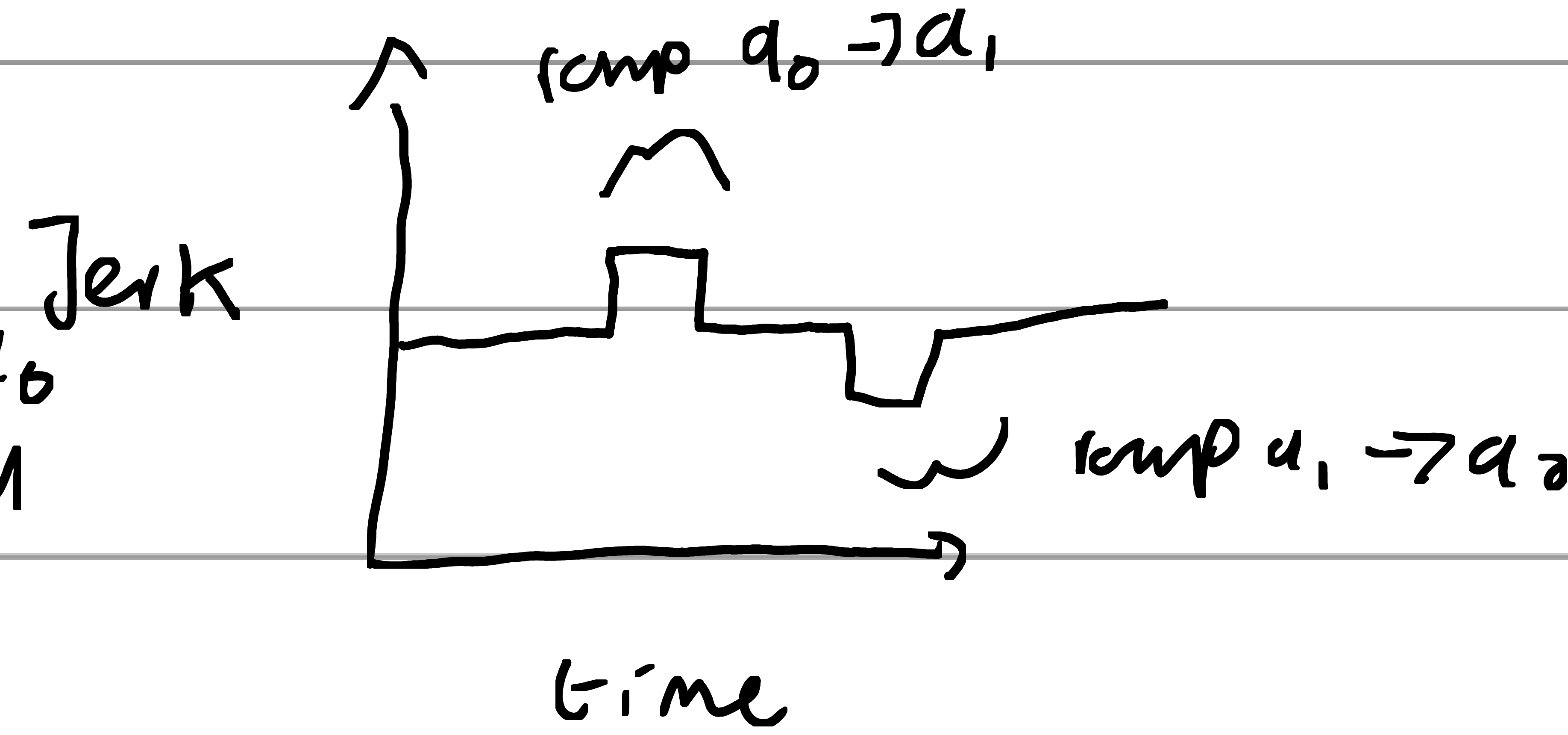
→ Some process responses increase infinitely when applying a step, this is called an integral response.
I think Jerk target is one of these.

AC motor system. (change duty)









$$K_p \sim e^{-\theta / 1 + T_p}$$

T_p - process lag (time constant)

θ - process delay

K_p - process gain

$$\omega, \alpha, J \qquad D(T)$$

$$\omega(T, D(T))$$

Friction

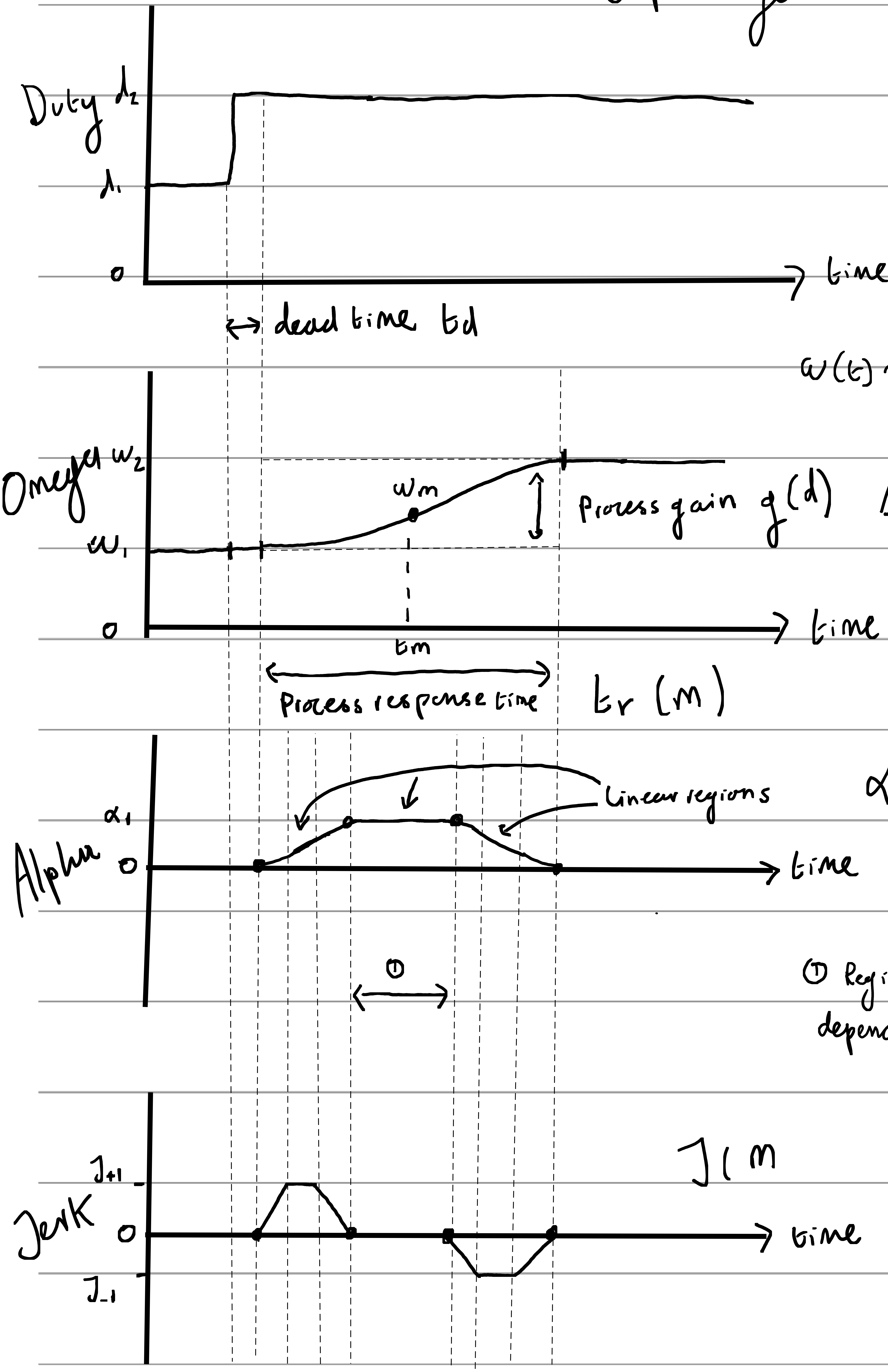
$$F(\omega(t))$$

$$\alpha\left(T, \frac{\partial D(T)}{\partial T}, F(\omega(t))\right)$$

its not just friction

$$q(\omega(t)) \qquad J$$

AC Motor Step change



$$t_m = \frac{t_{w_2} - t_{w_1}}{2}$$

$$\omega(t) \sim \frac{w_2}{1+e^{-g(t-t_m)}}$$

$\Delta d \sim \Delta w_{equilibrim}$

$\Delta w_{eq} = k \Delta d$

$t_r(m)$

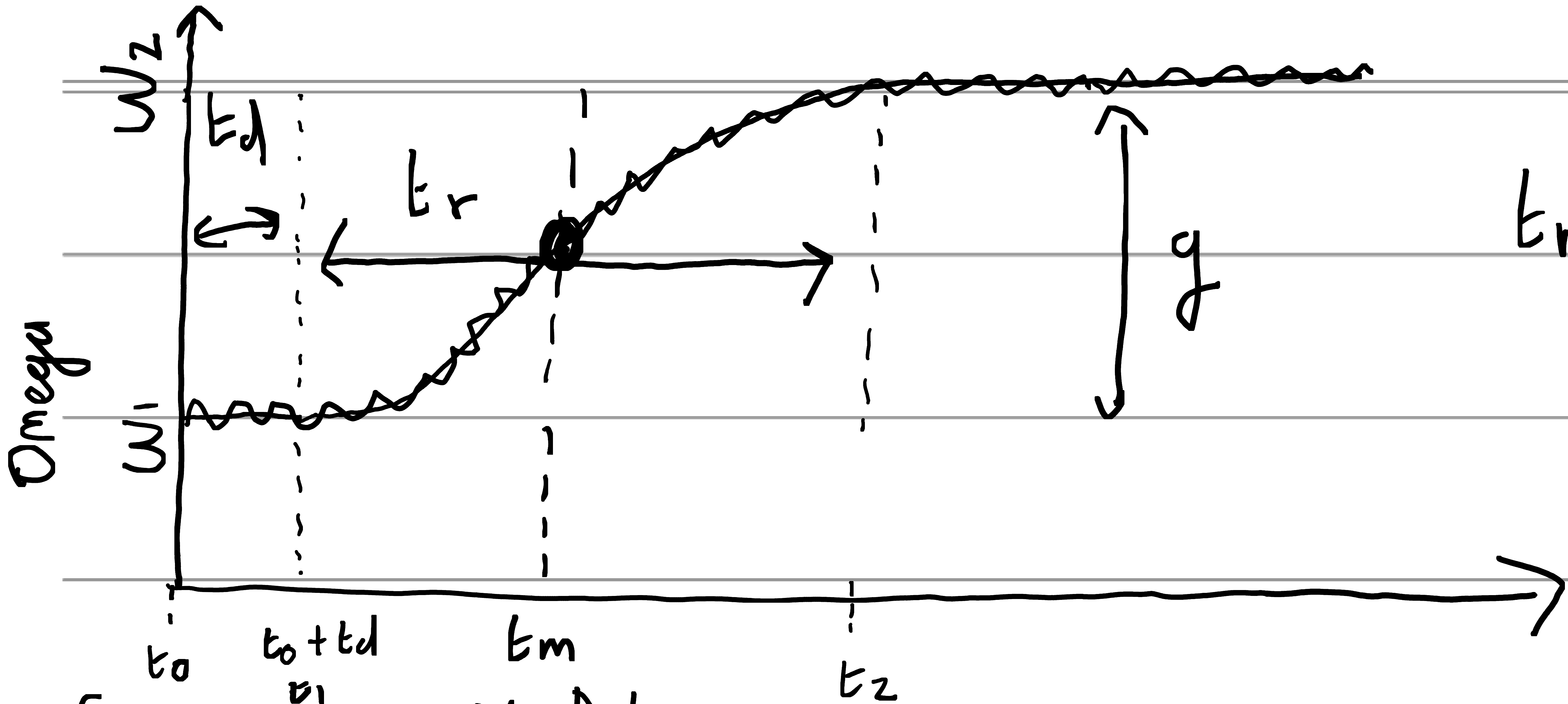
aka load

$\alpha(\Delta d, w, m)$

① Region with low dependancy of α on w

$J(m$

Integral response?



$$t_m = t_0 + t_d + \frac{t_r}{2}$$

Equilibrium Omega ~ Duty

$$\overline{\omega}_{1eq} \sim d_1 \qquad \overline{\omega}_{2eq} \sim d_2$$

$$\overline{\omega}_{1eq} = K d_1 \qquad \overline{\omega}_{2eq} = K d_2$$

K can be found by doing Δd step change & measuring Δω after a long time >> T_r by measuring $\overline{\omega}_1$ & $\overline{\omega}_2$ averages. (d, t, ω) source data

$g = K(d_2 - d_1)$ gain

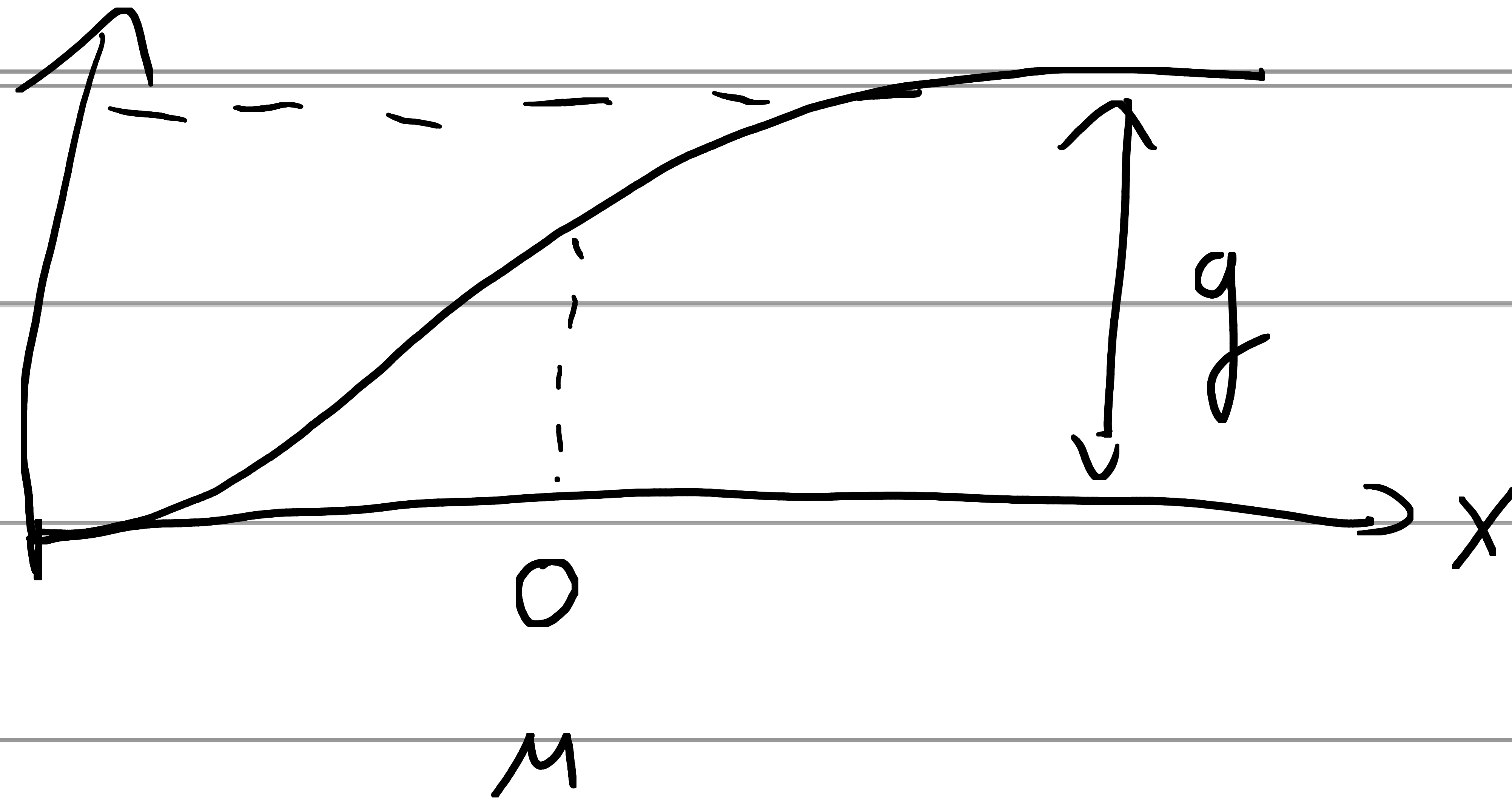
$t_m = \dfrac{t_2 - t_0 - t_d}{2}$ t_0 is where duty changes

$S_{\Delta D} = \text{Sign}(d_2 - d_1)$

$\beta \sim t_r = t_2 - t_0 - t_d$

$$\omega \approx \frac{g}{e^{\frac{-(t - t_m)}{\beta}} + 1} + (\text{noise}) + (\omega_1)$$

1



$$\frac{1}{e^{\frac{-(x-\mu)}{\beta}}+1}$$

$$\frac{g}{e^{\frac{-(x-t_m)}{\beta}}+1}$$

β is currently a guess!

$$e^{\frac{-(t_z-t_m)}{\beta}}=0$$

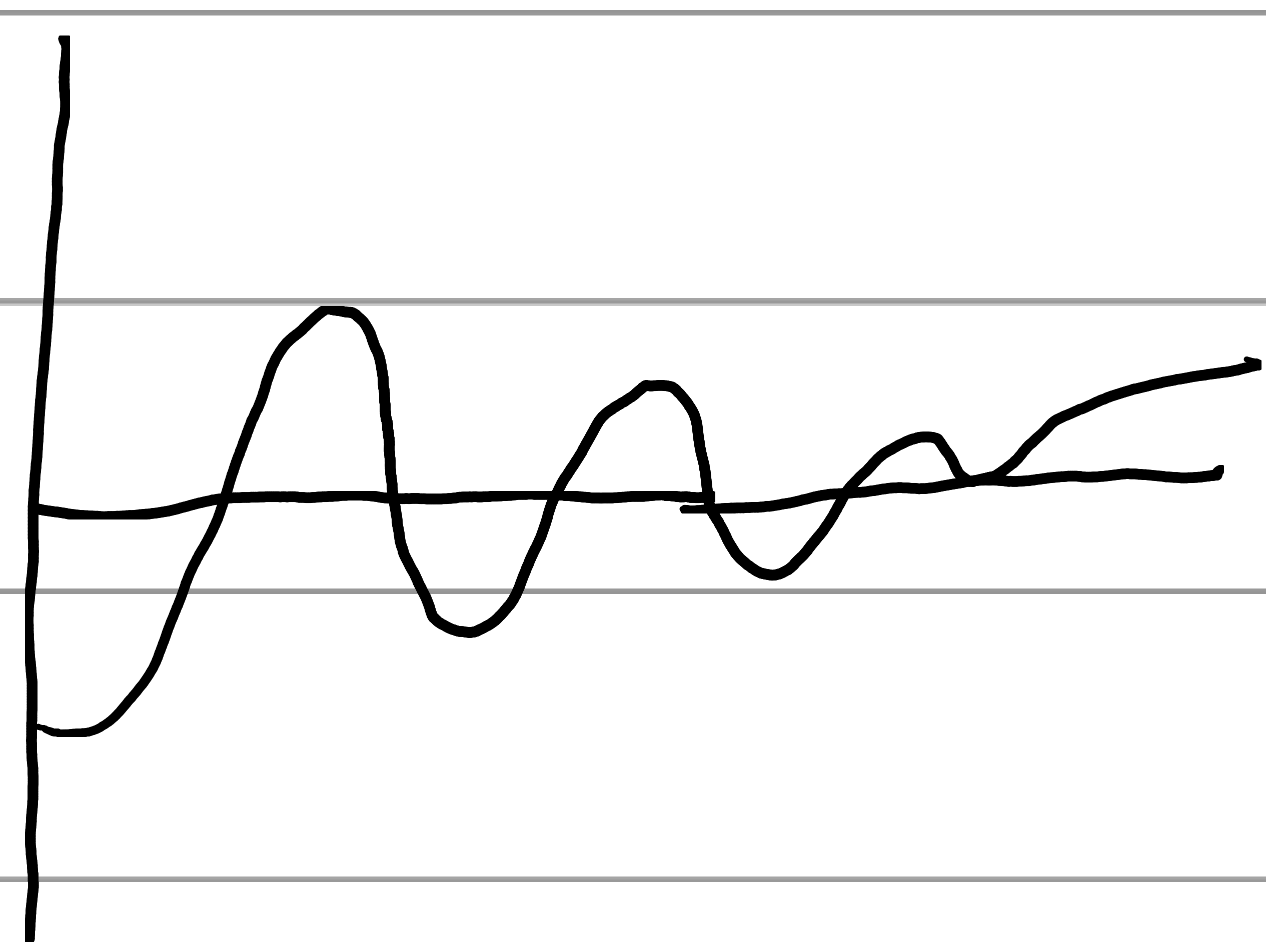
$$e^{\frac{t_z-t_m}{\beta}}=0$$

$$e^{t_z/\beta}\,e^{\frac{-t_m}{\beta}}=0$$

PB proportional P

Td time derivative D

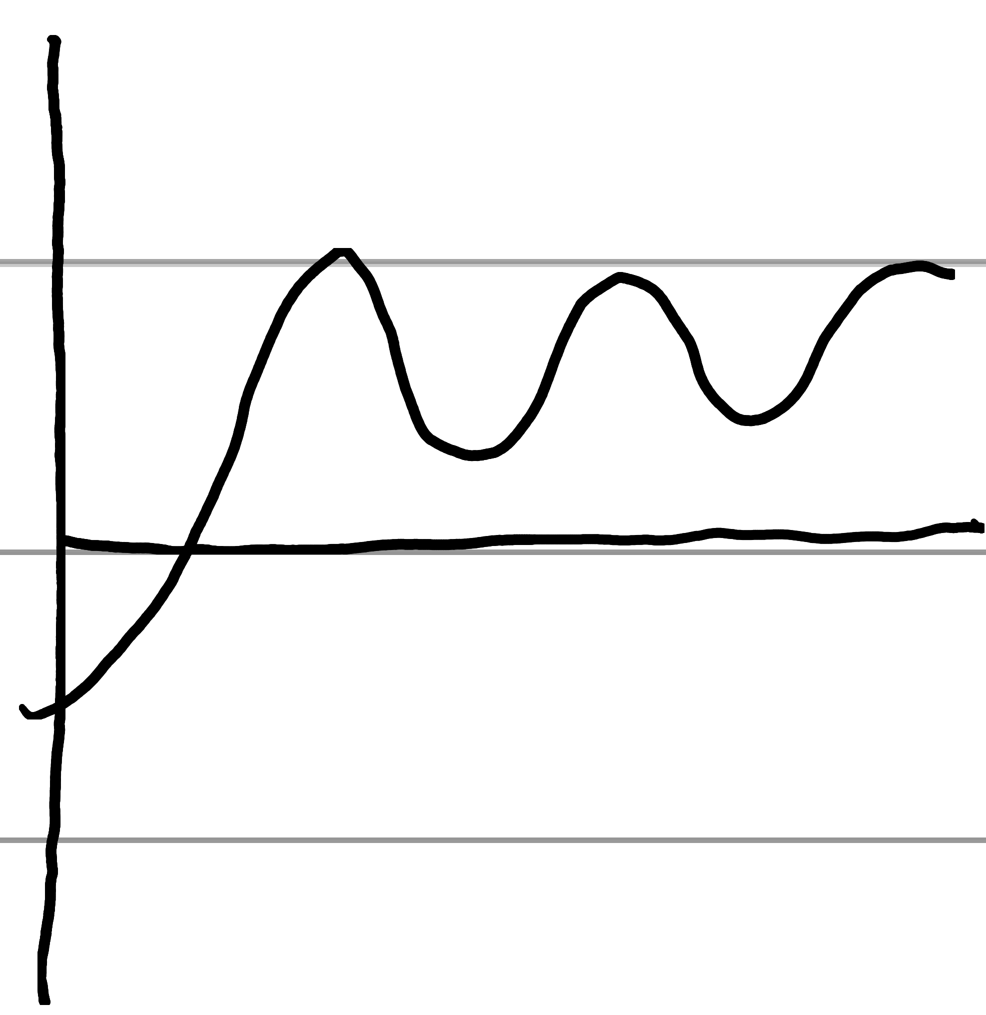
T_I integral I



when oscillating with a small overshoot

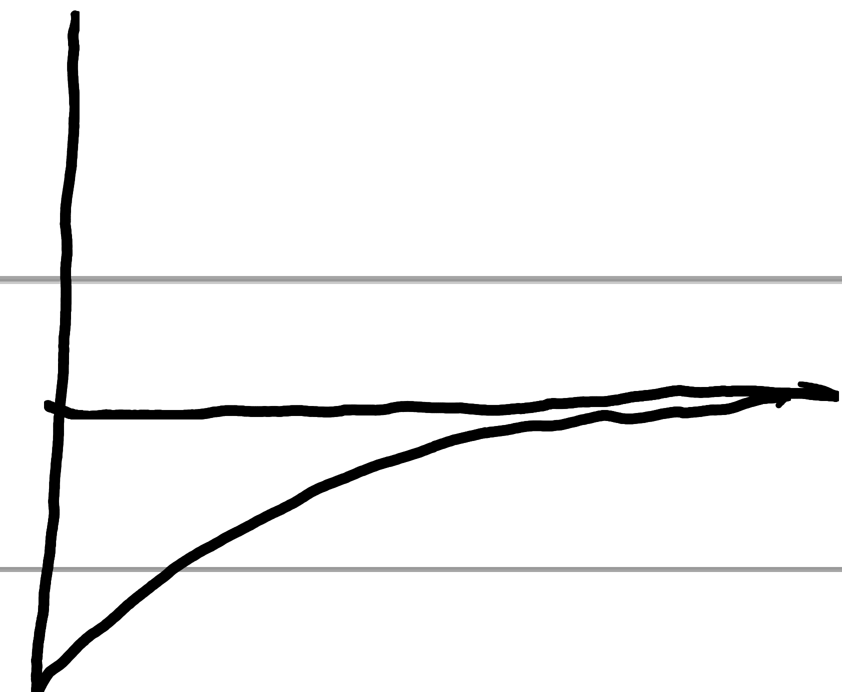
→ decrease K_I

→ increase or decrease K_p



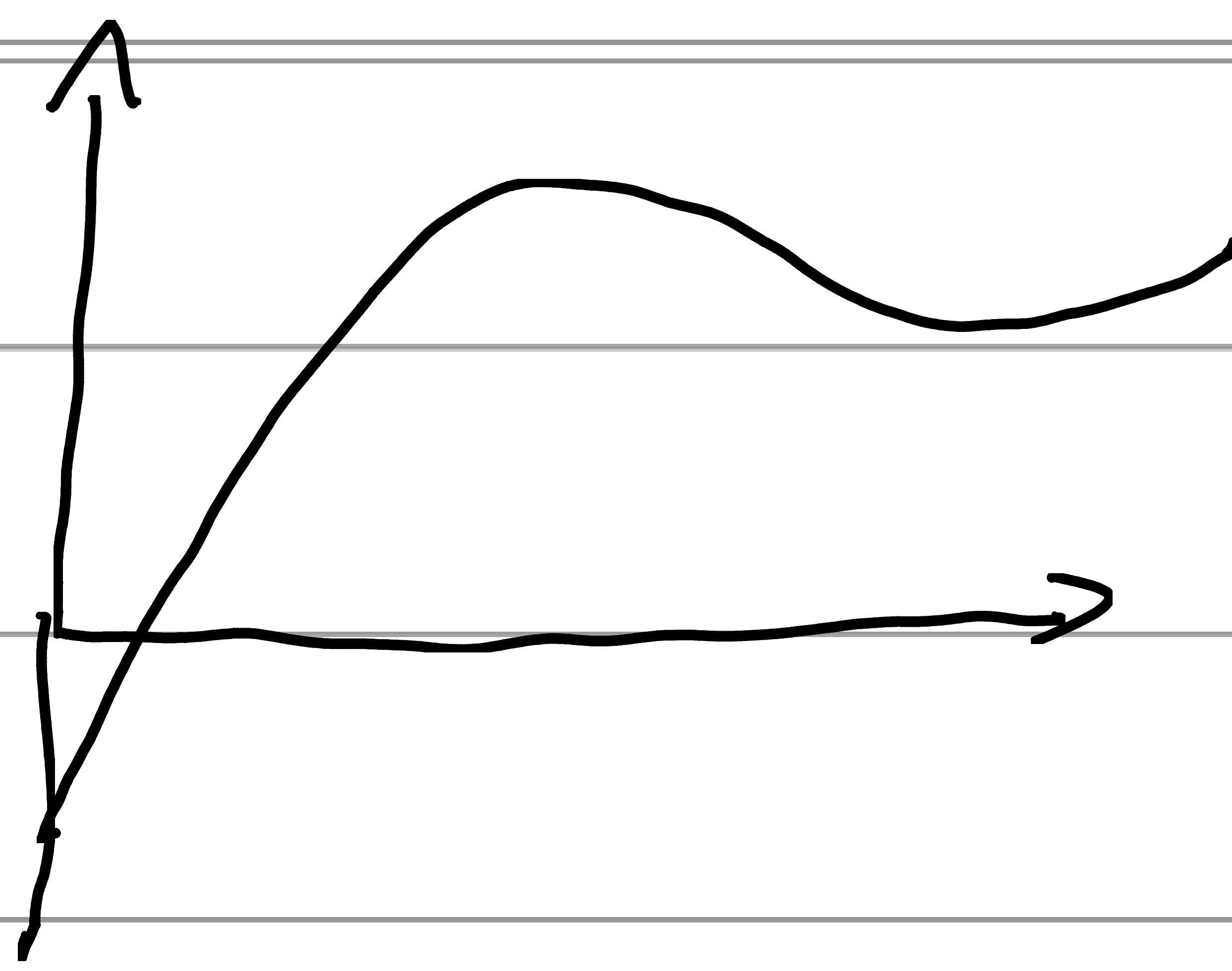
when large overshoot & just conversion

→ Increase or decrease both K_p & K_I



when no oscillation & slow conversion.

→ Decrease K_p & K_I



when slow oscillating

& slow conversion

→ Decrease K_p

→ Increase K_I

Slow conversion

yes → Oscillating

yes → K_p-- K_I++

no → K_p-- K_I--

no → Large overshoot → K_p-- K_I--

or

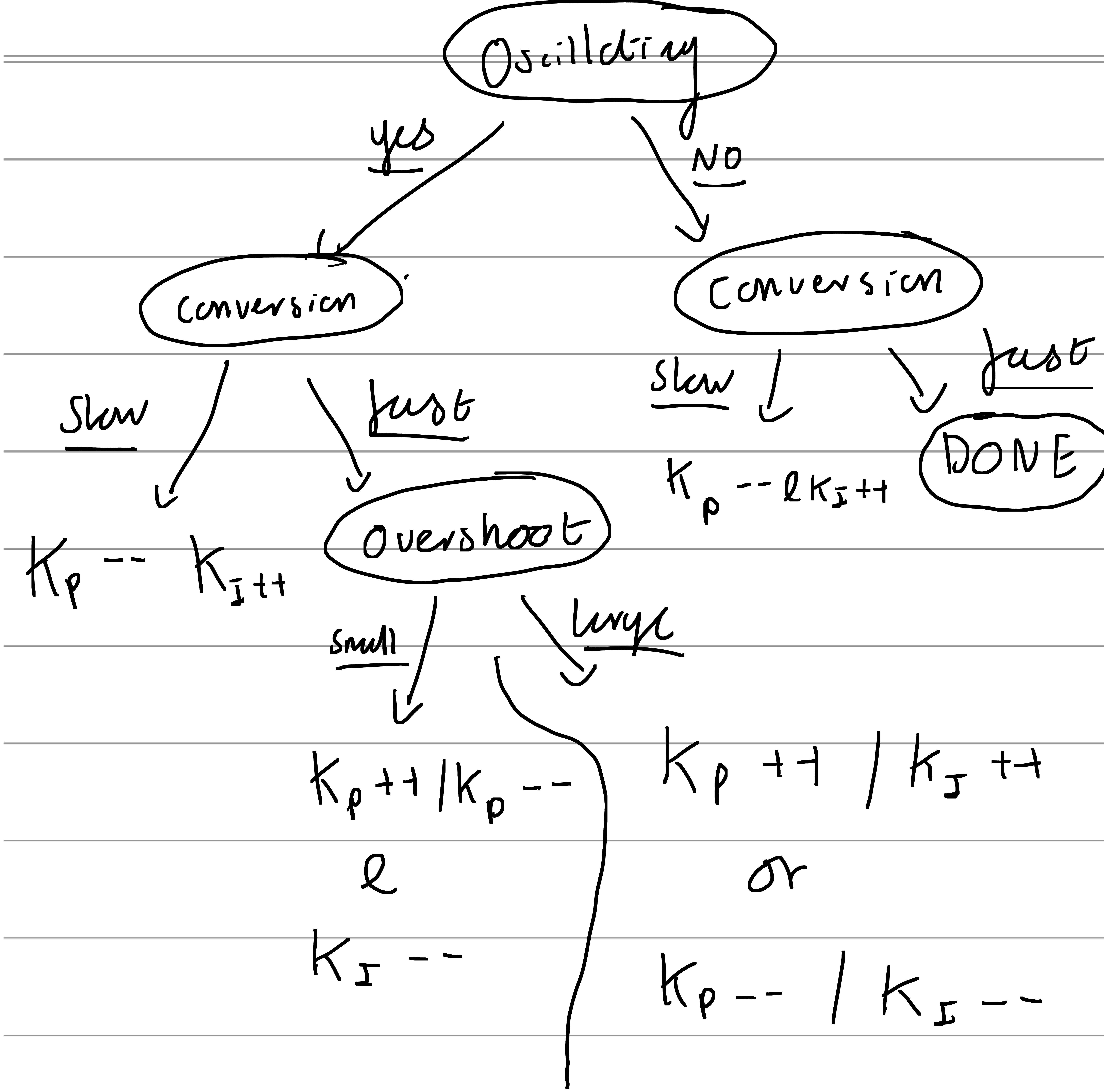
K_p++ K_I++

Oscillating

yes → small overshoot → K_p-- / K_p++

&

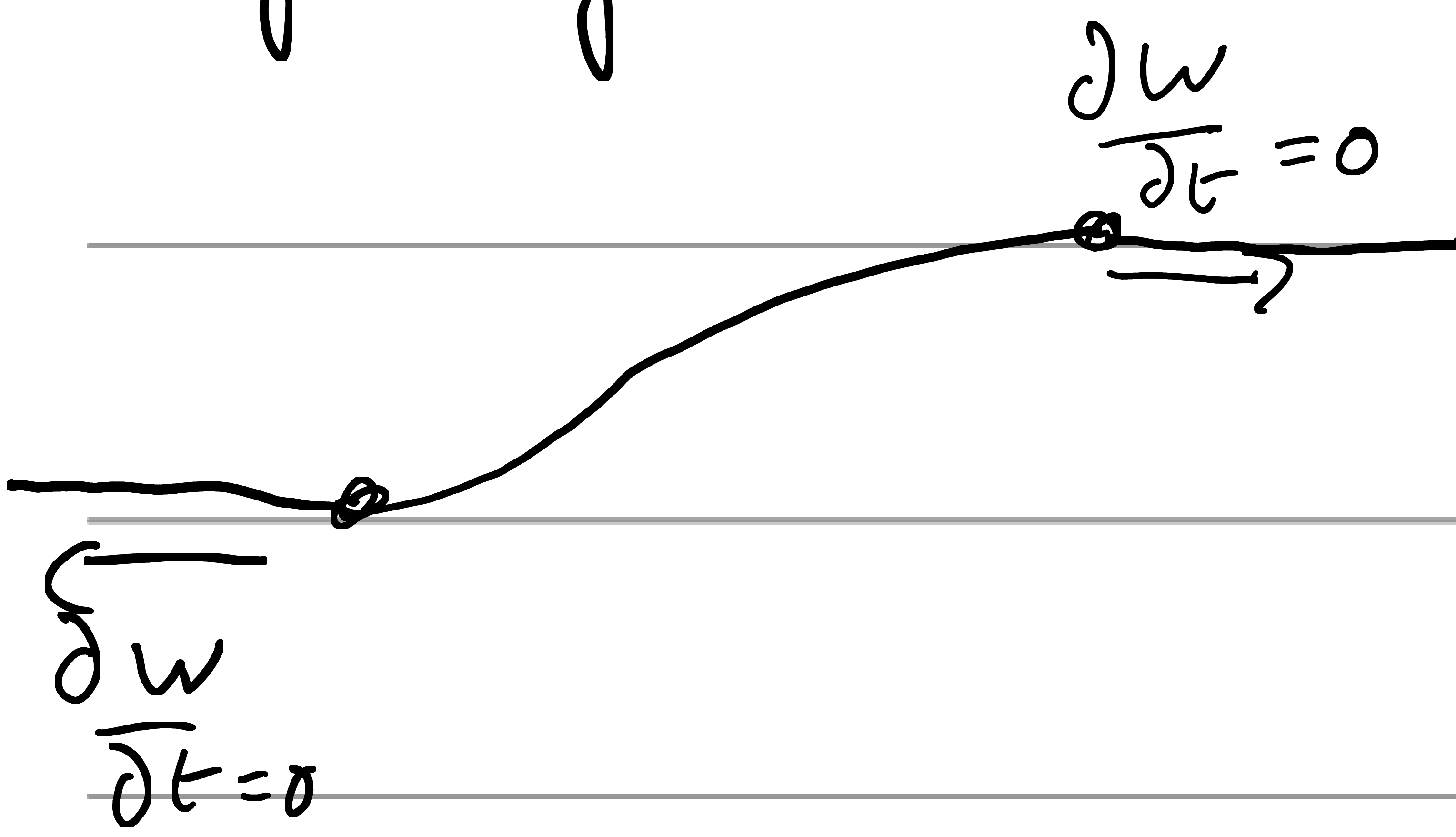
K_I--



What about K_d?

3/6/22

$$\text{gain} = g = K(d_2 - d_1) \qquad w(t) \sim \frac{K(d_2 - d_1)}{1 + e^{-\frac{(t - t_m)}{\beta}}} + \left(\text{sign} \times K d_1\right) + (\text{noise})$$



again!

$$w(t) = \frac{g}{1+e^{(t_m - t)\cdot s}} \qquad s = -\frac{1}{\beta}$$

$$w'(t) = g \cdot \frac{s e^{s(t_m - t)}}{\left(1+e^{((t_m - t)\cdot s)}\right)^2}$$

$$w'(t = t_m) = \frac{gs}{4} = \frac{-g}{4\beta}$$ say we set the grad to straight

$$\frac{W_2 - w_1}{t_2 - t_0 - t_d} = \frac{-(W_2 - w_1)}{4\beta} \qquad t_2 = t_0 + t_d + \frac{t_r}{2}$$

$$W_2 - w_1 = \frac{-(W_2 - w_1)(t_2 - t_0 - t_d)}{4\beta}$$

$$4\beta = \frac{-\cancel{(W_2 - w_1)}(t_2 - t_0 - t_d)}{\cancel{(W_2 - w_1)}}$$

$$\beta = -(t_2 - t_0 - t_d)/4 = \frac{-(\cancel{t_0} + \cancel{t_d} + \frac{t_r}{2} - \cancel{t_0} - \cancel{t_d})}{4}$$

$$= -\frac{t_r}{8}$$

$$\frac{gS}{4} = \frac{w_2 - w_1}{t_2 - t_0 - t_d} \qquad t_2 = t_0 + t_d + \frac{t_r}{2}$$

$$S = \frac{4\,(\cancel{w_2 - w_1})}{(t_2 - t_0 - t_d)\,(\cancel{w_2 - w_1})}$$

$$S = \frac{4}{\cancel{t_0} + \cancel{t_d} + \frac{t_r}{2} - \cancel{t_0} - \cancel{t_d}} = \frac{2 \cdot 4}{t_r}$$

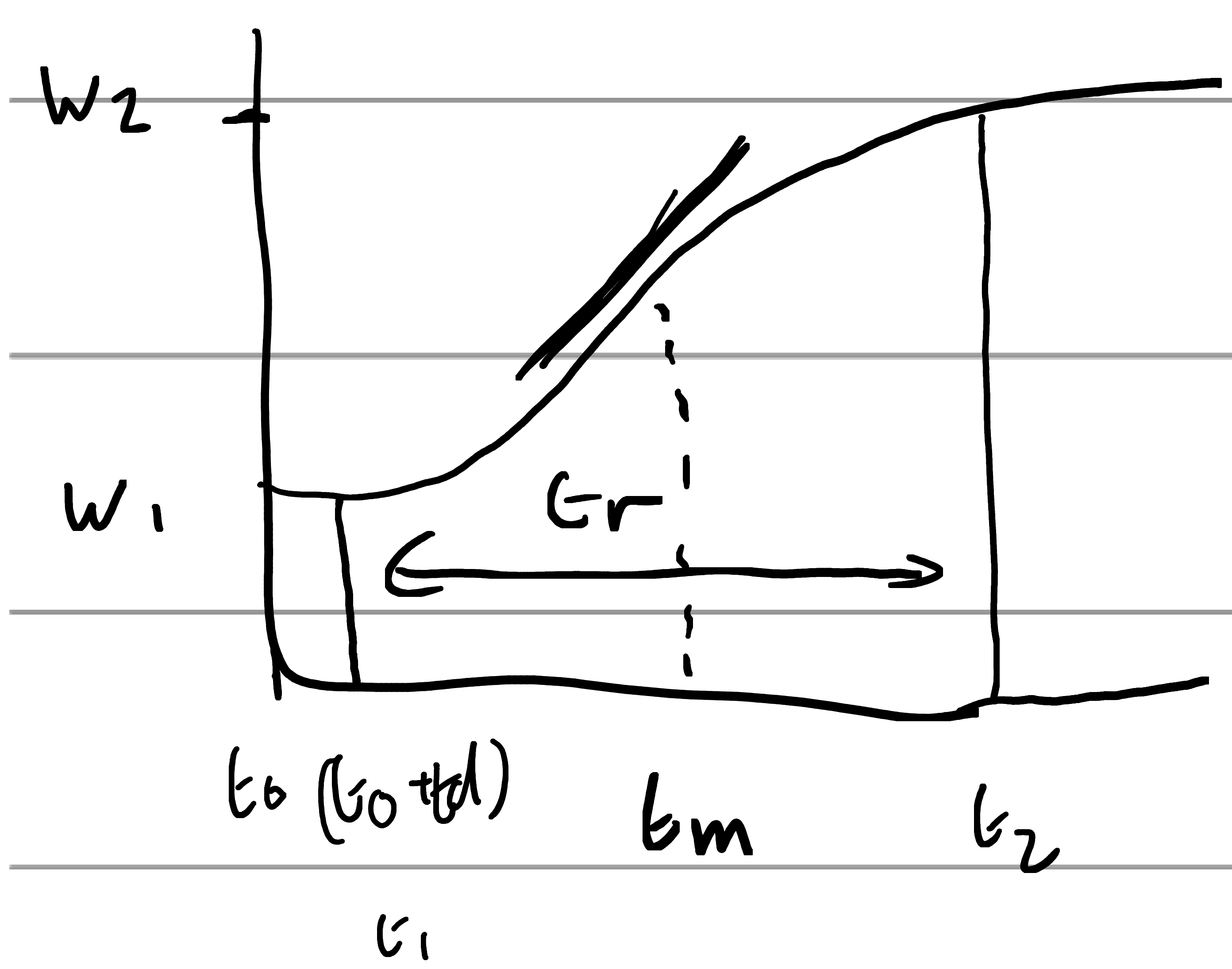
$$S = \frac{8}{t_r} \qquad \beta = -\frac{1}{S}$$

$$\beta = -\frac{t_r}{8}$$

$$\omega \approx \frac{w_2 - w_1}{e^{\frac{-(t-t_m)}{\beta}} + 1 + (w_1 \times S_{\Delta D})} + (\text{noise})$$

known

$$t_m = t_0 + t_d + \frac{t_r}{2}$$



$w = w_2$ @ $t = t_2$

$w = w_1$ @ $t = t_0$ $\qquad \frac{gs}{4}$

w' @ $t = t_m$ $\qquad w' = \frac{g}{4\beta}$ max

$$t_2 = t_0 + t_d + t_r$$

$$w_2 = \frac{w_2 - w_1}{e^{-\frac{(t_2 - t_m)}{\beta}} + 1} + w_1 \cdot \cancel{\text{sign}(\ldots)}$$

$$\frac{w_2 - w_1}{t_2 - t_0 - t_d}$$

$$\frac{g}{\cancel{t_0} + \cancel{t_d} + t_r} = \frac{w_2 - w_1}{t_r}$$

$$w_2 - w_1 \cdot \cancel{\text{sign}(\ldots)} = \frac{w_2 - w_1}{e^{-\frac{(t_2 - t_m)}{\beta}} + 1}$$

$$e^{-\frac{(t_2 - t_m)}{\beta}} = \frac{w_2 - w_1}{w_2 - w_1 \cdot \cancel{\text{sign}(\ldots)}} - 1$$

$$-\frac{(t_2 - t_m)}{\beta} = \ln\left(\frac{w_2 - w_1}{w_2 - w_1 \cdot \cancel{\text{sign}(\ldots)}} - 1\right)$$

$$\beta = \frac{-(t_2 - t_m)}{\ln\left(\frac{w_2 - w_1}{w_2 - w_1 \cancel{\text{sign}(\ldots)}} - 1\right)}$$

$$\beta = \frac{-(t_2 - t_m)}{\ln\left(\frac{w_2 - w_1}{\underbrace{w_2 - w_1 \cancel{\cdots}}_{\text{this is zero! Poo!}}} - 1 \right)}$$

$$t_2 = t_0 + t_d + t_r$$

$$t_m = t_0 + t_d + \frac{t_r}{2}$$

$$\beta = \frac{-\left(\cancel{t_0} + \cancel{t_d} + t_r - \left(\cancel{t_0} + \cancel{t_d} + \frac{t_r}{2}\right)\right)}{\cdots}$$

$$\beta = - \frac{t_r/2}{\ln\left(\frac{w_2 - w_1}{w_2 - (w_1 \cancel{\cdots})} \right) - 1}$$

$$\ln(1) = 0$$

$$\beta = \frac{-t_r}{-1} \cdot \frac{1}{2} = \frac{t_r}{2}$$

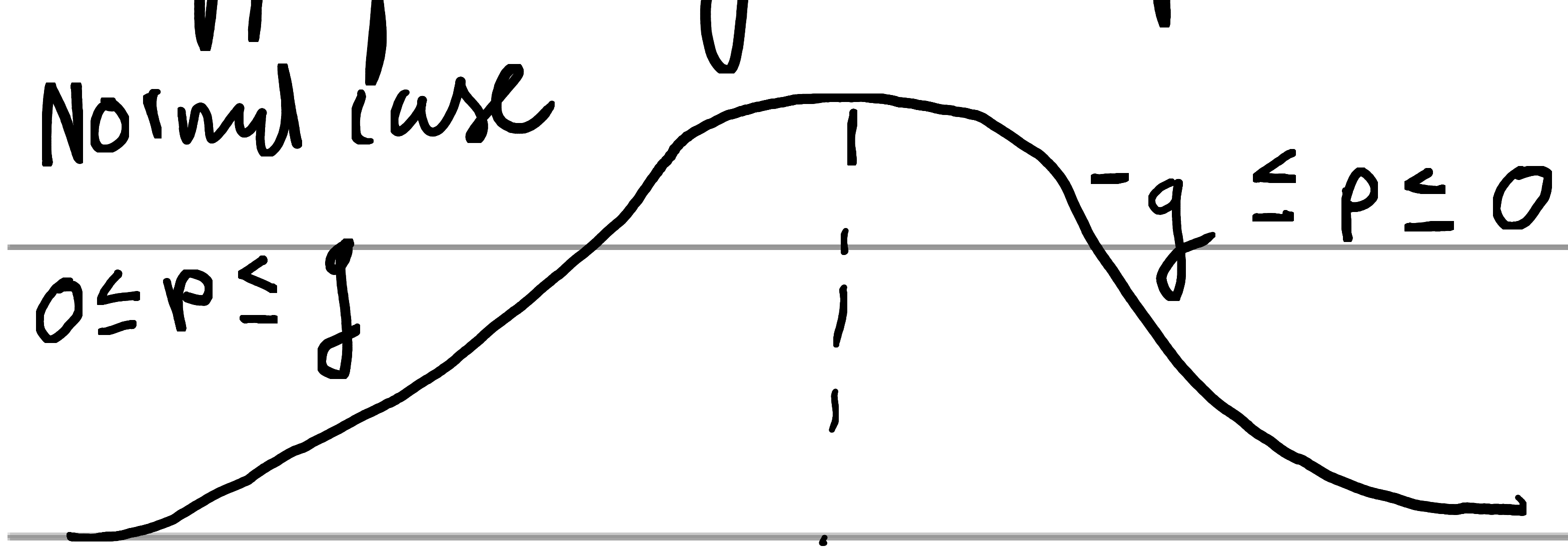
NOPE

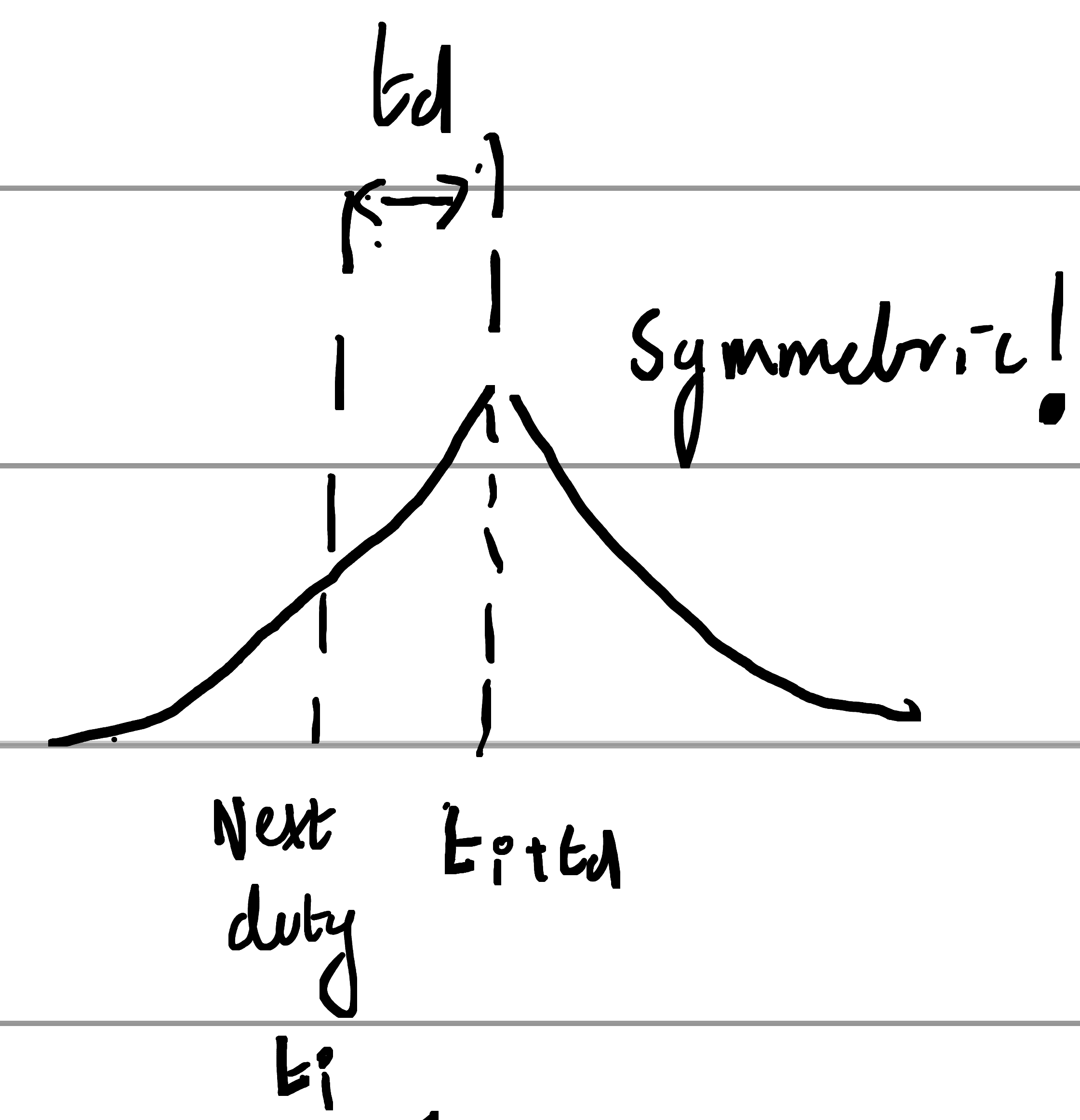
$$W_1 = \frac{W_2 - W_1}{e^{\frac{-(t_1 - t_m)}{B}}} + W_1$$

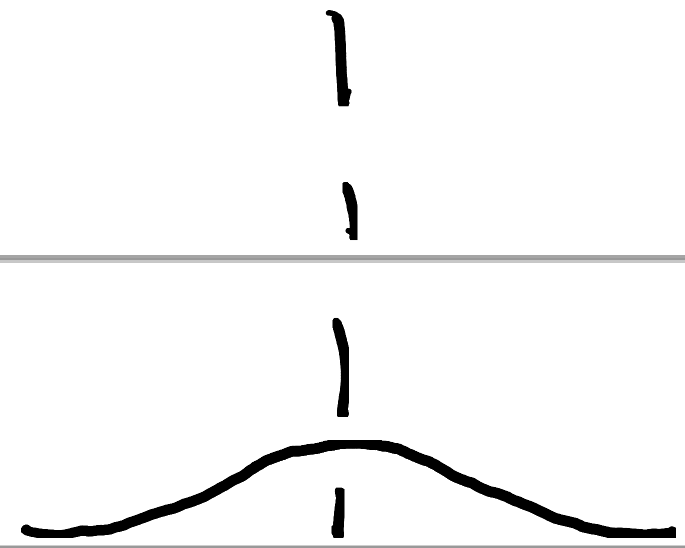
$$p_{factor} = \frac{g}{e^{\frac{-(t - t_m)}{B}}}$$

Todo - simulation

→ modify logistic function generator such that if a duty is incremented & then decremented before the transition response time that it behaves appropriately. Examples:



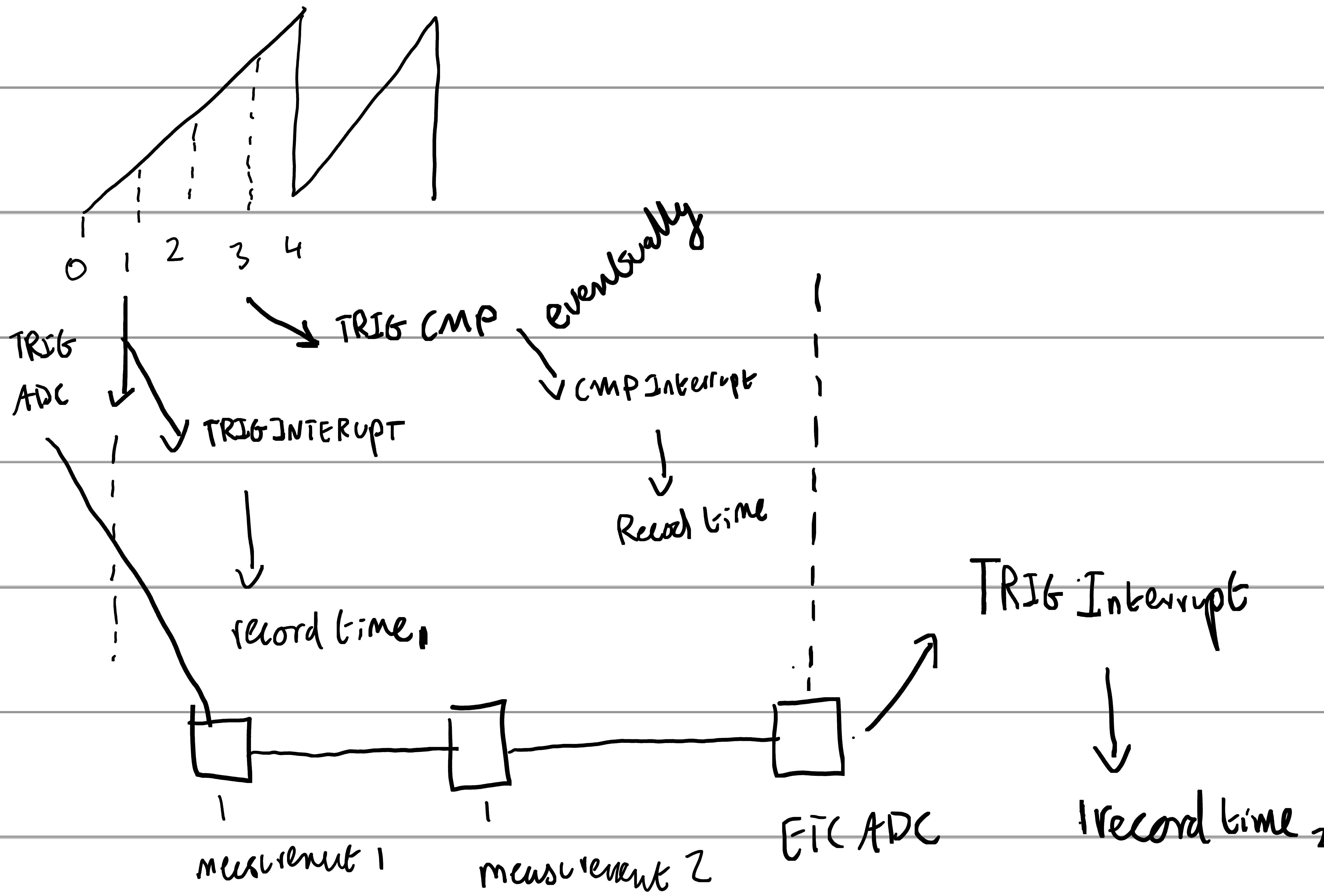




if $\Delta t_{d_1 \to d_2} < t_0 + t_d + t_r$

→ Custom behaviour

ADC



0 1 2 3 4
TRIG ADC
TRIG CMP
eventually
TRIG INTERUPT
CMP Interrupt
record time 1
Record time
TRIG Interrupt
record time 2
measurement 1
measurement 2
EIC ADC