

Wants

→ slow rpm, 3 phase, 6 pole motor.

Have

→ high rpm, beef sensing, positioning arduino,  
performs well.

Dif

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

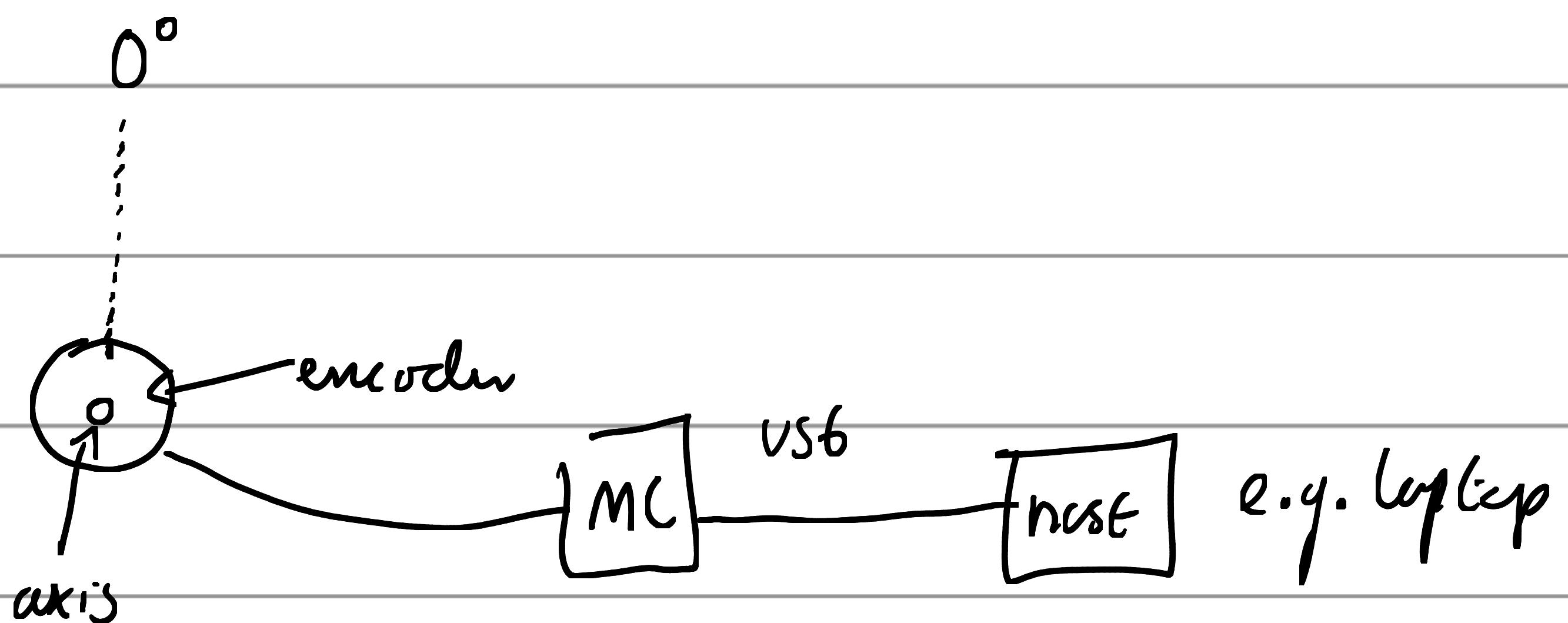
→ Arduino works beef competably

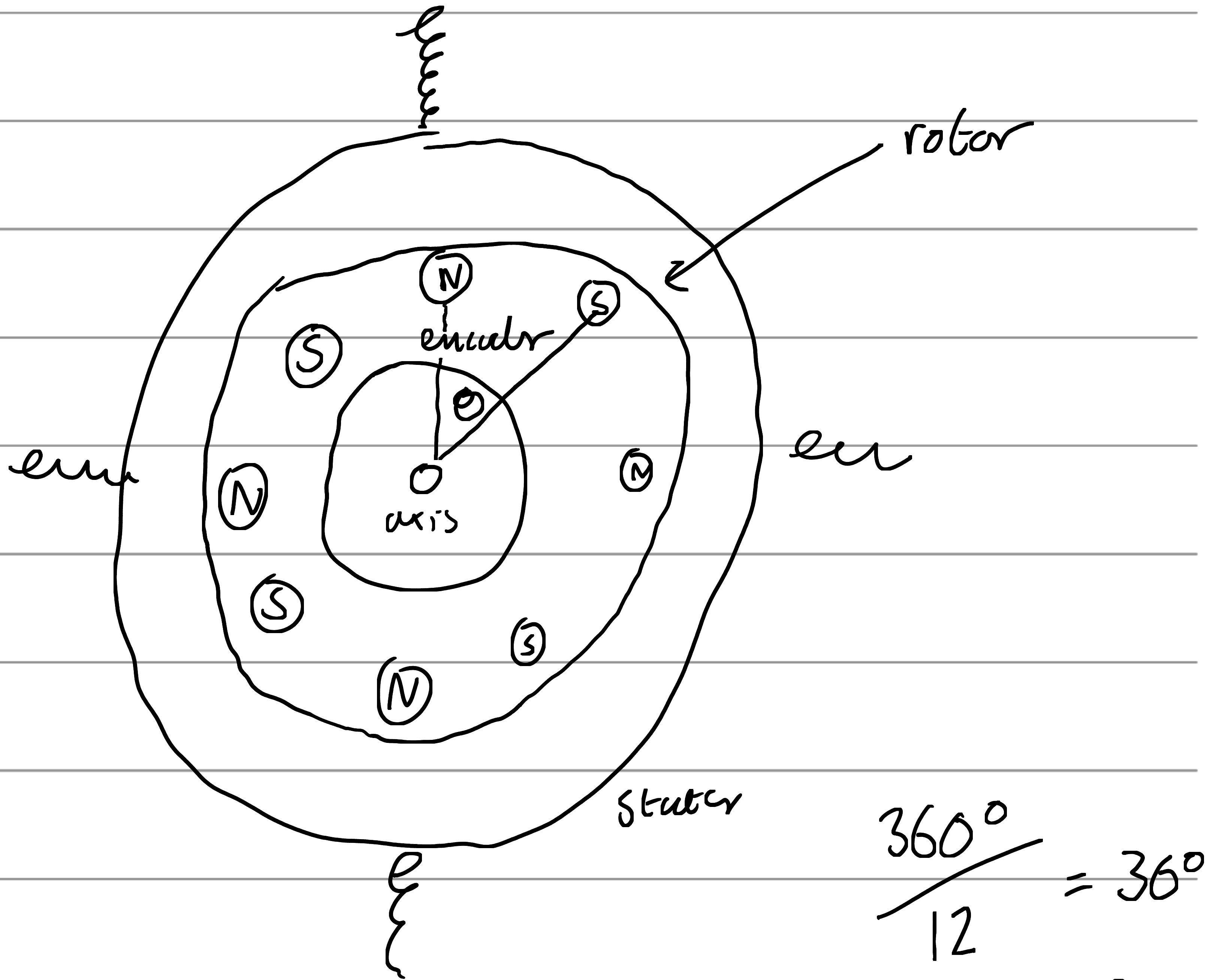
→ Teensy 4.0 work beef marginal, 100/255 steps

What we should consider.

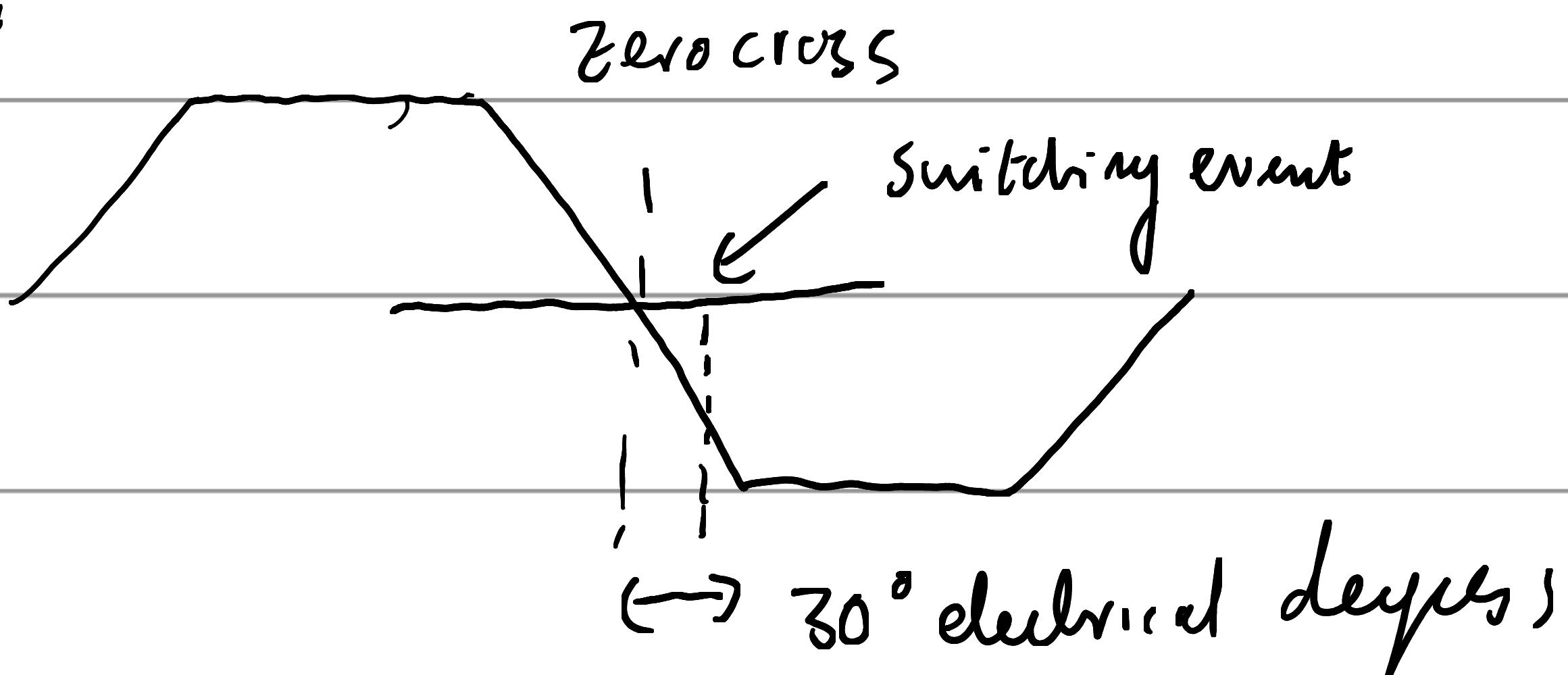
→ BEMF strategy will not work for low rpm, therefore  
we will not know position information reliably.

→ Rotary encoder (absolute)





Phase X:



Bldc 6 electrical steps per electrical revolution

I you will have  $n$  - electrical cycles per mechanical cycle.

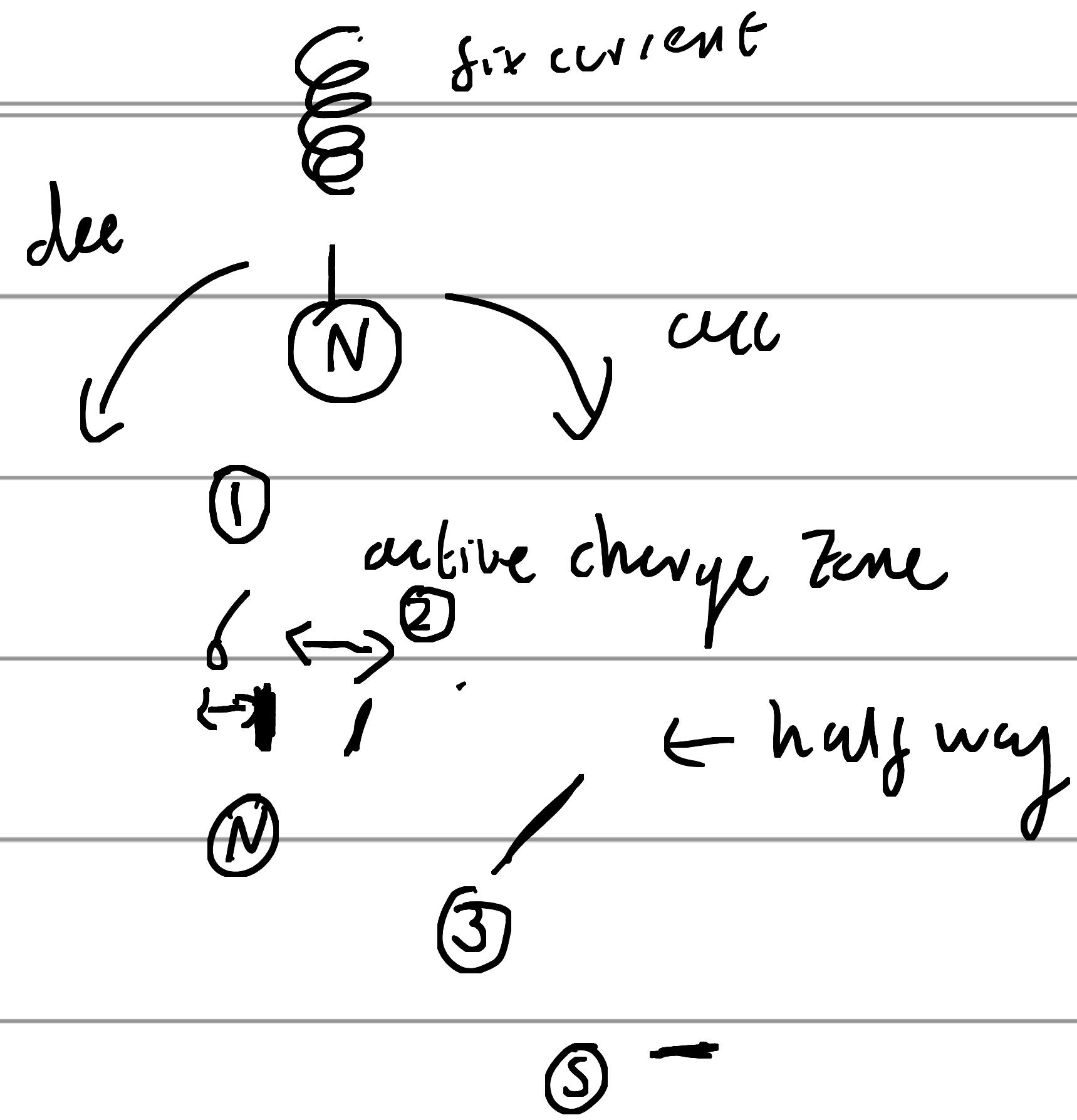
$$\text{Mechanical steps} = \frac{N \text{ poles}}{\text{per mechanical cycle}} \cdot 6$$



$$\begin{aligned} \text{Slots} &= \text{poles} \cdot \text{phase} \\ \text{Slots} &= \text{stator cant.} \end{aligned}$$
$$= \frac{12}{2} \cdot 6 = 36$$

$$[6 \cdot 6]$$

9 coils 12 poles



Step 1 charge coils, Step 2 discharge coils, step 3  
charge cancellation step A  $\rightarrow$  B

processsing that we charge the relevant coil  
when the encoder has indicated precession beyond the  
central point of the pole (above 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)

Arduino

256

reset

Tensy  
40

256, 512, 1024, 8192  
2048

U096

8192

problems

z

switching  
enz

driving  
H bridge

(X)

Positional encoder

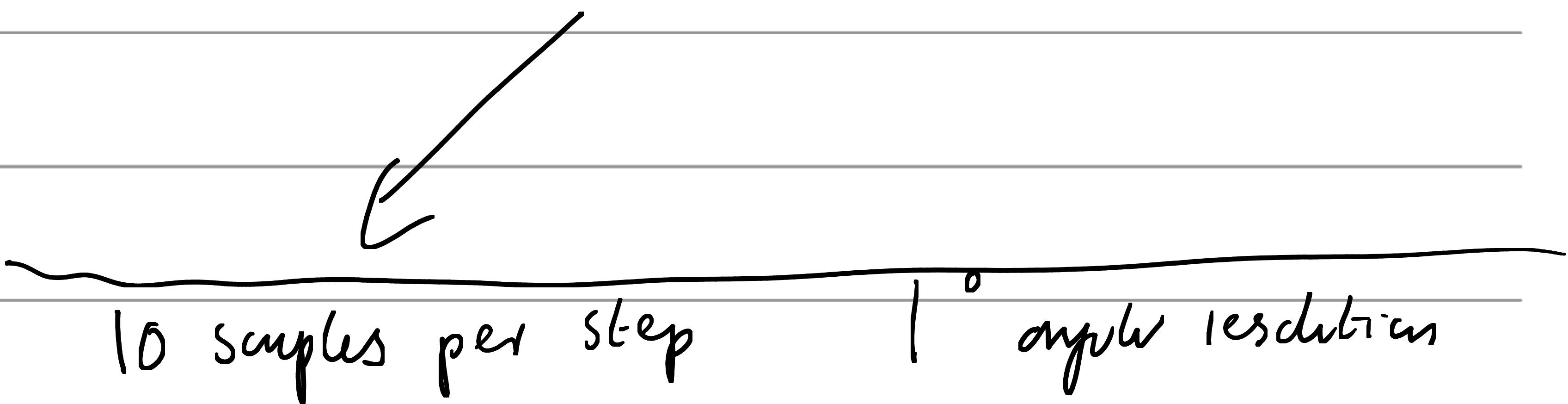
might still be an issue

removes this problem

$360^\circ \rightarrow 36 \text{ steps}$

$10^\circ \text{ per pr step.}$

↑  
mechanical



maybe we want 100 samples per step

$0.1^\circ$

what we want from an rotary encoder

→ absolute position

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

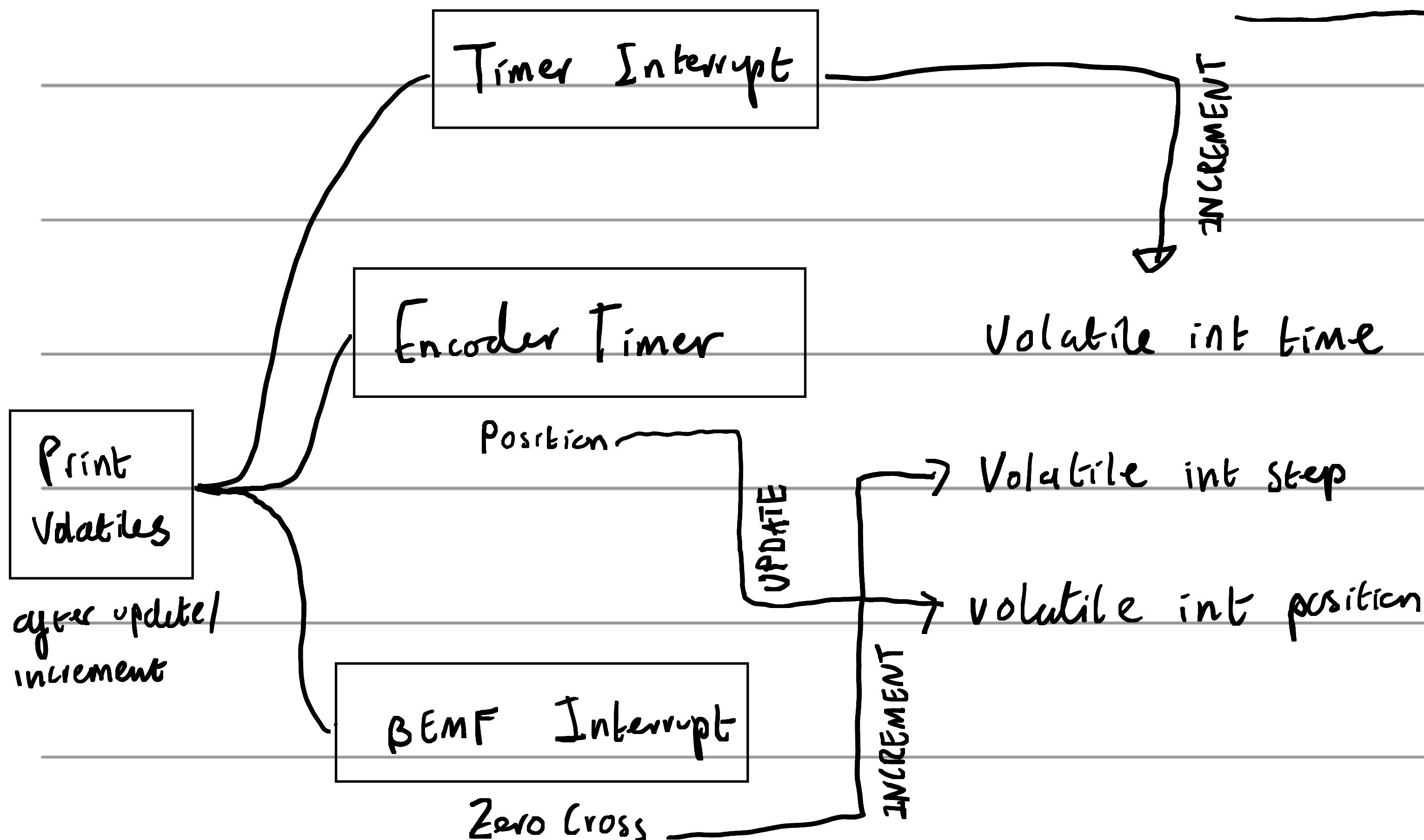
Nice to have

optical (no dust), no interfere bearing, arduino + teensy 40 support.



# CALIBRATION

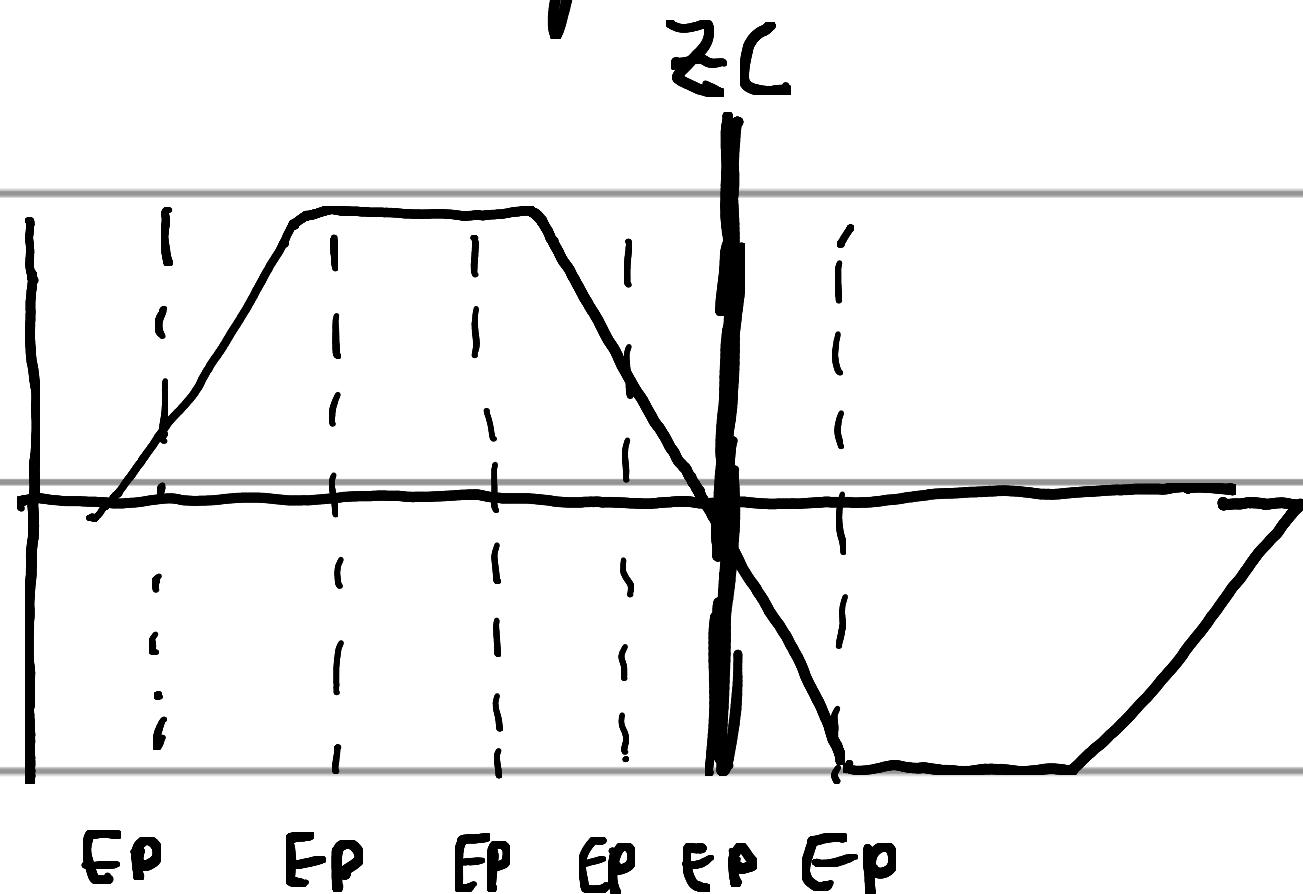
18 / 5 / 22



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

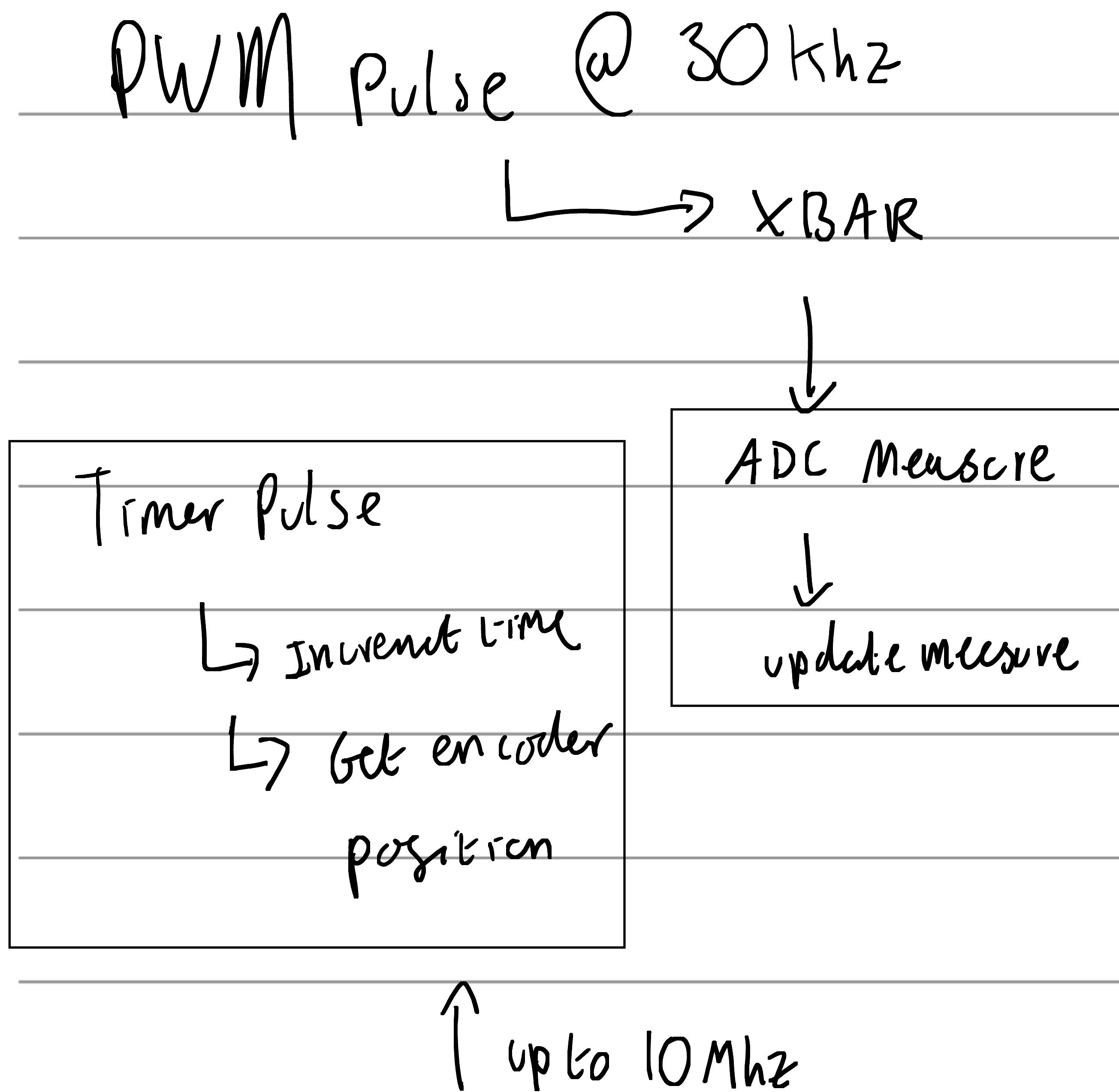
ZC - zero cross

We can find the data via the encoder value the position

I therefore get an average error for the zero-cross angular positions. Ideal switching points are  $30^\circ$  (electrical) after the ZC positions.

One could use the ADC to confirm this.

The way this could work:



21/5/22

Chosen the ADC method as the primary method.

Found 2 good magnetic encoders: Both 3.3 → 5V

→ AS5600 12-bit I<sup>2</sup>C Not for high RPM



x0a6

→ AS5147P 14-bit SPI 28k rpm



↙ 16,384



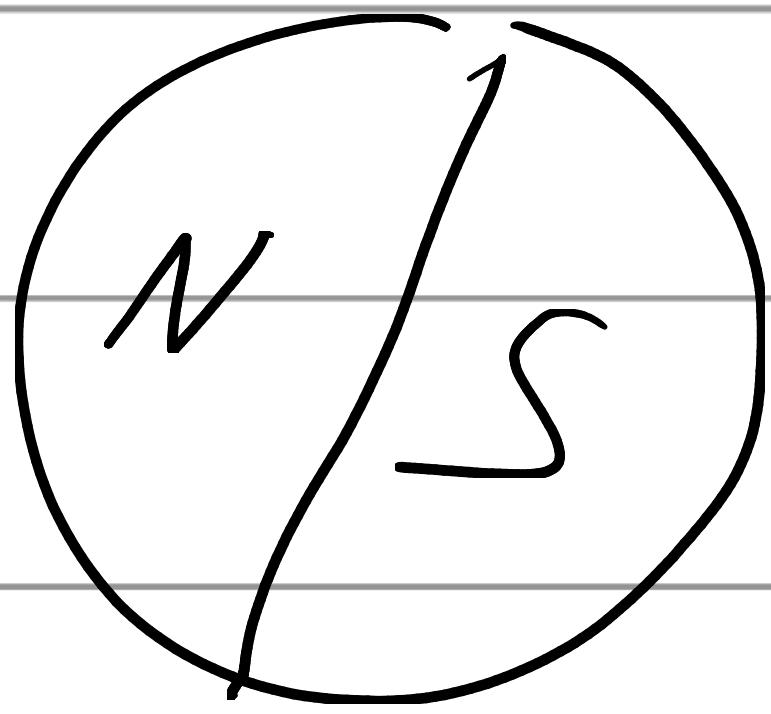
This one is dedicated to motor control

applications like bldc.

AS5147P

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

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



$2^{12}$  4096

3601

4096

6.088°

$0.2 \rightarrow 0.7$

optical

## Jerk model

We will be making position & time measurements

$$P_i = [S_i, T_i]$$

$S$  - displacement angular

Euler method

$T$  - time interval

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

$$\omega_{i+2} = \frac{S_{i+2} - S_{i+1}}{T_{i+2} - T_{i+1}}$$

$$\alpha_{i+2} = \dot{\omega}_{i+2} = \frac{\omega_{i+2} - \omega_{i+1}}{T_{i+2} - T_{i+1}}$$

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

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

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

What are we measuring  $\dot{\theta}_i$ ,  $\ddot{\theta}_i$ ,  $\dddot{\theta}_i(t)$

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

$$\dot{\omega}_i = \ddot{\theta}_i = \frac{d\omega_i}{dT_i} = \frac{d^2\theta_i}{dt^2} \quad \text{angular acceleration}$$

$$\dot{\ddot{\theta}}_i = \dddot{\theta}_i = \frac{d\dot{\omega}_i}{dT_i} = \frac{d^3\theta_i}{dt^3} \quad \text{angular jerk.}$$

Jerk model  $X_i = [\overset{\circ}{\theta}, \overset{\circ\circ}{\theta}, \overset{\circ\circ\circ}{\theta}, \overset{\circ\circ\circ\circ}{\theta}]$  in 1D

White noise  $w(t)$  applying to jerk

$$X = Ax_i + Bw(t) \quad a \text{ is a constant}$$

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

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

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

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

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

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

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

$U(k)$  is the discrete white noise

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

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

$$p_1 = (2 - 2\alpha T + \alpha^2 T^2 - 2e^{-\alpha T}) / 2\alpha^3$$

$$q_1 = (e^{-\alpha T} - 1 + \alpha T) / \alpha^2$$

$$r_1 = (1 - e^{-\alpha T}) / \alpha$$

$$s_1 \sim e^{-\alpha T} = 1 - \alpha T + \frac{\alpha^2 T^2}{2!} - \frac{\alpha^3 T^3}{3!} \dots$$

where  $\alpha T$  is small

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

# 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

$$K_t = P_{t|t-1} H_t^T \left[ H_t P_{t|t-1} \circ H_t^T + R_t \right]^{-1}$$

Q4) Update the observation estimated value

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

S) 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 f, h, Z, K, X+, H, P

Initialisation, 3 measurements to estimate  $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3$

$\hat{\theta}$  taken to be zero initially.

$$\hat{\theta}_3 = M_x(3) \quad \hat{\theta}_3 = \frac{M_{xc}(3) - M_x(2)}{\tau}$$

$$\hat{\theta}_3 = \left[ \frac{M_{xc}(3) - 2M_x(2) + M_x(1)}{\tau^2} \right] \quad \hat{\theta}_3 = 0$$

constant time interval  $\tau$

## Measurement noise

$$M_x(1) = \bar{x}(1) + v(1)$$

$$M_x(2) = \bar{x}(2) + v(2)$$

$$M_x(3) = \bar{x}(3) + v(3)$$

Error in  $\overset{\circ}{\theta}, \overset{\circ}{\theta}, \overset{\circ}{\theta}, \overset{\circ}{\theta}, \overset{\circ}{\theta}$        $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$

$$\begin{aligned}\epsilon_1(3|3) &= \bar{x}(3) - \hat{x}(3|3) = \bar{x}(3) - \bar{x}(3) - v(3) \\ &= -v(3)\end{aligned}$$

$$\epsilon_2(3|3) = \dot{x}(3) - \dot{\hat{x}}(3|3) =$$

## Euler's

$$X_j^o(k+1) = F_j X_j(k) + w_j(k)$$

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

$$X_j = \begin{bmatrix} x & \dot{x} & \ddot{x} & \cdots \end{bmatrix}^T$$

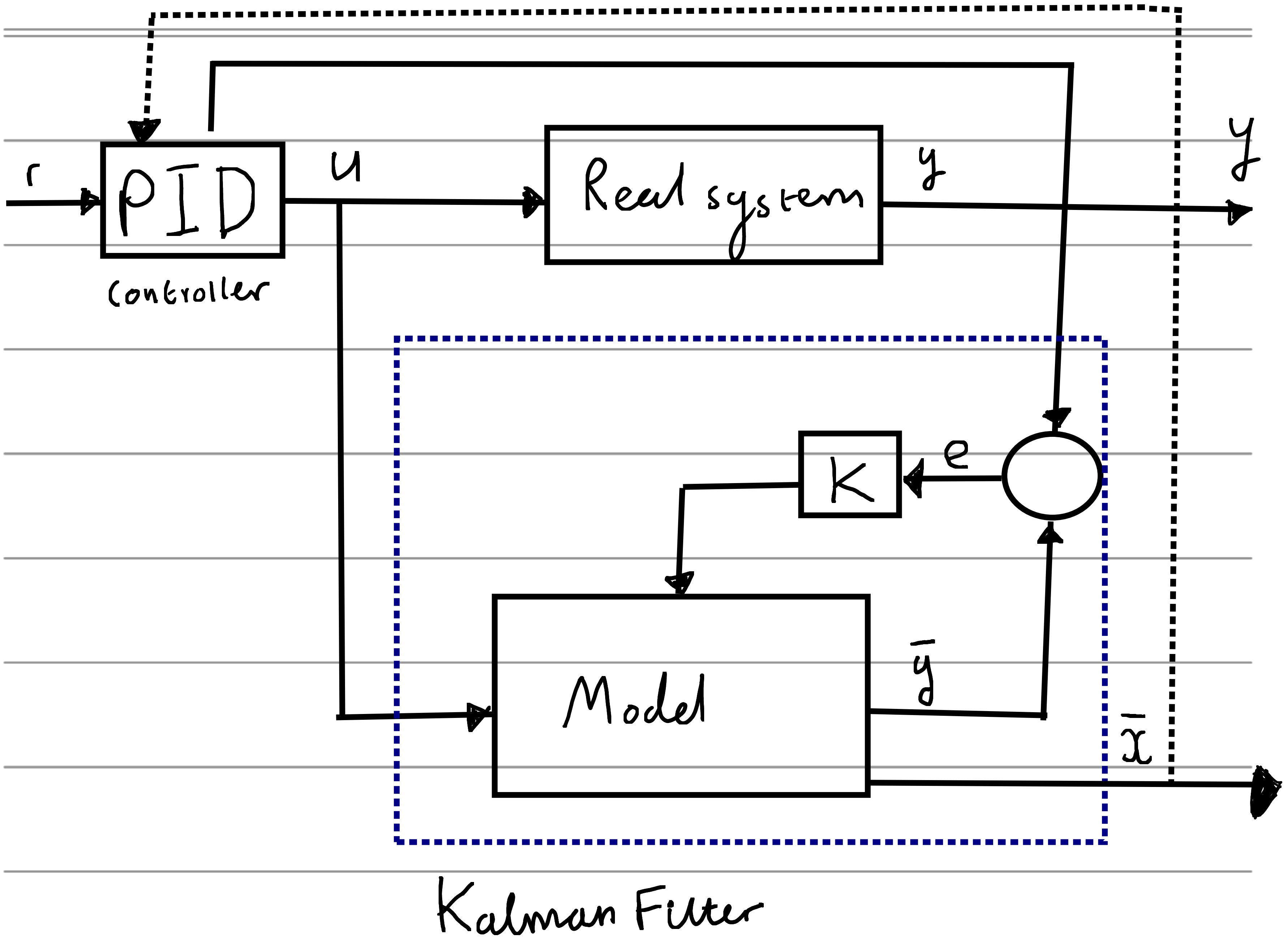
$$w_j = [u_{1j} \ u_{2j} \ u_{3j} \ u_{4j}]^T$$

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

$$Z_j = [M_{dc}]^T$$

$$v_j = [\text{noise } n_x]^T$$

$$F_j = \begin{bmatrix} 1 & T & T^2/2 & P_1 \\ 0 & 1 & T & Q_1 \\ 0 & 0 & 1 & R_1 \\ 0 & 0 & 0 & S_1 \end{bmatrix}$$



Kalman Filter

## Measurement errors

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

$\theta$  degrees  $0 \rightarrow 360$

radious  $0 \rightarrow 2\pi$

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

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

$$\theta = (c \cdot 0.02^\circ) \pm 0.01^\circ$$

error  $\uparrow$  ?  
value

We are measuring time as ticks as an int32 with 31280Hz

$$\frac{1}{31280 \text{ Hz}} = 3.2 \times 10^{-5} [\text{s}] = 32 \times 10^{-6} [\text{s}] = 32 \mu\text{s}$$

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

ticks  $\uparrow$   $(t \pm \Delta t)$

$$\text{With an overflow time of } 2^{16} \times (32 \times 10^{-6}) = 137,439 [\text{s}] \sim 38 \text{ [hrs]}$$

$$\text{Speed error } \omega_i = \frac{d\theta_i}{dt_i}$$

$$\Delta\omega_i = \omega_i \left( \frac{\Delta\theta_i}{\theta_i} + \frac{\Delta t_i}{t_i} \right)$$

CONSTANT

Speed error  $\omega_i = \frac{d\theta_i}{dt}$

$$\Delta\omega_i = \omega_i \left( \frac{\Delta\theta}{\theta_i} + \frac{\Delta E}{E_i} \right)$$

CONSTANT

In reality these are digital measurements

So we have  $Z^{14} = 16,384 \pm 0.5$  Encoder steps angle  
 $S_E \pm \Delta S_E$

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

$$\text{or } \frac{0.5^\circ}{\text{max}} = \frac{1}{z}$$

we have  $Z^{32}$  time max  $\approx 31,280 \text{ Hz} \times 32 \mu\text{s}$   
 $\approx 16 \mu\text{s}$

$$\frac{\Delta S_T}{S_T} = \frac{0.5^\circ}{Z^{32}} = 1.6 \times 10^{-10}$$

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

## Kalman plan

Jerk model try law a model 3<sup>rd</sup> order.

Measurement function  $\rightarrow$  calculate velocity  $\rightarrow$  jerk

Test sets  $\rightarrow$  Pendulum  $\theta, t$

$\rightarrow$  Double pendulum lower half  $\theta, t$

jerk

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

Jerk correlation

$$r_j(\tau) = \sigma_j^2 \exp(-\alpha |\tau|) \quad \begin{matrix} \leftarrow \\ \text{small time} \end{matrix}$$

variance of target jerk

variance of  $w(t)$

$$r_w(\tau) = 2 \alpha \sigma_j^2 \int_0^\tau r_j(\tau') d\tau' \quad Q = 2 \alpha \sigma_j^2$$

state

$$X_{i+1} = AX_i + \beta w_i(t)$$

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

Measurement

Noise vector

$$Z(k+1) = H \tilde{X}(k+1) + V(k+1)$$

$\tilde{X}$   
system measured  
matrix

$$F(\Gamma) = I + \frac{\Gamma^2}{2} P_1$$

$$0 \ 1 \ T \ q_1$$

$$0 \ 0 \ 1 \ r_1$$

$$0 \ 0 \ 0 \ s_1$$

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

low  $\alpha T$

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

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

↑

selecting just 0

$$S_j = 0_6 |$$

$$q = Z d \sigma^2 \cdot \left[ \frac{1}{Zs^2} \right]$$

$$\frac{1}{Zs^2} \quad \frac{1}{36}$$

$\frac{1}{7}$

$$0.01 = Z \cdot 7 \cdot d \cdot \sigma^2$$

$$\frac{0.01}{Z \cdot 7} = \cancel{\sigma^2}$$

Initialise 3 measurements & jerk set to zero.

$\hat{\theta}$

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

$\hat{\dot{\theta}}$

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

$\hat{\ddot{\theta}}$

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

$\hat{\theta}$

$T_2 -$

$\hat{\theta} = 0$

$$M_\theta(1) \quad M_\theta(2) \quad M_\theta(3) \quad |$$

$T_1$

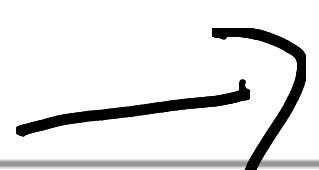
$|$

$T_3$

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

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

timer



Time measurement + static position

Encoder

position + static time ↓  
| periodically  
no dt  
 $= 0$



position  $S(t)$

time  $a, b$

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

position @  $b$   $S(b)$

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

position @ b =  $s(b)$

$$\text{speed @ } b = \frac{s(b) - s(a)}{b-a} = v(b)$$

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

position @ c =  $s(c)$

$$\text{speed @ } c = v(c) = \frac{s(c) - s(b)}{c - b}$$

$$\text{acceleration } a_{all(c)} = \frac{v(c) - v(b)}{c - b}$$

$$\Rightarrow \frac{s(c) - s(b)}{c - a} - \frac{s(b) - s(a)}{b - a}$$
$$\underline{\underline{c - b}}$$

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

Simplify

$$(b-a)(S(c) - S(b)) - (c-a)(S(b) - S(a))$$

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

$$bS(c) - bS(b) - aS(c) + aS(b)$$

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

$$(c-a)(b-a)$$

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

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

$$\text{or } \dots \times a(S(b) - S(c) + S(b) - S(a)) + c(S(a) - S(b))$$

# Trusting polynomial

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

Solution

$$c - b$$

$$a(S(c) - 2S(b) + S(a))$$

$$+ b(S(b) - S(c))$$

$$+ c(S(b) - S(a))$$

$$(a-b)(a-c)(b-c)$$

$$= a(c)$$

$$\overset{\circ}{\theta}, \overset{\circ}{\theta}, \overset{\circ}{\theta} \text{ at } a, b, c$$

Final  $\hat{\theta}(c) = \theta(c)$

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

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

$$+ c(\theta(b) - \theta(a))$$

$$\hat{\theta}(c) \approx \theta$$

$$(a-b)(a-c)(b-c)$$

$\theta, \dot{\theta}, \ddot{\theta}, \ddot{\dot{\theta}}$  from time  $a, b, c, d$

$a, b, c, d$

needed for  $a(t)$

$a \rightarrow b$

$b \rightarrow c$

$c \rightarrow d$

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

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

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

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

$$\begin{aligned} \hat{\ddot{\theta}}(c) &= a(c) = \frac{a(\theta(a) - 2\theta(b) + \theta(c))}{(a-b)(a-c)(b-c)} \\ &\quad + b(\theta(b) - \theta(a)) \\ &\quad + c(\theta(b) - \theta(a)) \end{aligned}$$

combine \*'s with  
 $\Delta$  expression.

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

Replace  $A = \theta(a)$   $C = \theta(c)$

$B = \theta(b)$   $D = \theta(d)$

Wolfram

$$\hat{\ddot{\theta}} = \frac{1}{(a-b)(a-c)(b-c)}$$

$$-a(A - 2B + C) + A_c - B(b+c) + BC$$

$$(a-b)(a-c)$$

$$-b(B - 2C + D) + Bd - ((c+d)_+)_c$$

$$(b-d)(d-c)$$

Simplified:

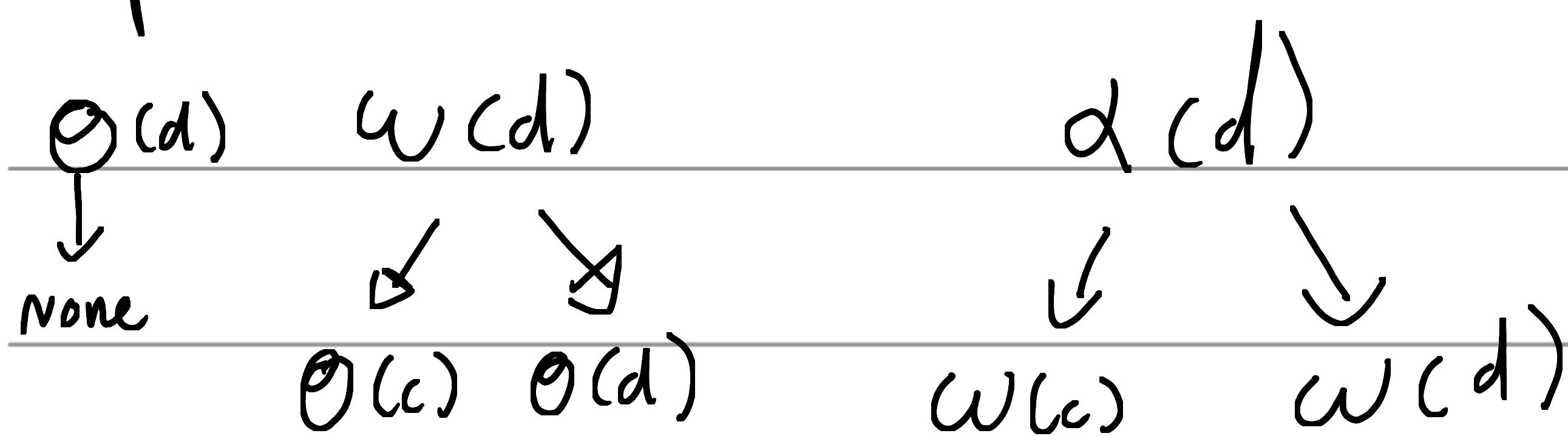
$$\hat{\Theta}(d) = \Theta(d)$$

$$\hat{\Theta}(d) = \frac{\Theta(d) - \Theta(c)}{d - c} = w(d)$$

$$\hat{\Theta}(d) = \frac{w(d) - w(c)}{d - c} = \alpha(d)$$

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

Dependencies First Level



$J(d)$

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

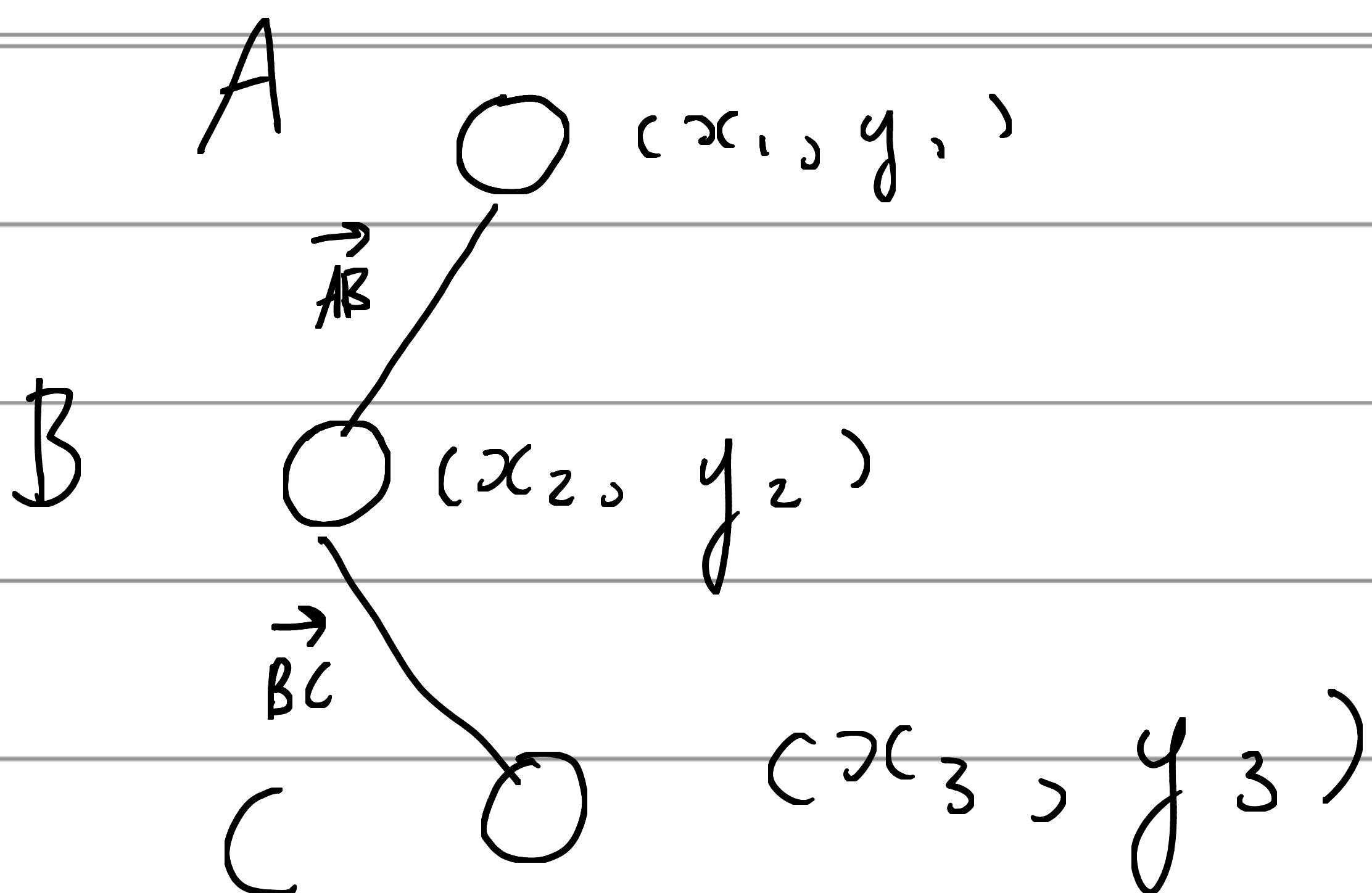
$\downarrow$

$w(c), w(d), ddc$

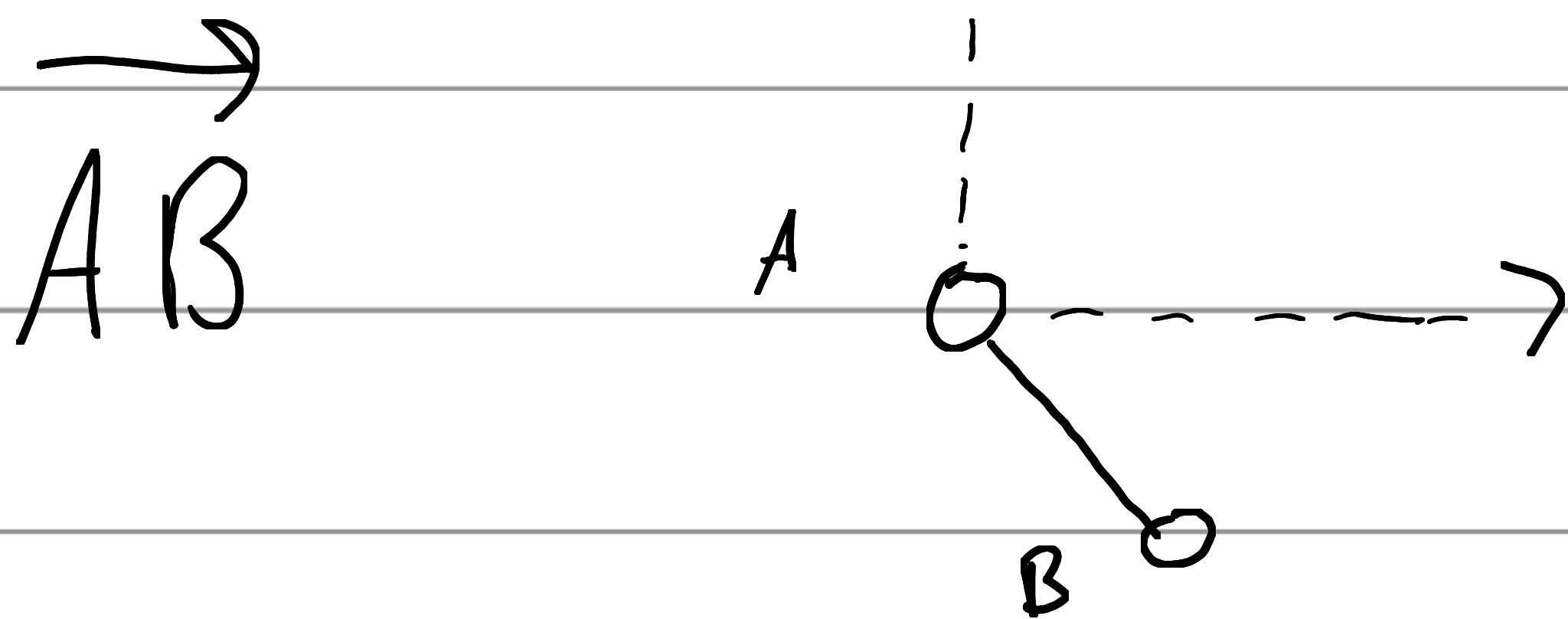
$\alpha(c) \alpha(d)$

$\Theta(c), \Theta(d), ddc$

Incremental scenes more same as have to account  
for  $0 \rightarrow 360 \rightarrow 0$  discontinuity.



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



$$\frac{x}{r} = \frac{x \cos\theta}{r \sin\theta}$$

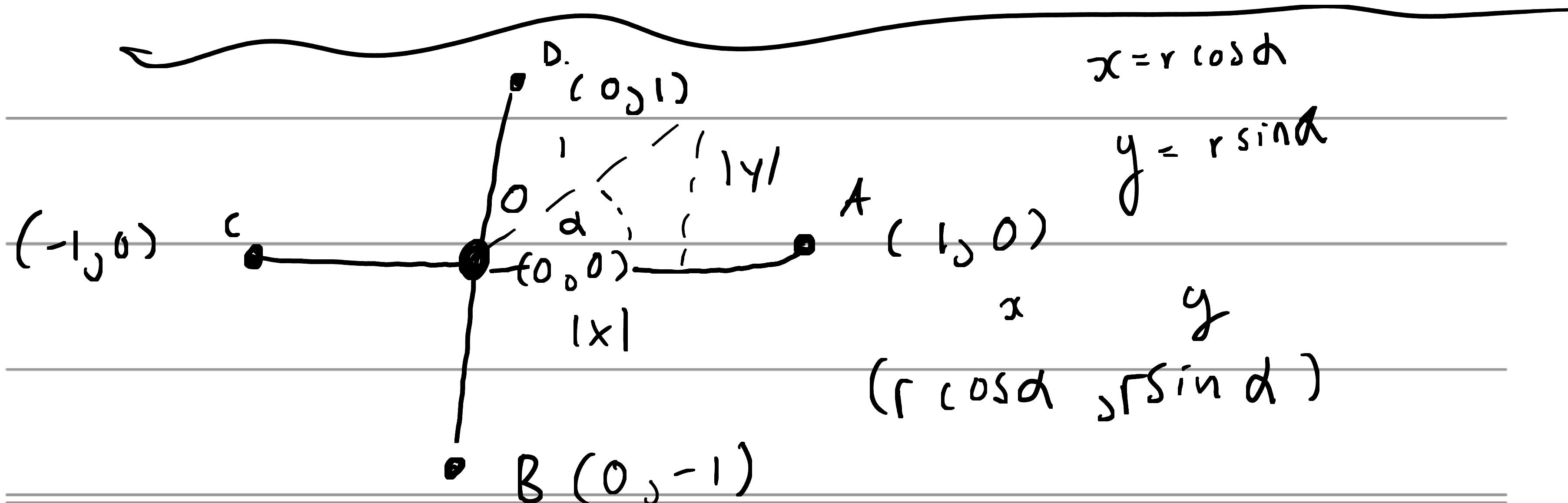
$$A(0,0) = a$$

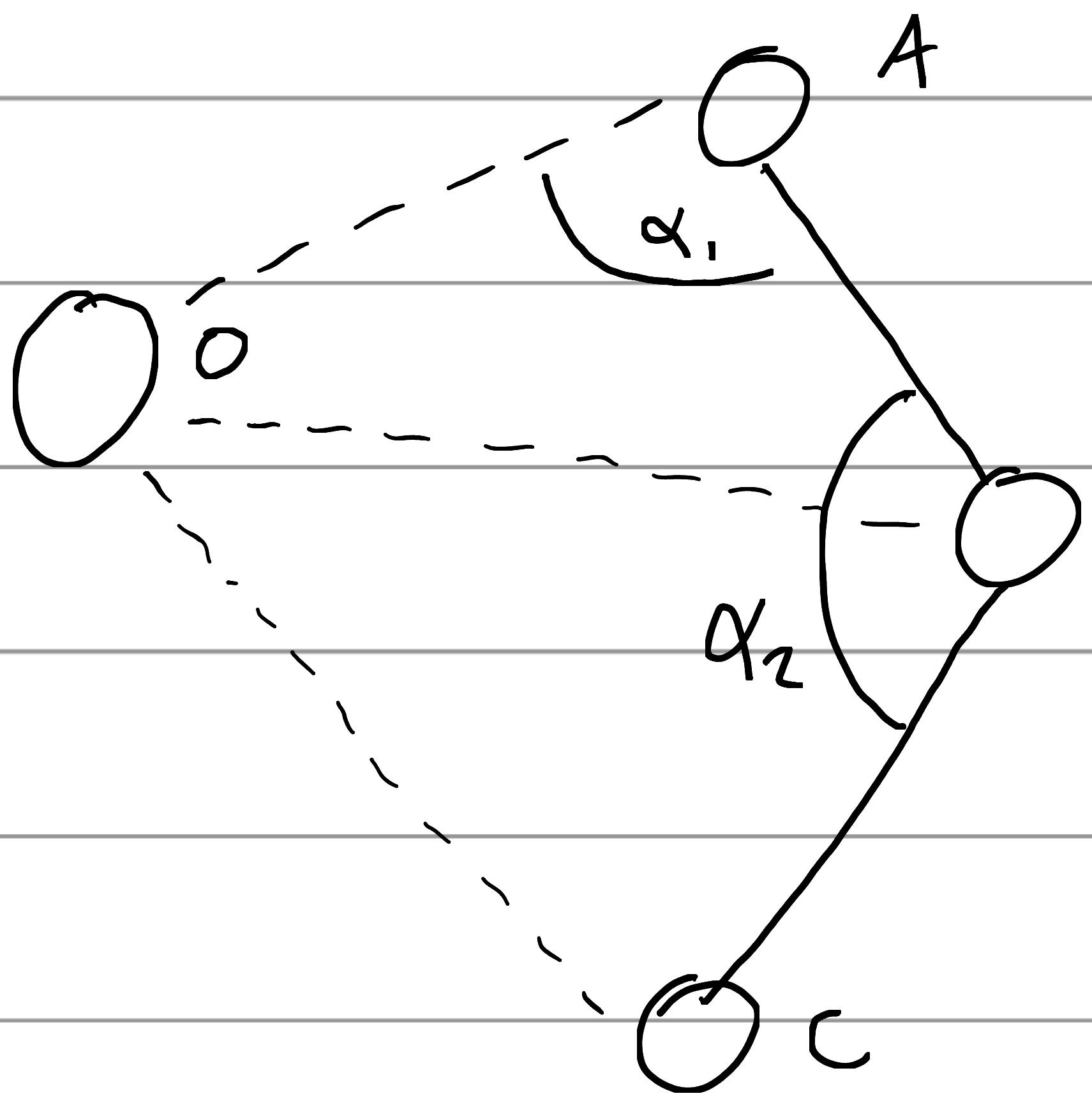
$$\frac{y}{r} = \tan\theta$$

example

$$B(0, -1) = b \quad |a| = 0$$

$$|b| = 1$$



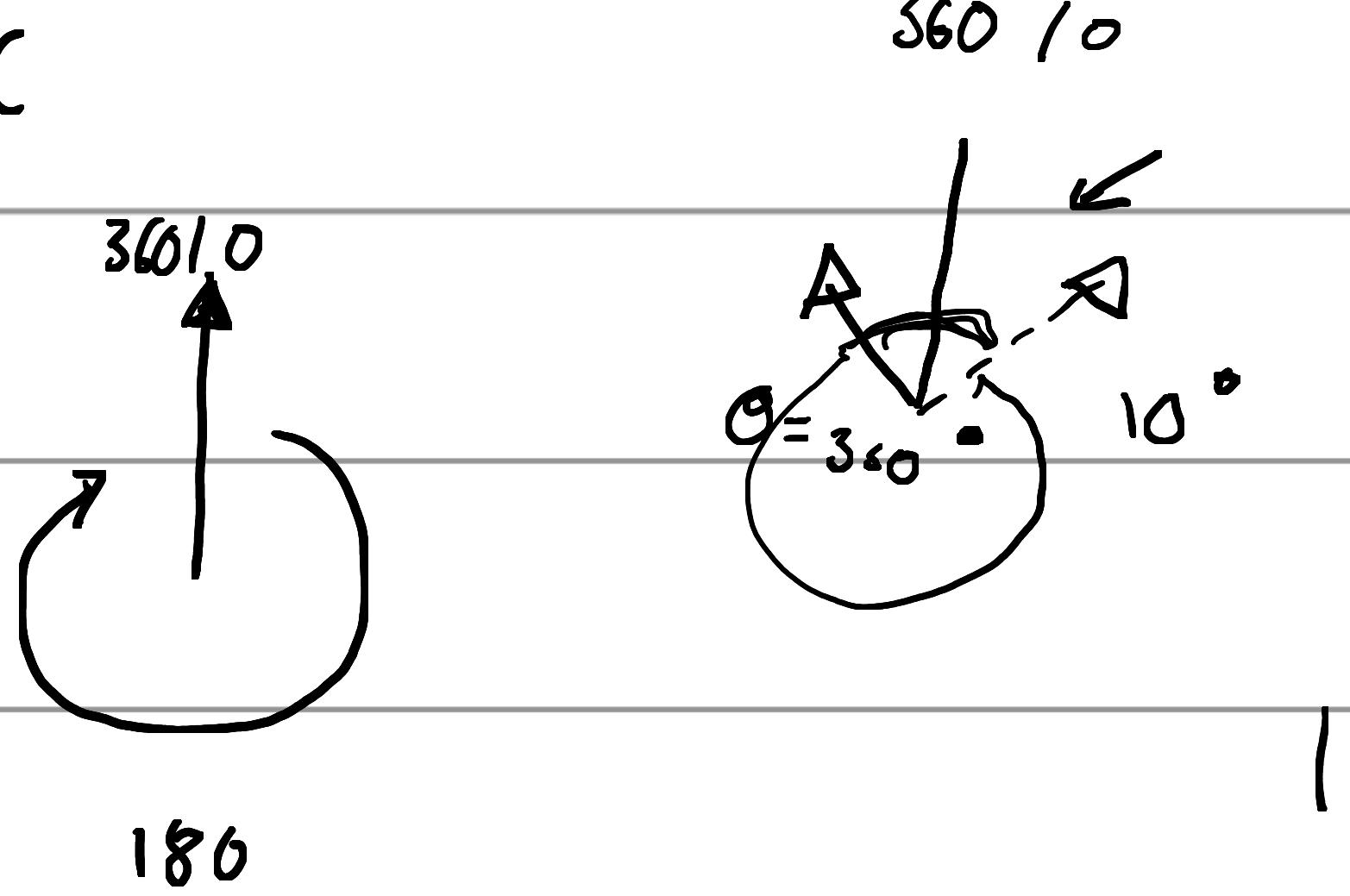


$$\vec{OA} \cdot \vec{OB} = |\vec{OA}| |\vec{OB}| \cos \alpha_1$$

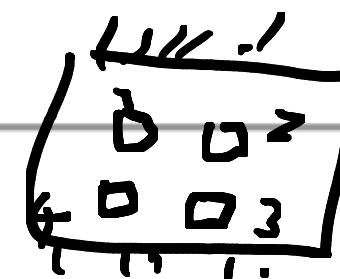
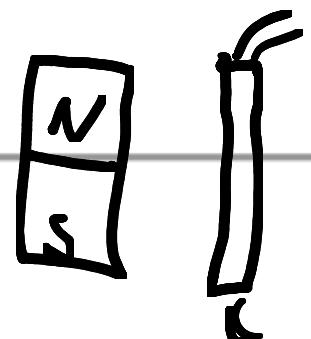
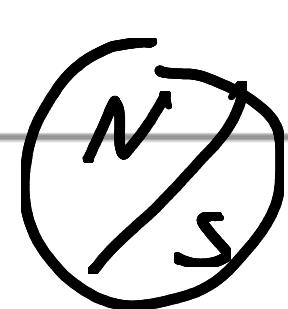
$$\vec{AB} \cdot \vec{AC} = |\vec{AB}| |\vec{AC}|$$

$$\cos \alpha_2$$

Example  $\delta(0,0) AC$



10



$O \rightarrow Z^{+4}$

o

↓

$\sim 16,000^{\circ}$

27/5/22

→ Pendulum angle code + modular turns done. ✓

→ charting of state estimates graph bokeh ✓

→ Need real-time hw timer for dt python      } done ... need  
dt ~ 0.00228      } scripts for user ✓

Next

Repo 1

→ Encoder SPI interface + Jerk estimate      } Need sensor  
+  
basic rotation  
assembly, manual  
rotation

Repo 2

→ ADC -> Encoder

Repo 3

→ LMP + Encoder

serial plot check

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

→ 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.

z81S1Z2

→ Did draft of plotter & created test script

→ venv setup if ! env

To do npm scripts → pip freeze / install

→ Activate

→ run bokeh

29/5/22

ADC pwm frequency, recover old work, New  
hardware timer integer counter, square wave to map the  
zero crossing forwards/backwards. longest continuous  
run with visual neutral above 0.

Positioning, Kalman C++ version. Arduino mpu6050 library.

ADC + time measurement, + ZC on host



@Pwm hz

30/5/22

Kalman

$$P = \begin{bmatrix} \sigma_x^2 & \left( \sigma_x^2 / T + \frac{\sigma_x^2}{T^2} \right) & 0 \\ \sigma_x^2 / T & 2\sigma_x^2 / T^2 & \frac{3\sigma_x^2}{T^3} \\ \sigma_x^2 / T^2 & \frac{3\sigma_x^2}{T^3} & \frac{6\sigma_x^2}{T^4} \end{bmatrix}$$

$$\frac{6}{6} \leq \frac{\sigma_j^2 T^2}{\sigma_j T} \leq \frac{\sigma_j^2}{\sigma_j}$$

Measurements

$$P \quad 6 \times 6 \quad H \quad 2 \times 6$$
$$\begin{bmatrix} x_{1,4} \\ x_{1,4} \end{bmatrix} \quad S = H P H^T + R$$
$$2 \times 6 \quad 6 \times 6 \quad 6 \times 2 \rightarrow 2 \times 2$$

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

$$S = H P H^T + R$$

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

$$\begin{matrix} & & & & \\ 1 \times 4 & 4 \times 4 & 4 \times 1 & 1 \times 1 & P[0] \\ & & & & \text{row} \end{matrix}$$

$$K = \underbrace{P}_{4 \times 4} \underbrace{H^T}_{4 \times 1} \times \underbrace{S^{-1}}_{1 \times 1}$$

31 | size<sup>2</sup>

→ Finished prototype of Kalman code in its branch.

seems to work well with regard 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

Predict state forward (involves  $X$ )  $\rightarrow$  double

Predict error forward can be float Matrix!

Calculate Kalman gain can be floats Nonmatrix

Calculate innovation (involves  $X$ )  $\rightarrow$  double

update covar matrix can be float Matrix

Is float  $\rightarrow$  double worth it?

(calculating  $S$  only needs to extract one matrix value)

We care about the bus not overflowing /

getting out of sync

Custom code for X prediction with F (double)

Multivit code for P error prediction / 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.

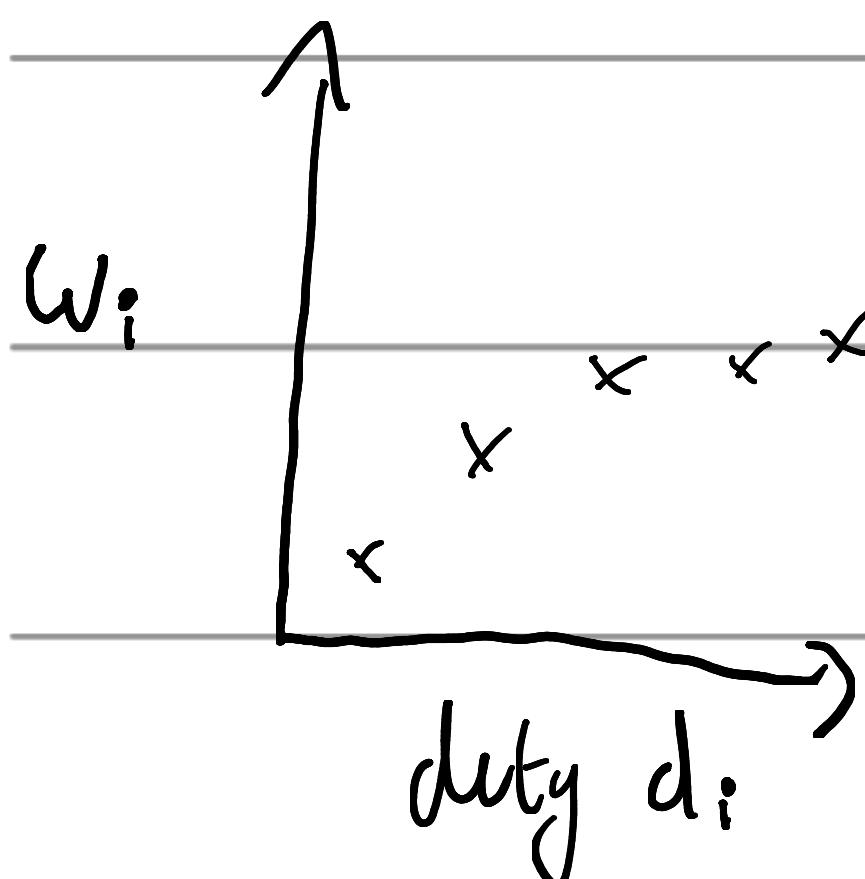
$$T_M = m \left( \alpha + \int_{T_1}^{T_2} J dt \right) \quad \text{motor torque}$$

@ Equilibrium

$$0 = T_M - T_r \quad \text{motor resistance / load}$$

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

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



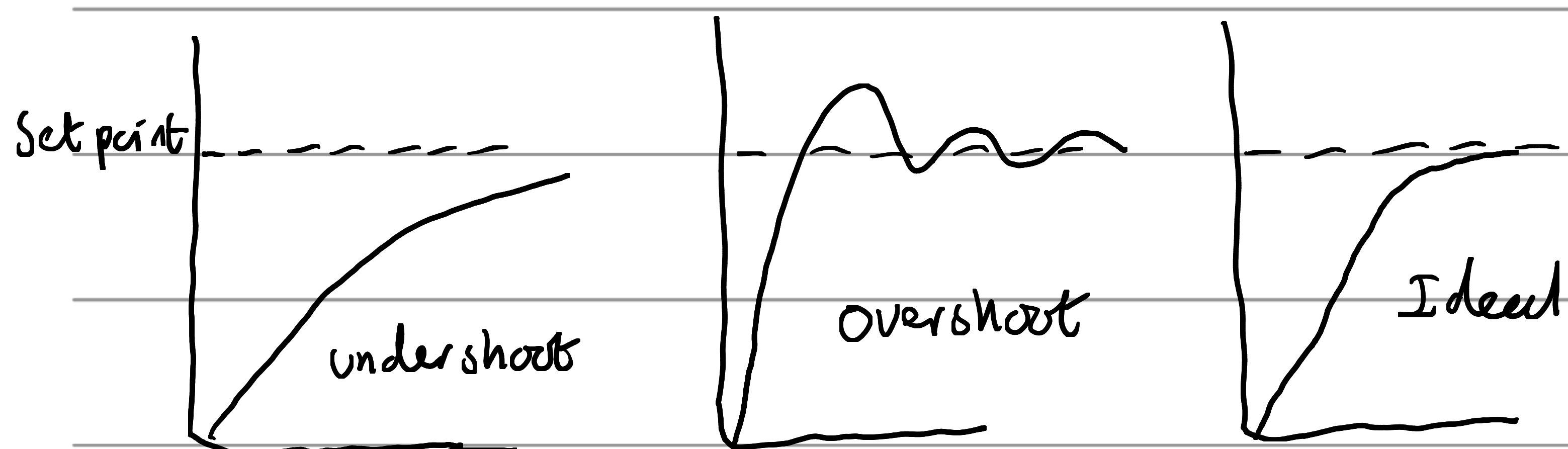
11/6/22

## Tuning:

- 1) Set all gains to zero
- 2) Increase gains until speed fluctuates around set point
- 3) Set P(D) to minimize oscillation
- 4) Raise I until you reach setpoint & oscillations dampen.

$$f_{(s)} = \frac{L}{HAC} e^{Bs}$$

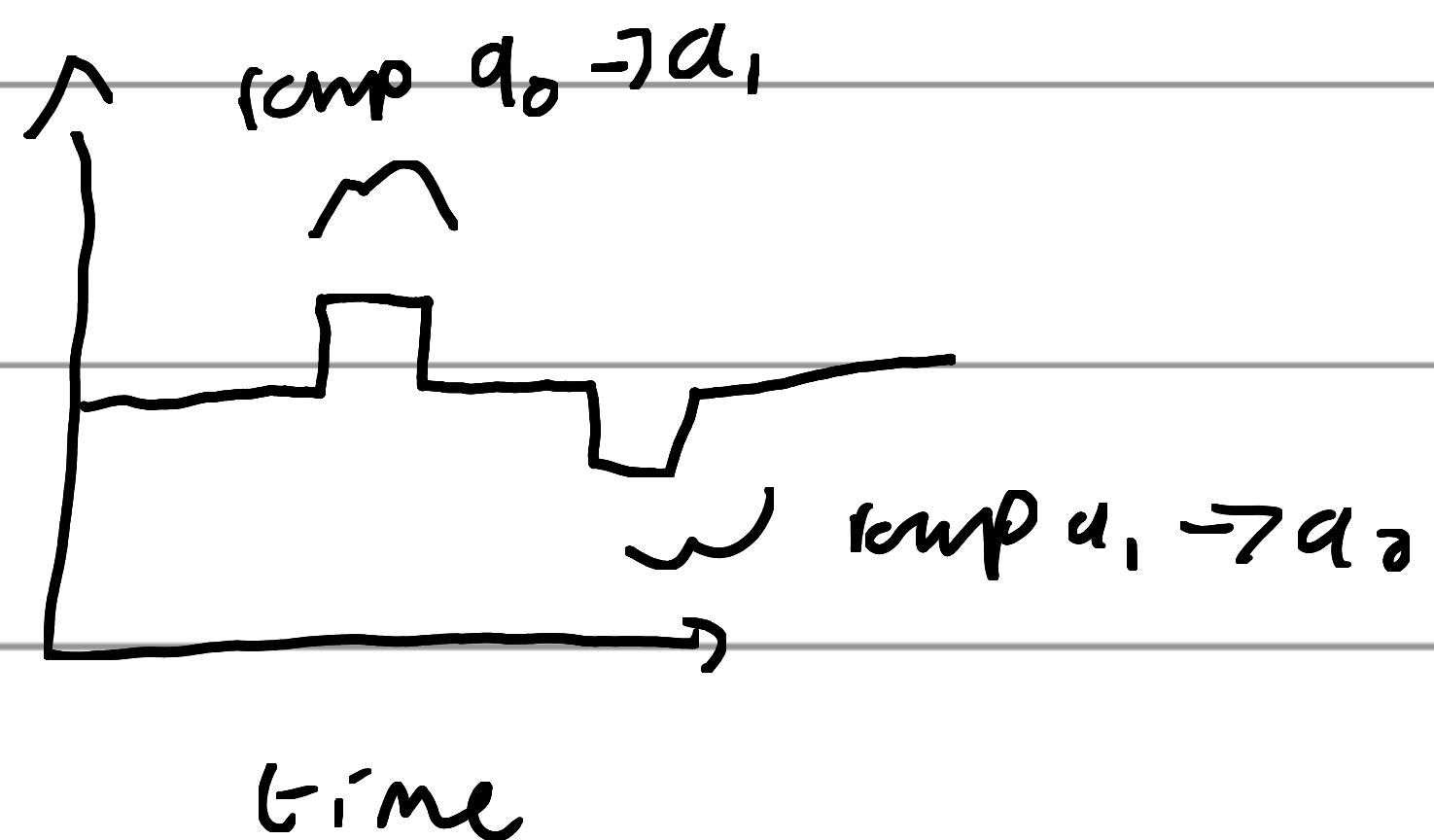
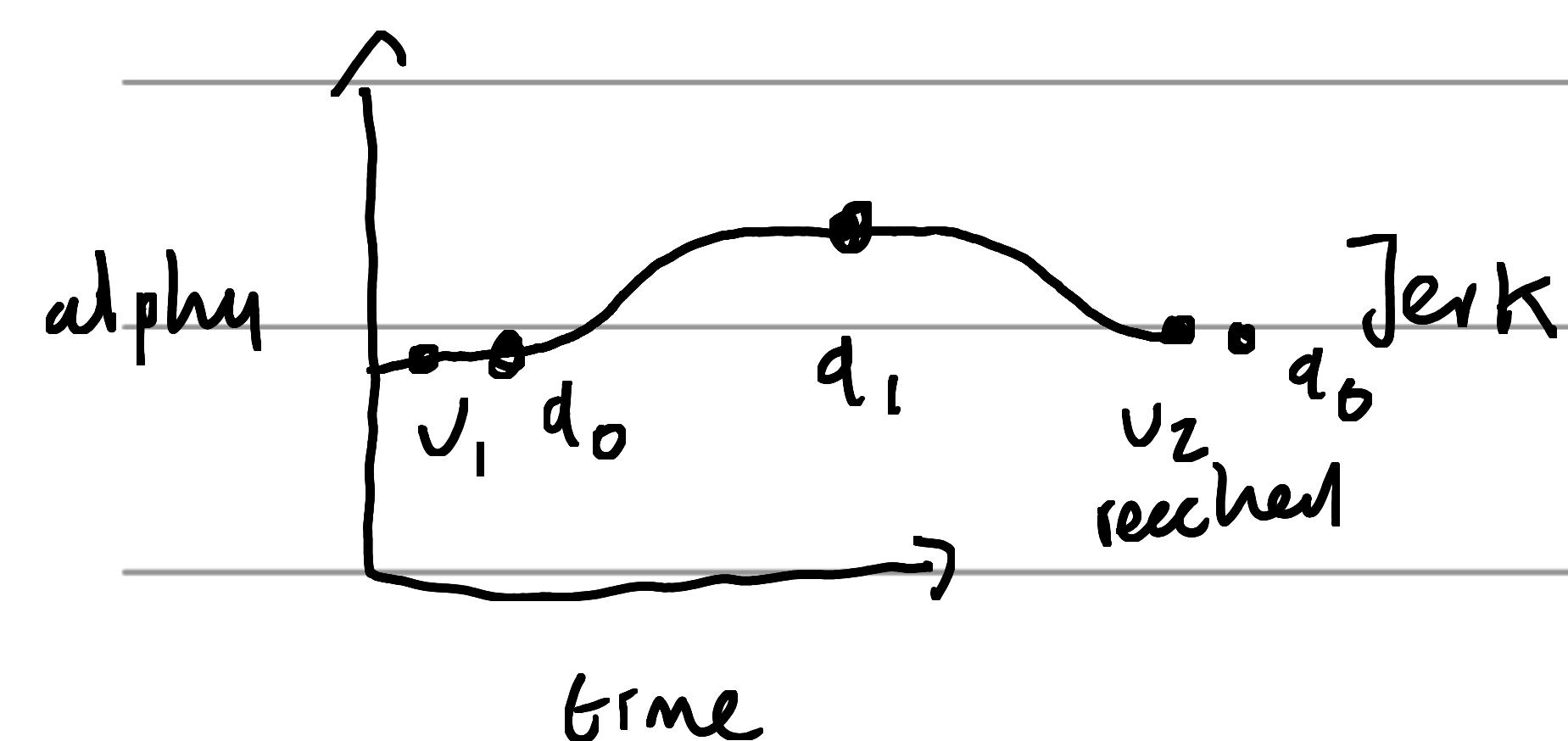
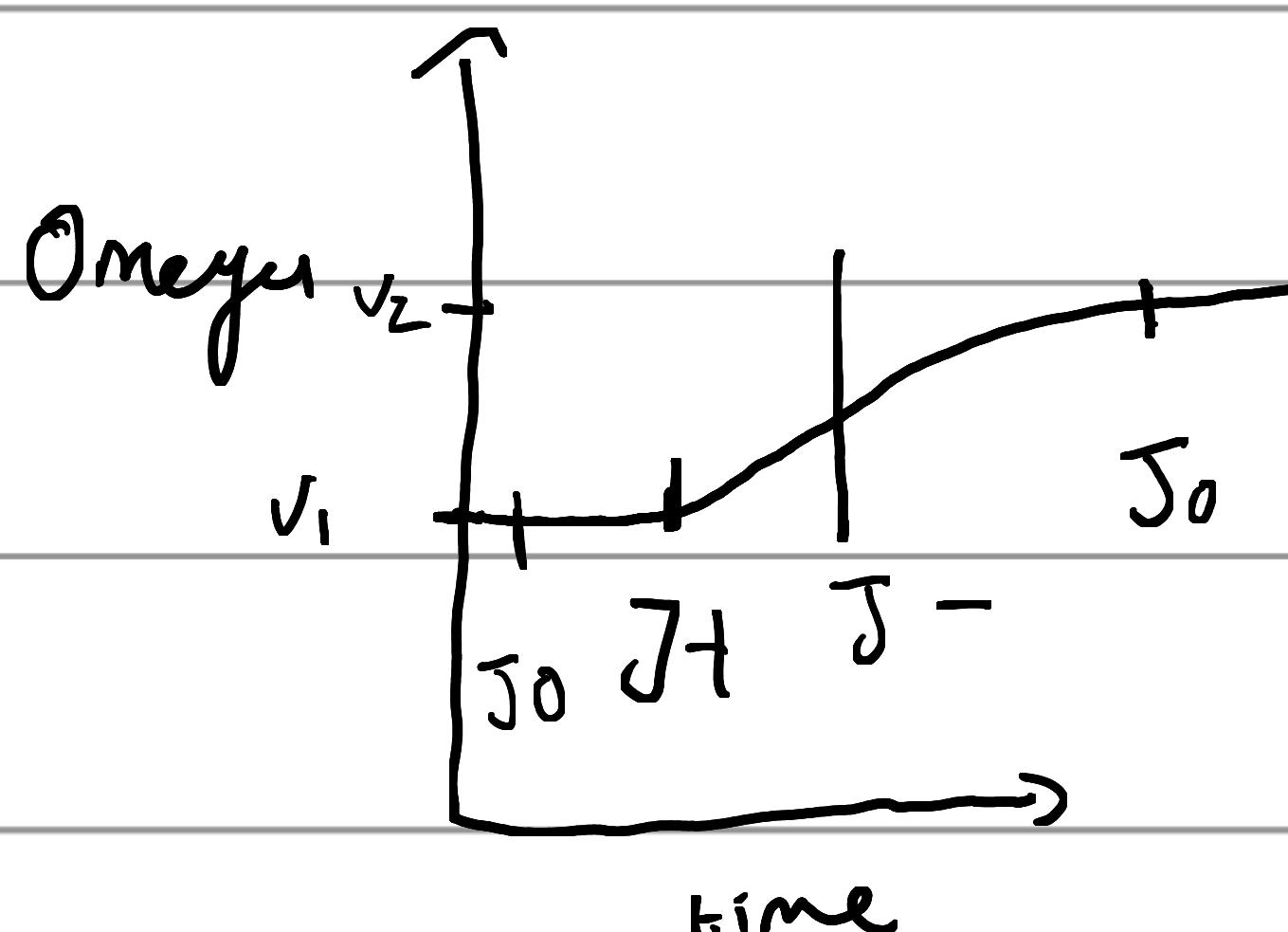
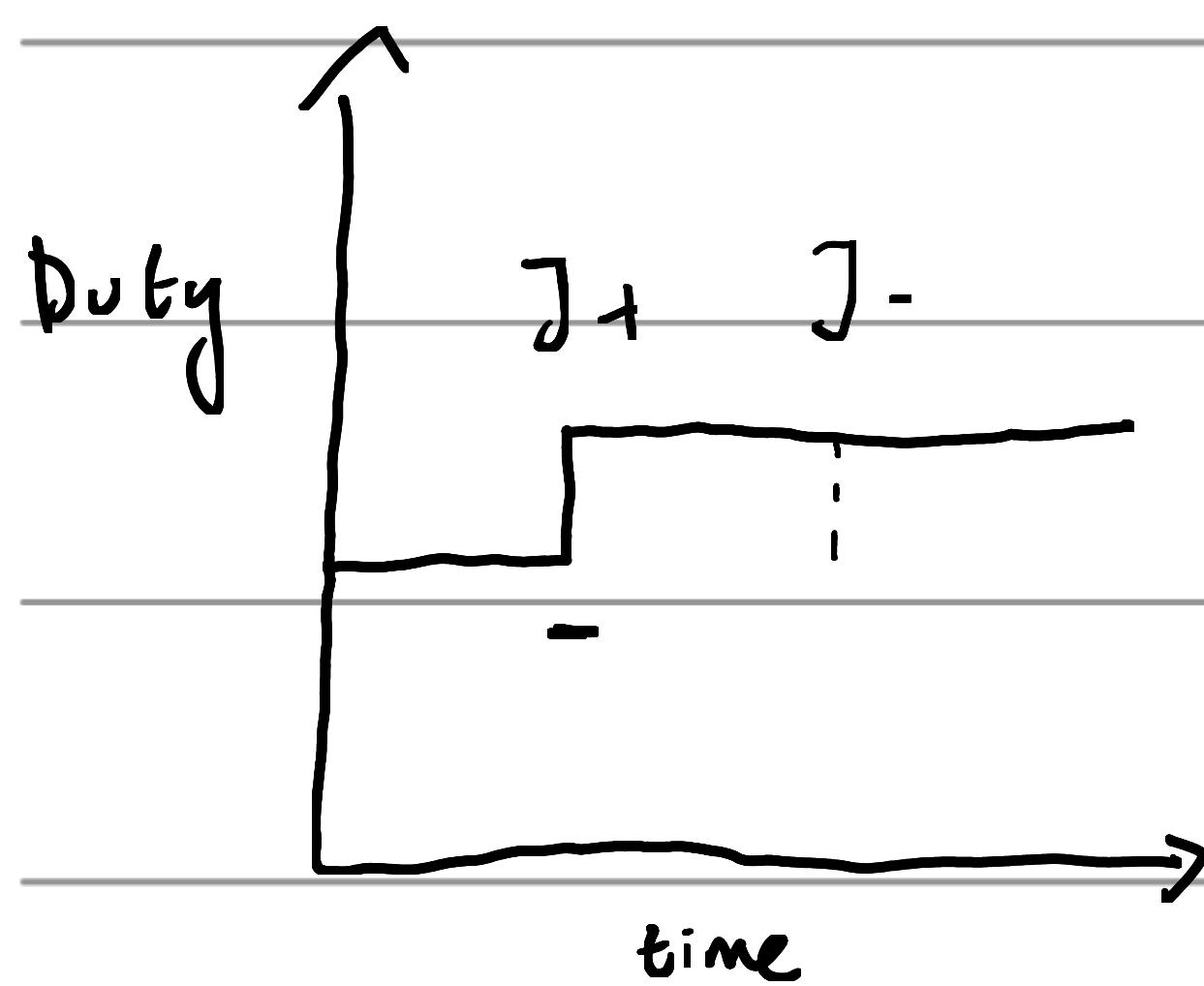
What can a PID controller do?



## 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 (theory)
- Some process responses increase infinitely when applying a step, this is called an integral response.  
I think Jerk theory is one of these.

# AC motor system. (change duty)



$$K_p \sim e^{-\theta / (t + T_p)}$$

$T_p$  - process lag (time constant)

$\theta$  - process delay

$K_p$  - process gain

$\omega, \alpha, J$

$D(\tau)$

friction

$\omega(\tau, D(\tau))$

$F(\omega(t))$

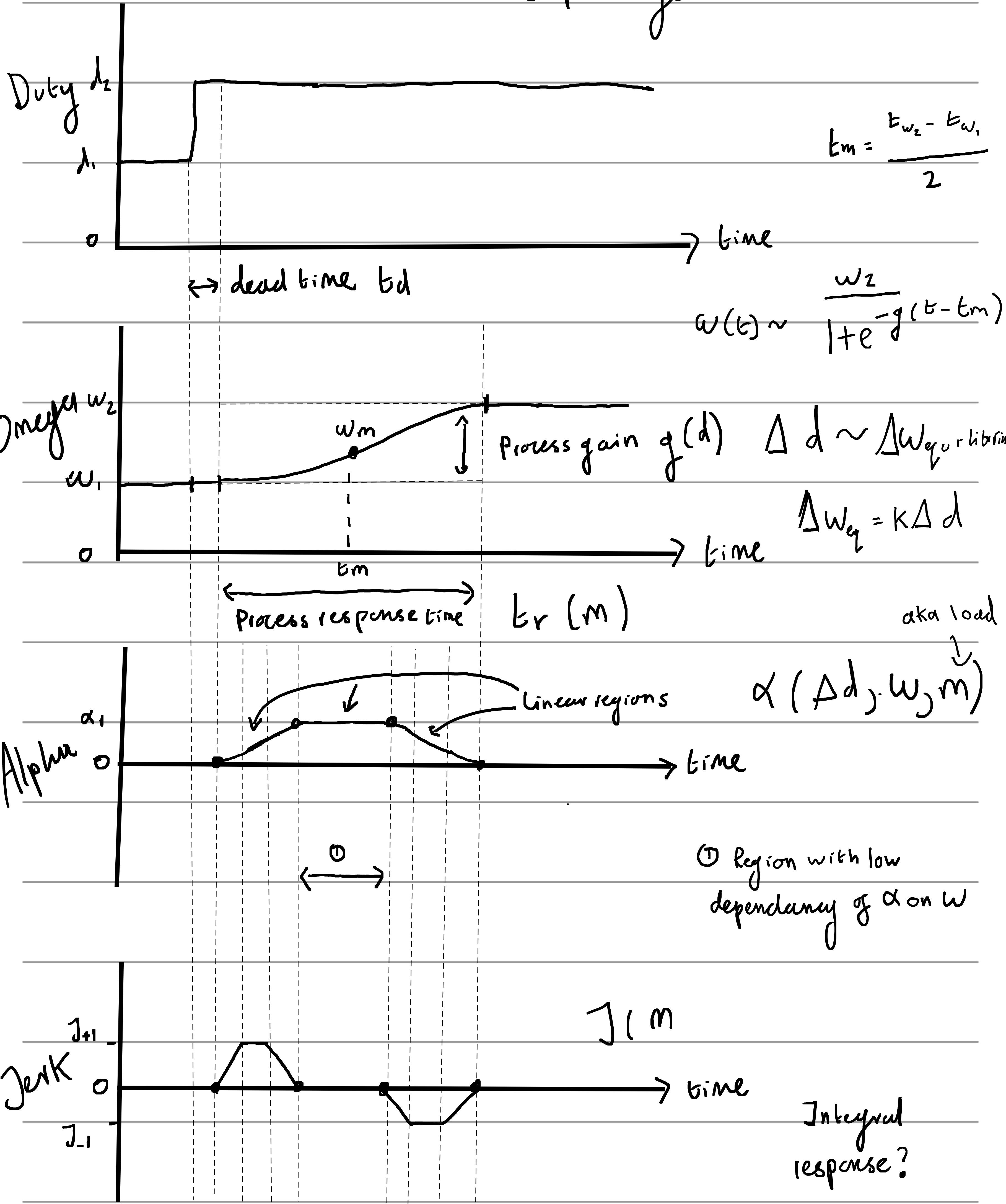
it's not just friction

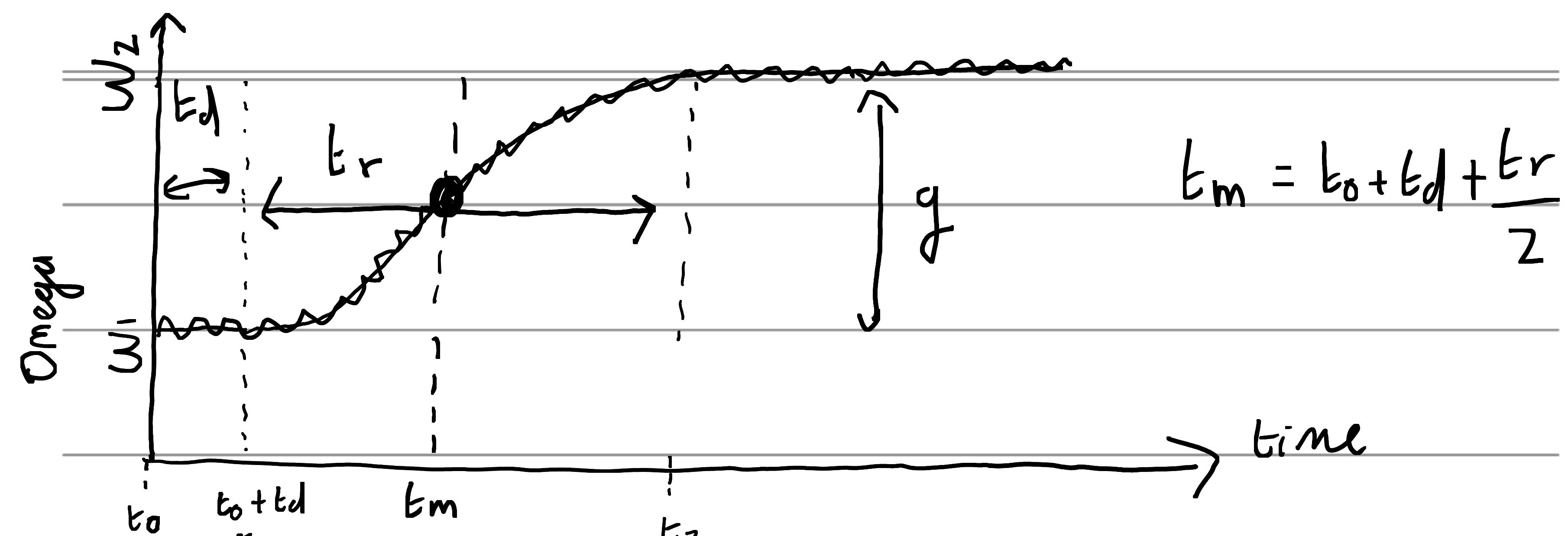
$\dot{F}(\tau, \frac{\partial D(\tau)}{\partial \tau}, F(\omega(t)))$

$q(\omega(t))$

$J$

# AC Motor Step change





Equilibrium Omega ~ Duty

$$\bar{\omega}_{1\text{eq}} \sim d_1, \quad \bar{\omega}_{2\text{eq}} \sim d_2$$

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

K can be found by

doing a step change &

$$\bar{\omega}_{1\text{eq}} = Kd_1, \quad \bar{\omega}_{2\text{eq}} = Kd_2$$

measuring  $\Delta\omega$  after

a long time  $\gg T_r$  by

measuring  $\bar{\omega}_1$ ,  $\bar{\omega}_2$   
averages.  $(d, t, \omega)$

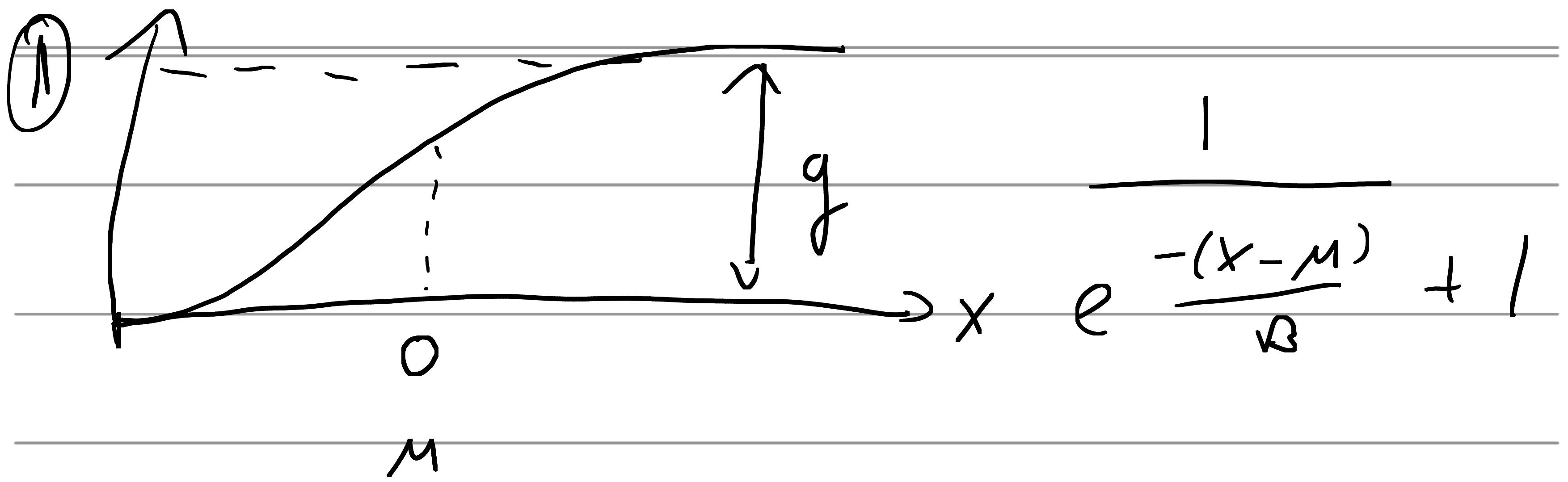
$$g = K(d_2 - d_1) \quad \text{gain}$$

$$t_m = \frac{t_2 - t_0 - t_d}{2} \quad t_0 \text{ is where duty changes}$$

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

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

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



$\beta$  is currently a guess!

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

P B propylene

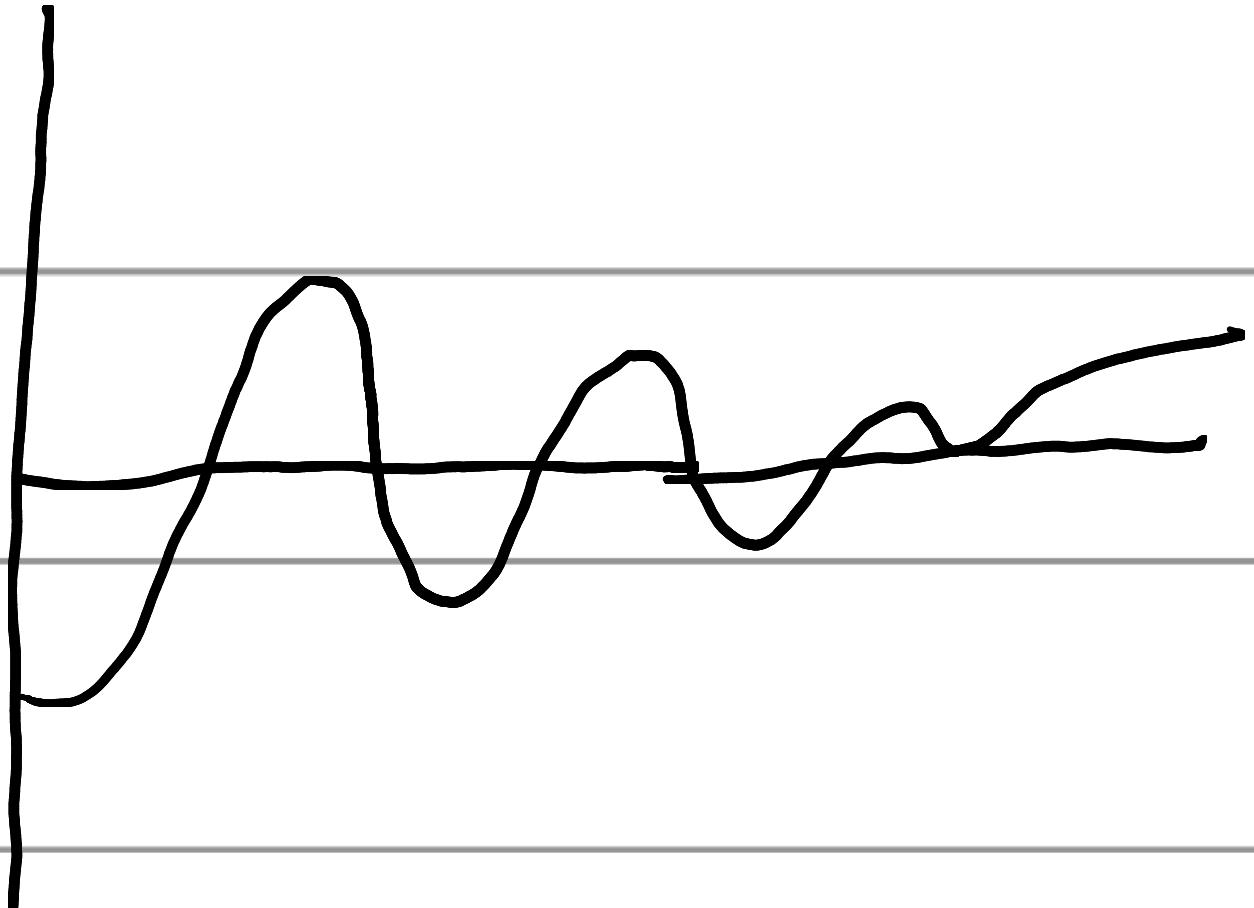
P

Td fine derivative

D

T<sub>i</sub> integral

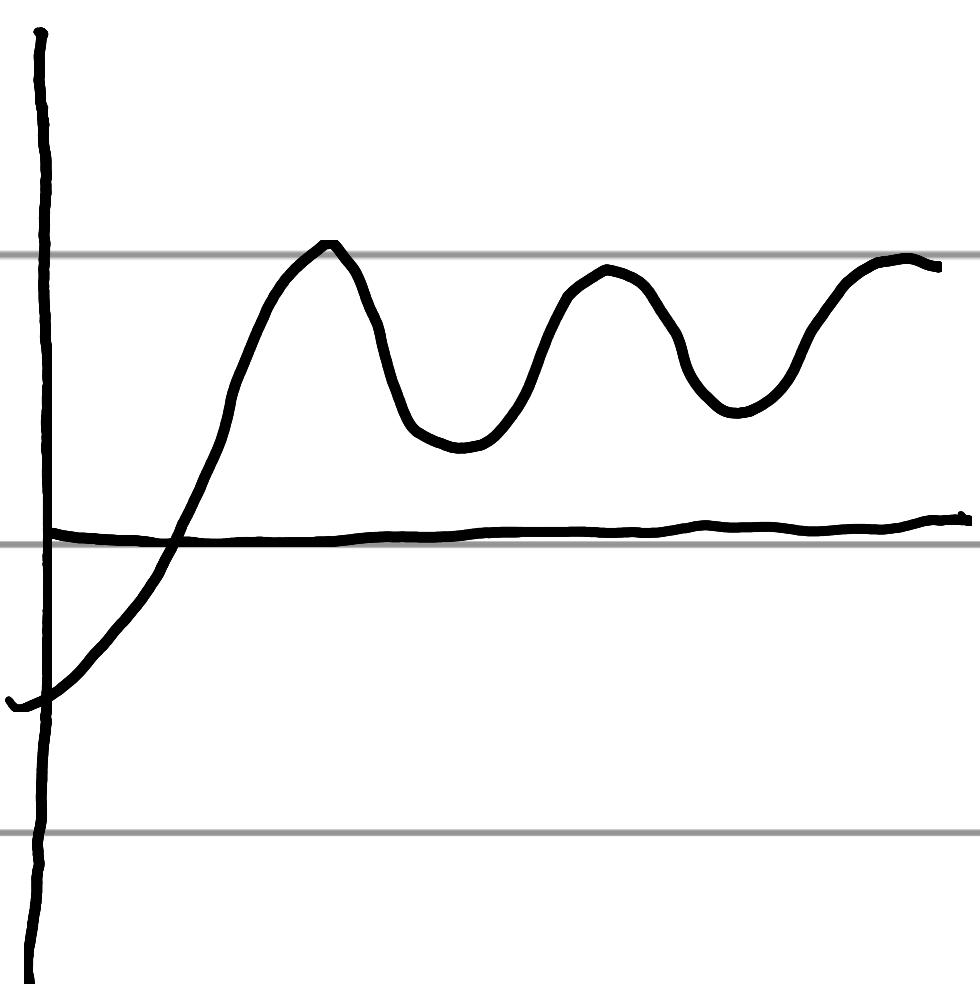
I



When oscillating  
with a small overshoot

→ decrease  $K_I$

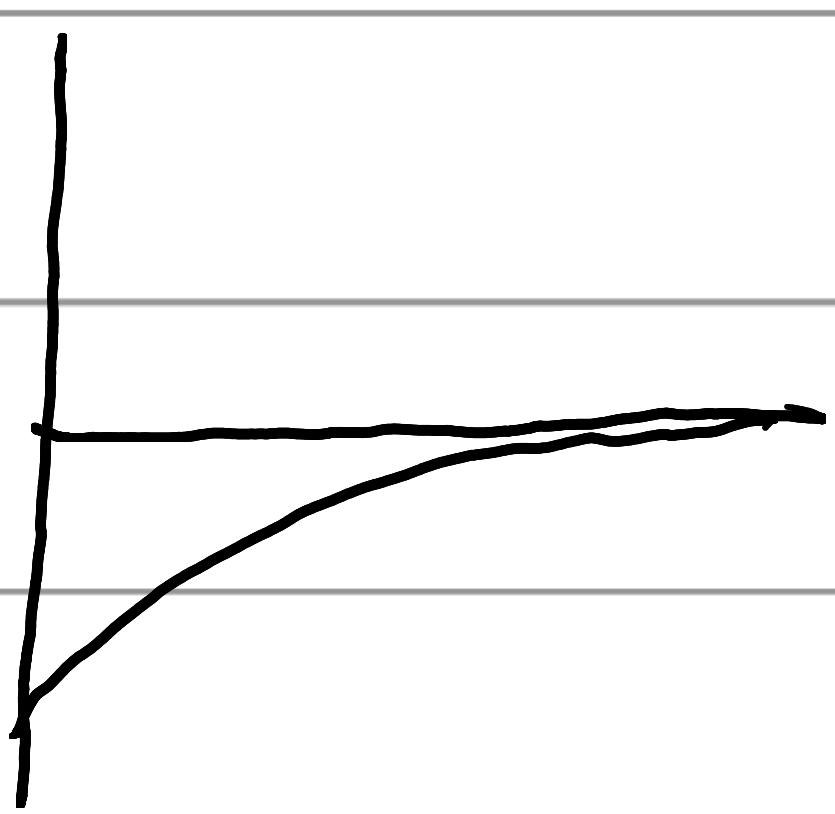
→ increase or decrease  $K_p$



when large overshoot  
l fast conversion

→ increase or decrease

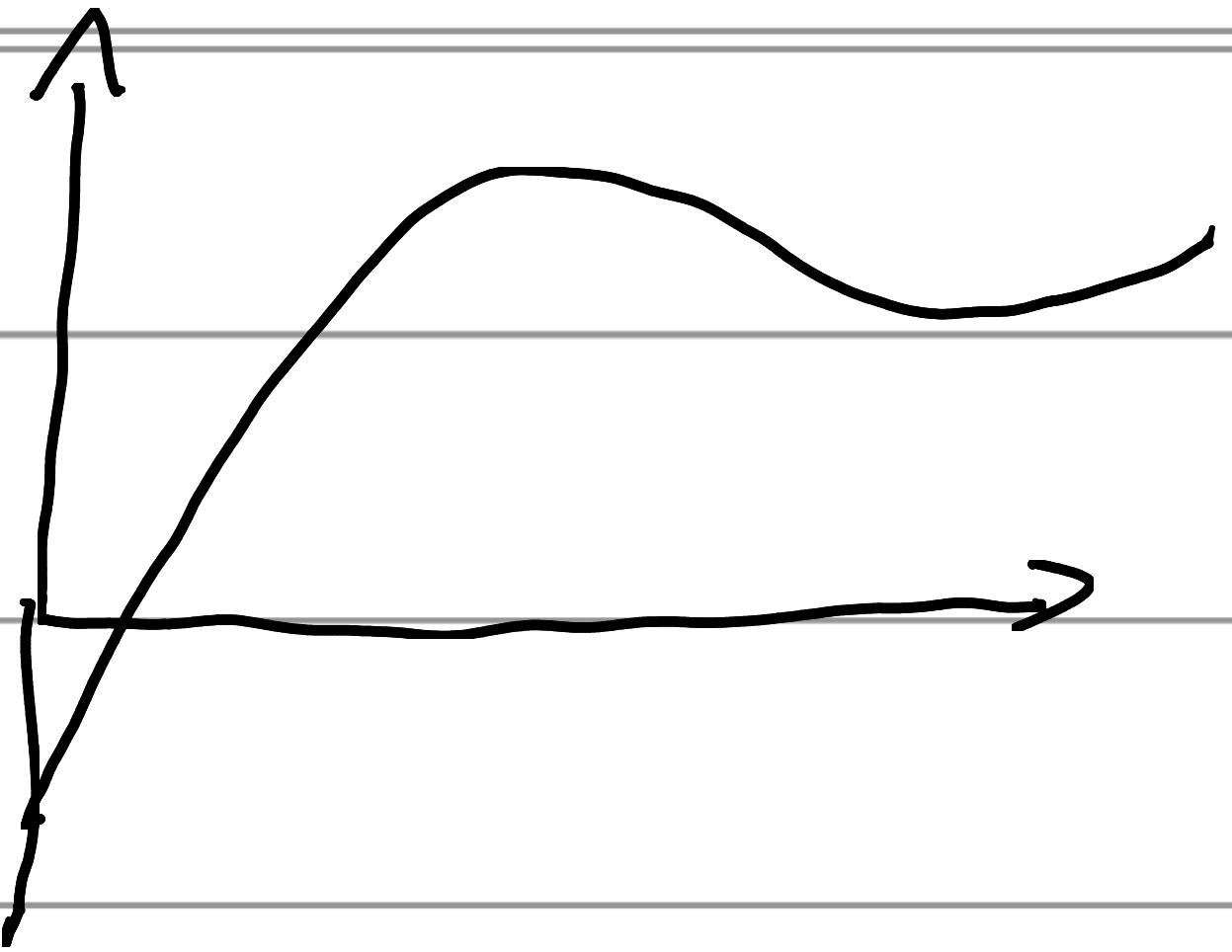
both  $K_p$  &  $K_I$



when no oscillation l

slow conversion.

→ Decrease  $K_p$  &  $K_I$

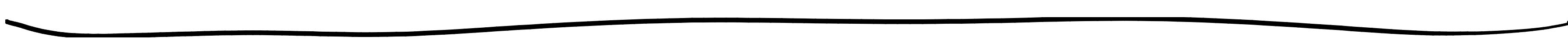


when slow oscillating

↓ slow conversion

→ Decrease  $K_p$

→ Increase  $K_I$



Slow conversion

yes

yes

$K_p -- K_I ++$

no

no

$K_p -- K_I --$

Large overshoot  $\rightarrow K_p -- k_z --$

or

$K_{p++} K_I ++$

Oscillating

yes small overshoot

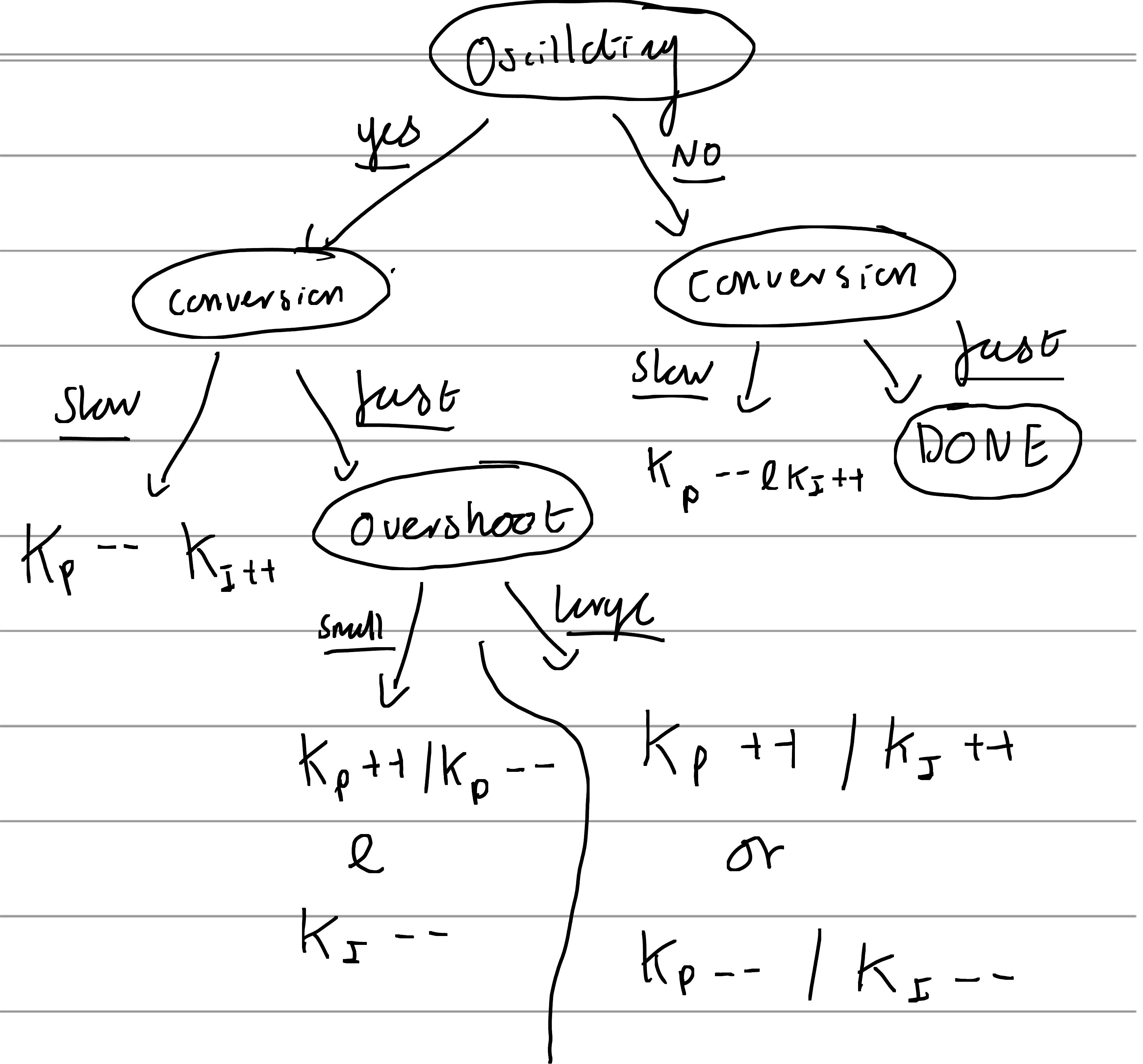
overshoot

$\rightarrow$

$K_p -- / K_p ++$

$\ell$

$K_I --$



What about  $K_d$ ?