4/1

Mass Spring Damper $\qquad m\ddot{y} + \eta\dot{y} + ky = u$

$$\mathcal{L}[m\ddot{y} + \eta\dot{y} + ky] = \mathcal{L}[u]$$

$y = y(t)$
$u = u(t)$

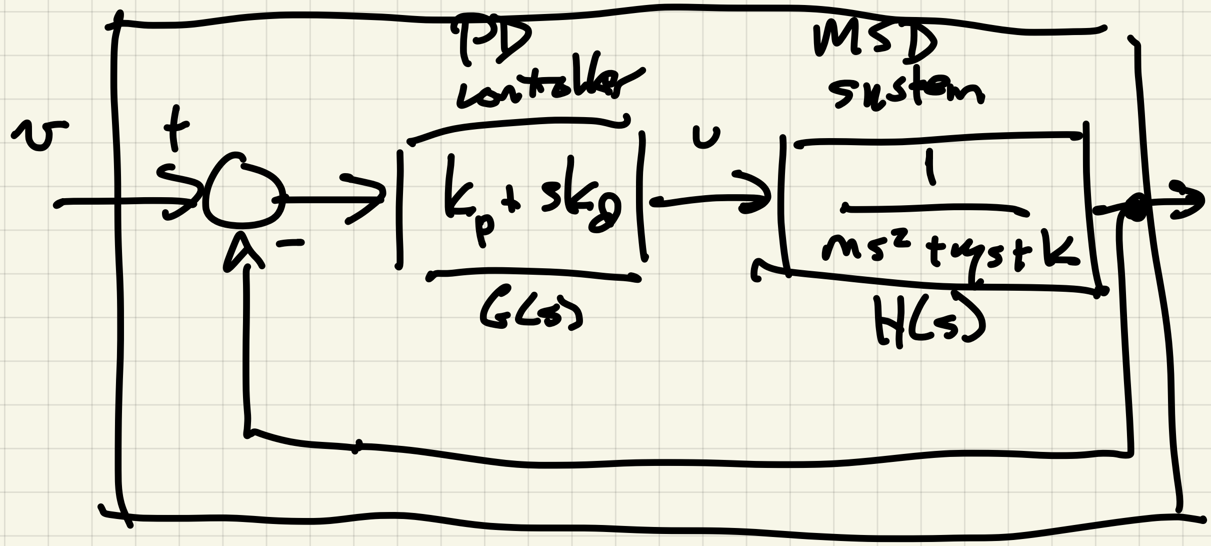
$$ms^2 Y(s) + \eta s Y(s) + kY(s) = U(s)$$

$$(ms^2 + \eta s + k) Y(s) = U(s)$$

$$Y(s) = \underbrace{\frac{1}{ms^2 + \eta s + k}}_{H(s)} U(s)$$

$$H(s) = \frac{Y(s)}{U(s)}$$

PD



$m > n$
not causal

$$G(s) = k_d s + k_p = \frac{k_d s + k_p}{1}$$

$$T_{cl} = \frac{G(s)H(s)}{1 + G(s)H(s)}$$

$$= \frac{(k_d s + k_p)\left(\frac{1}{ms^2 + \gamma s + k}\right)}{1 + (k_d s + k_p)\left(\frac{1}{ms^2 + \gamma s + k}\right)}$$

$$= \frac{k_d s + k_p}{ms^2 + \gamma s + k + k_d s + k_p} = \frac{k_d s + k_p}{ms^2 + (\gamma + k_d)s + (k + k_p)}$$

Steady State (step response)

$$y_{ss} = \lim_{t \to \infty} y(t) = \lim_{s \to 0} \cancel{s} H(s) \frac{1}{\cancel{s}} = \frac{k_p}{k + k_p} < 1$$

PID

controller $G(s) = k_p + k_d s + \frac{k_i}{s}$

$$= \frac{k_d s^2 + k_p s + k_i}{s}$$

still not causal!

$$T_{cl}(s) = \frac{GH}{1+GH} = \frac{\frac{k_d s^2 + k_p s + k_i}{s} \cdot \frac{1}{ms^2 + \gamma s + k}}{1 + \frac{k_d s^2 + k_p s + k_i}{s} \cdot \frac{1}{ms^2 + \gamma s + k}}$$

$$= \frac{k_d s^2 + k_p s + k_i}{s(ms^2 + \gamma s + k) + k_d s^2 + k_p s + k_i}$$

$$= \frac{k_d s^2 + k_p s + k_i}{ms^3 + (\gamma + k_d)s^2 + (k + k_p)s + k_i}$$

$$y_{ss} = \lim_{s \to 0} \cancel{s}\, H(s) \frac{1}{\cancel{s}} = \frac{k_i}{k_i} = \underline{1} \;!!$$

Causality:

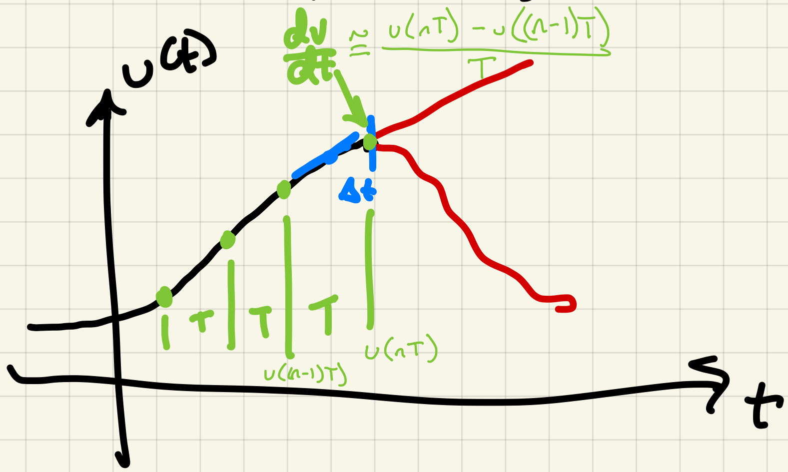
$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$m < n$ strictly causal $\Rightarrow$ ss realization (A, B, C)

$m \leq n$ causal $\Rightarrow$ ss realization (A, B, C, D)

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

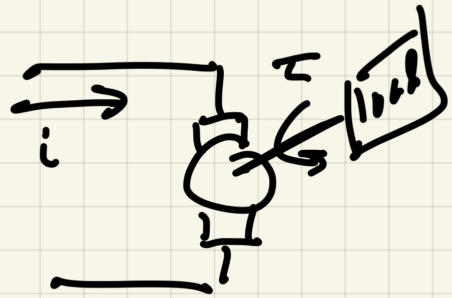
$m > n$ noncausal $\Rightarrow$ no ss representation



Mobile Robots: levels of control

Brushed DC Motor

assume current is input



torque constant

$$\tau = K_T i$$

motor velocity: ω

$$i \rightarrow \boxed{H(s)} \rightarrow \omega$$

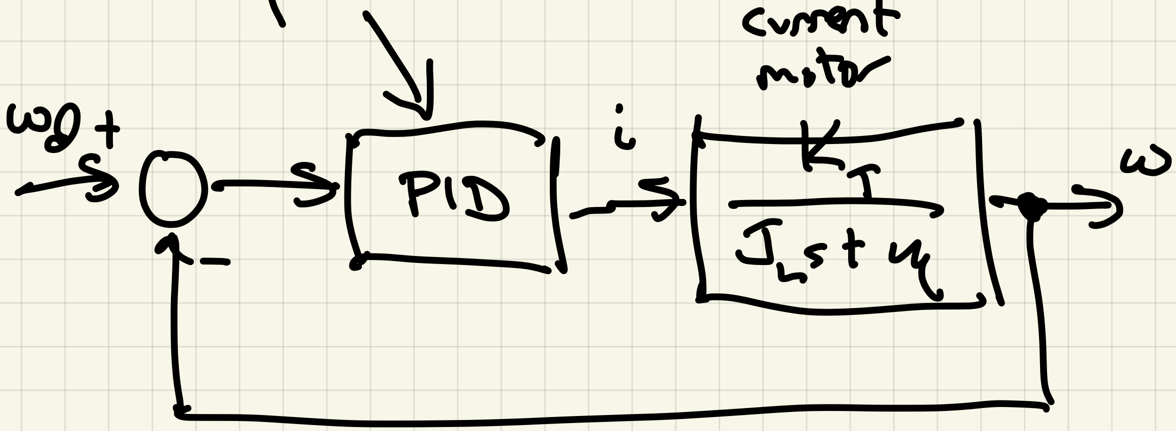
ODE (newton's law

moment of inertia of load

$$\tau = J_L \dot{\omega} + \eta \omega = K_T i$$

$$H(s) = \frac{K_T}{J_L s + \eta}$$

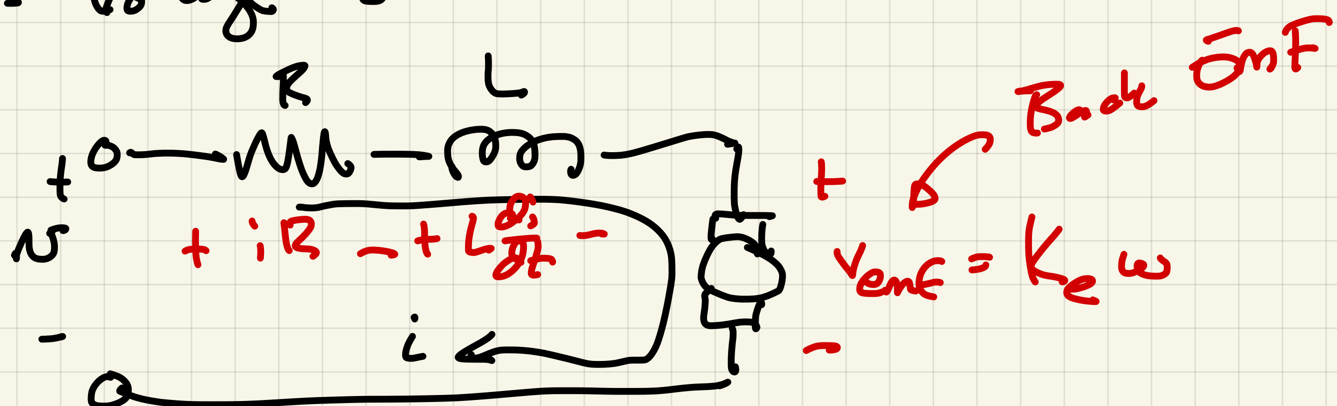
current motor speed controller



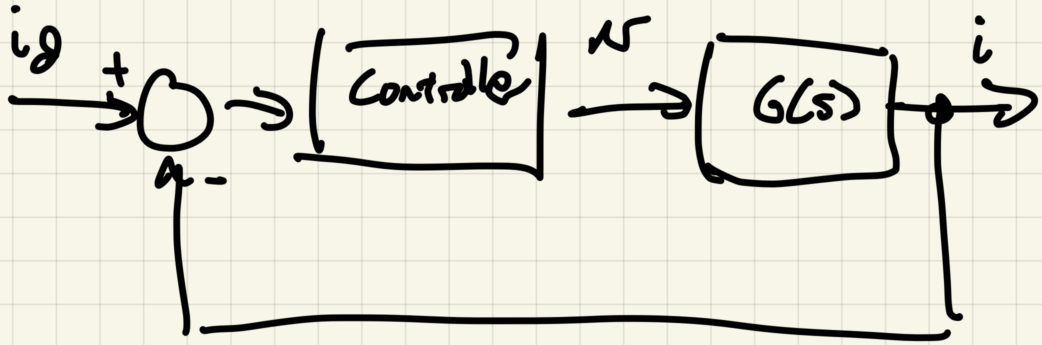
$\omega \to \omega_d$

But, we don't really control motor current, so it is not a good input

input as voltage v



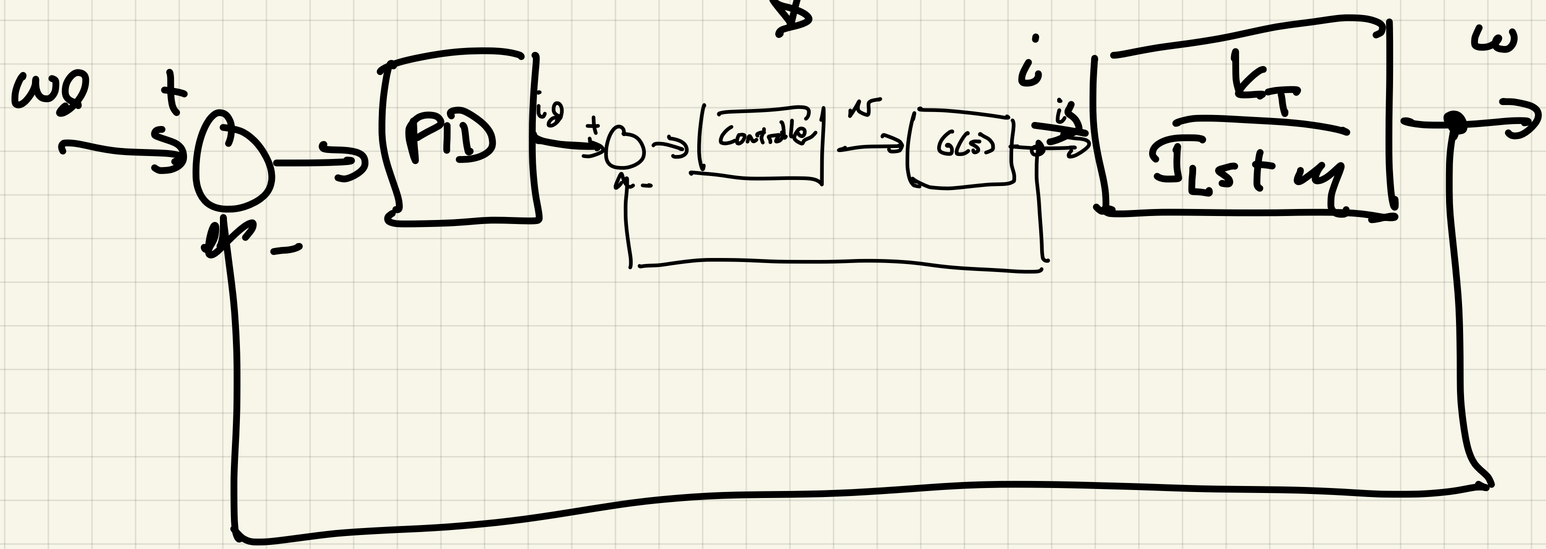
consider i as output

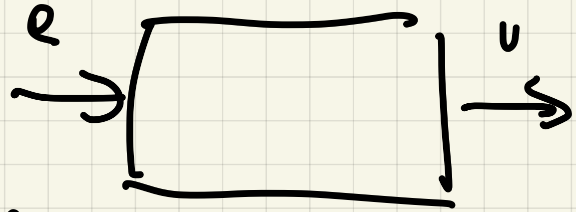


$i \to i_d$

back to current as input

voltage to current amplifier





$$\mathcal{L}\left[u = k_p e + k_d \frac{de}{dt} \right]$$

$$U(s) = k_p \bar{E} + s K_d \bar{E}$$

$$= \underbrace{\left(k_p + s K_d\right)}_{G(s)} \bar{E}$$

4/6 Differential Drive Robot

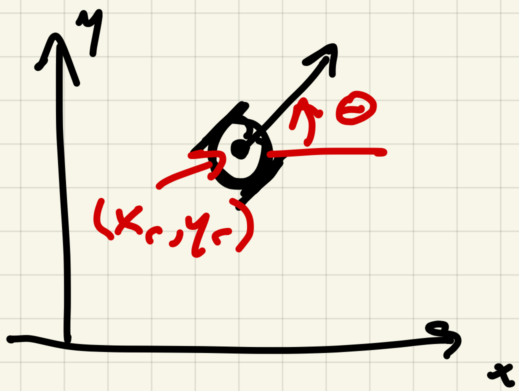
Model: given wheel speeds, what are the robot

$$\begin{bmatrix}\omega_\ell \\ \omega_r\end{bmatrix} \xrightarrow{\text{wheel speeds}} \boxed{} \longrightarrow \text{robot speeds} \begin{bmatrix} v_f \\ \omega \end{bmatrix}$$

Kinematic Model (as opposed to a dynamic model)

↳ the study of motion without regard to the forces that cause it

Robot Pose: position and orientation

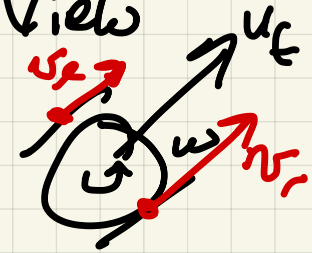


$$q = \begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix} = x \quad \text{pose configuration}$$

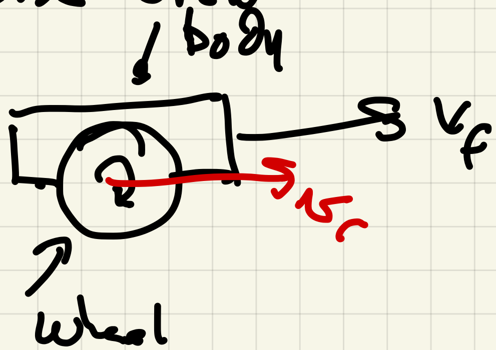
$$v = \begin{bmatrix} v_f \\ \omega \end{bmatrix}$$

Robot Cartoons

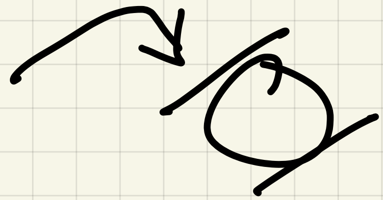
Top View



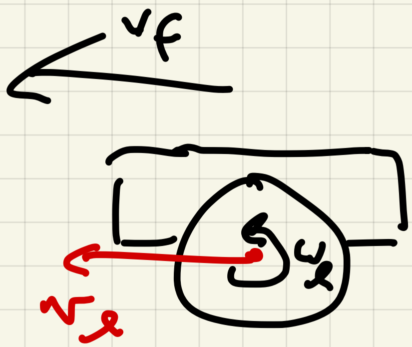
Side View



Intermediate Speed: wheel linear velocities $\begin{bmatrix} v_l \\ v_r \end{bmatrix}$



Side view from left



R = wheel radius

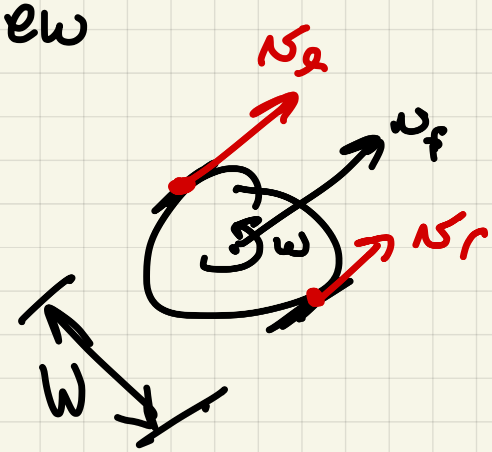
$v_l = R\omega_l$

Side view from right

$v_r = -R\omega_r$

Robot Velocities

Top View



$$v_f = \frac{v_l + v_r}{2}$$

W = wheelbase

$$\omega = \frac{v_r - v_l}{W}$$

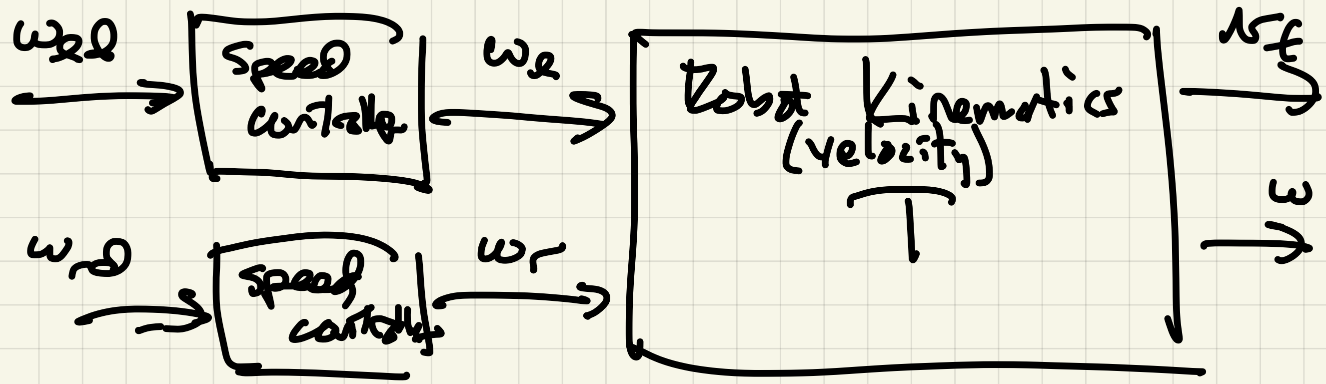
<u>combine</u>:

$$v_f = \frac{-\omega_r R + \omega_l R}{2} = -\frac{R}{2}\omega_r + \frac{R}{2}\omega_l$$

$$\omega = \frac{-\omega_r R - \omega_l R}{W} = -\frac{R}{W}\omega_r - \frac{R}{W}\omega_l$$

$$\begin{bmatrix} v_f \\ \omega \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{R}{2} & -\frac{R}{2} \\ -\frac{R}{W} & -\frac{R}{W} \end{bmatrix}}_{T} \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix}$$

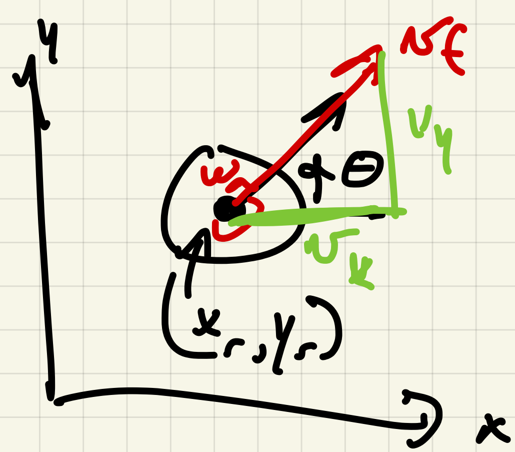
$$\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} \longrightarrow \boxed{T} \longrightarrow \begin{bmatrix} v_f \\ \omega \end{bmatrix}$$



Another Model takes robot velocities to robot pose

$$\begin{bmatrix} v_f \\ \omega \end{bmatrix} \to \boxed{\quad} \to \begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix} = x$$

Still kinematic



$$\dot{x} = \begin{bmatrix} v_f \cos\theta \\ v_f \sin\theta \\ \omega \end{bmatrix}$$

input $\begin{bmatrix} v_f \\ \omega \end{bmatrix} = u$ state $x = \begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix}$

$$\dot{x} = f(x, u)$$

$$\dot{x} = \begin{bmatrix} v_f \cos\theta \\ v_f \sin\theta \\ \omega \end{bmatrix} = \begin{bmatrix} u_1 \cos x_3 \\ u_1 \sin x_3 \\ u_2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \cos x_3 \\ \sin x_3 \\ 0 \end{bmatrix}}_{g_1} u_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{g_2} u_2$$

drift free affine state space model

$$= \sum_{i=1}^{m} g_i u_i$$

g_1, g_2 control vector fields

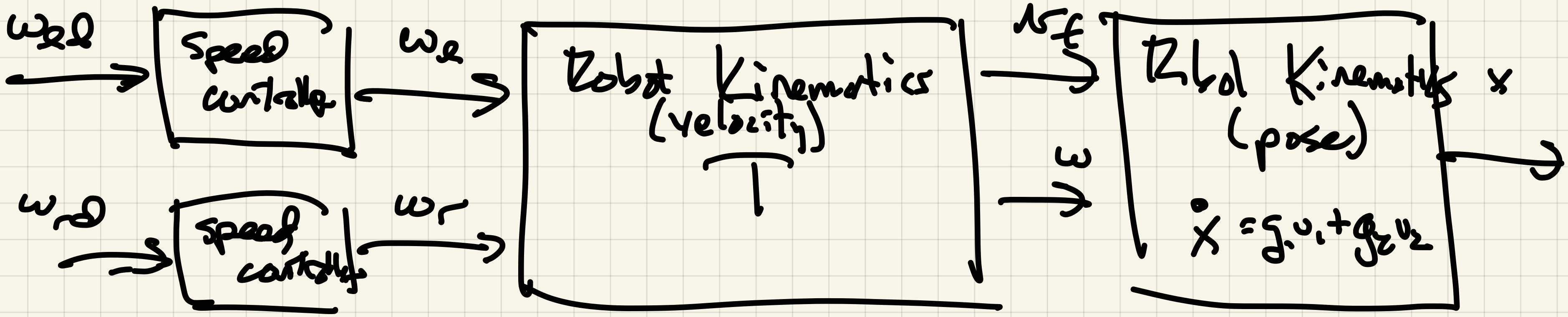
<u>Drift</u> <u>Free</u> when $u = 0$ the $\dot{x}$ also is 0

every state is an equilibrium point!

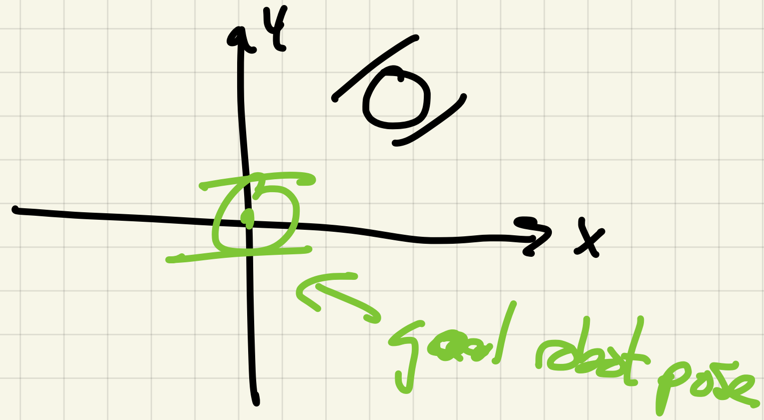
(affine system with drift)

$$\dot{x} = f(x) + g_1(x) u_1 + g_2(x) u_2 + \cdots + g_m(x) u_m$$

$f(x)$: drift vector field



First Try to Linearize



$$\dot{x} = f(x,u) = \begin{bmatrix} u_1 \cos x_3 \\ u_1 \sin x_3 \\ u_2 \end{bmatrix}$$

$$\dot{x} = \frac{\partial f}{\partial x}\Big|_{x=0,u=0} x + \frac{\partial f}{\partial u}\Big|_{x=0,u=0} u$$

$$= \begin{bmatrix} 0 & 0 & -u_1 \cancelto{0}{\sin x_3} \\ 0 & 0 & u_1 \cancelto{0}{\cos x_3} \\ 0 & 0 & 0 \end{bmatrix}\Bigg|_{x=0,u=0} x - \begin{bmatrix} \cancelto{1}{\cos x_3} & 0 \\ \cancelto{0}{\sin x_3} & 0 \\ 0 & 1 \end{bmatrix}\Bigg|_{x=0,u=0} u$$

$$\dot{x} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} u$$

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} v$$

Check controllability

$$Q = \begin{bmatrix} B & AB & A^2B \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \text{rank} = 2 \Rightarrow \text{not controllable}$$

Original nonlinear system: is it controllable? Yes!

But - linearization is not controllable!

And - dimension of control space is 2

dimension of state space is 3

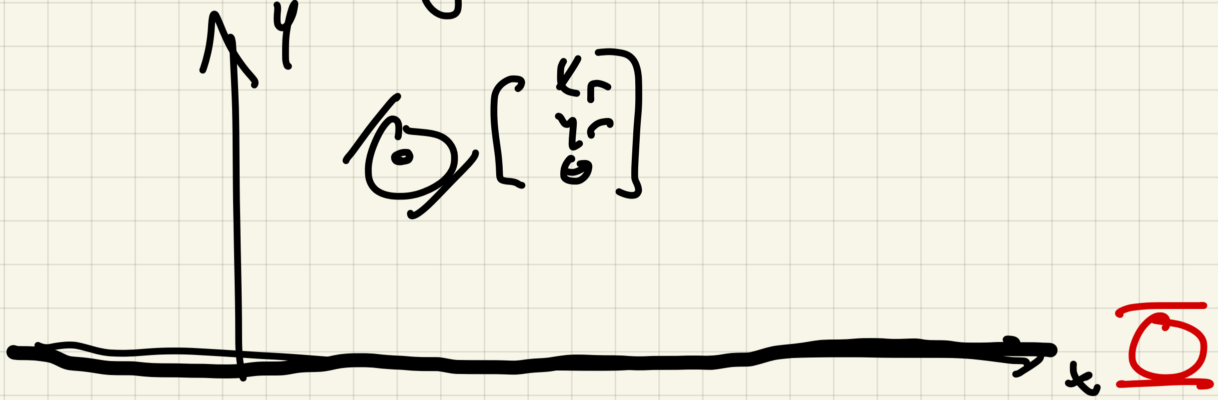
$2 < 3$

"not altogether lawful"

$\Rightarrow$ there is a nonholonomic constraint.

$\Rightarrow$ the system is under actuated.

Line following



let v_f be constant

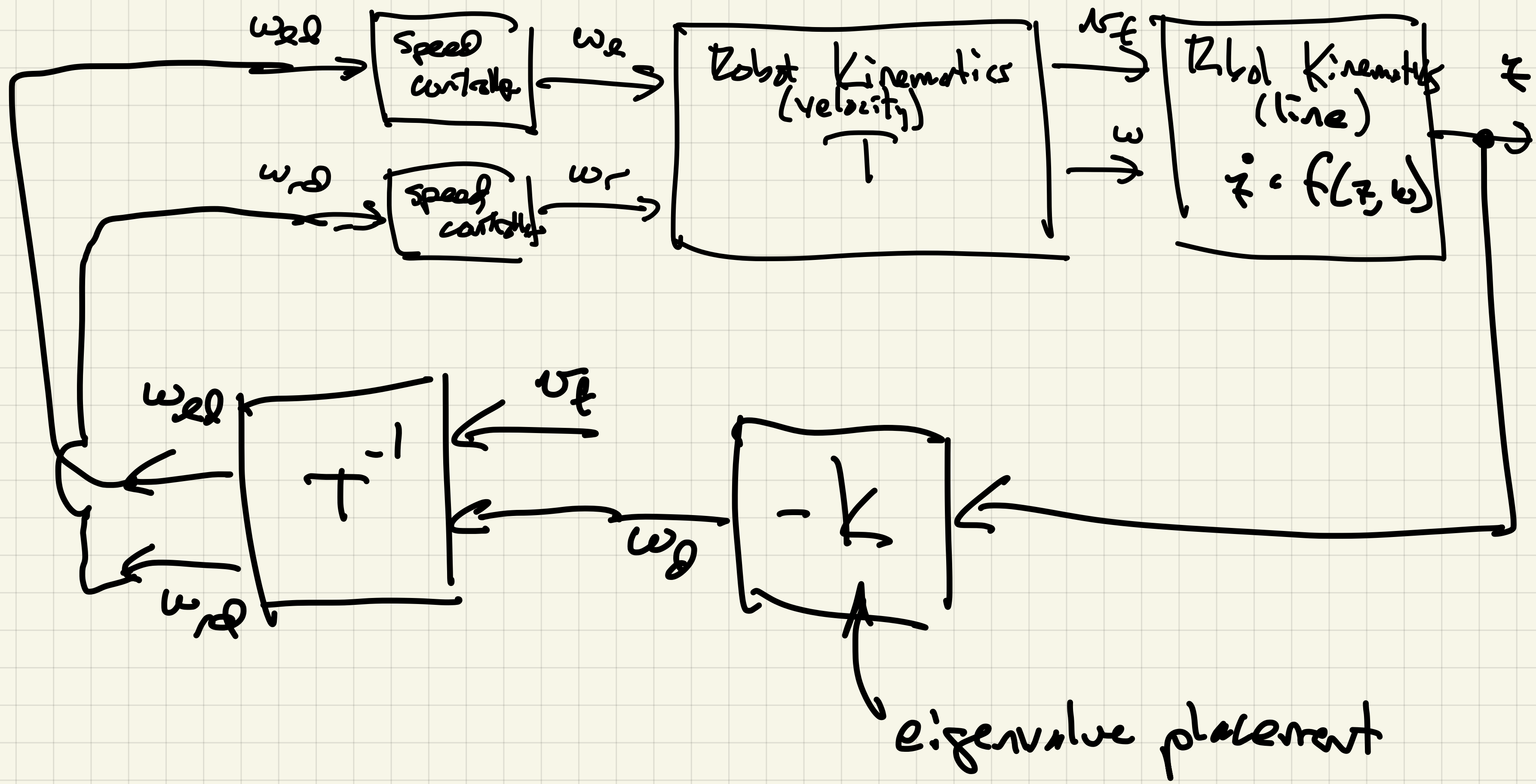
new state $z = \begin{bmatrix} y_r \\ \theta \end{bmatrix}$ $\qquad \dot{x} = f(x,u) = \begin{bmatrix} u_1 \cos x_3 \\ u_1 \sin x_3 \\ u_2 \end{bmatrix}$

$$\dot{z} = \begin{bmatrix} v_f \sin z_2 \\ \omega \end{bmatrix}$$

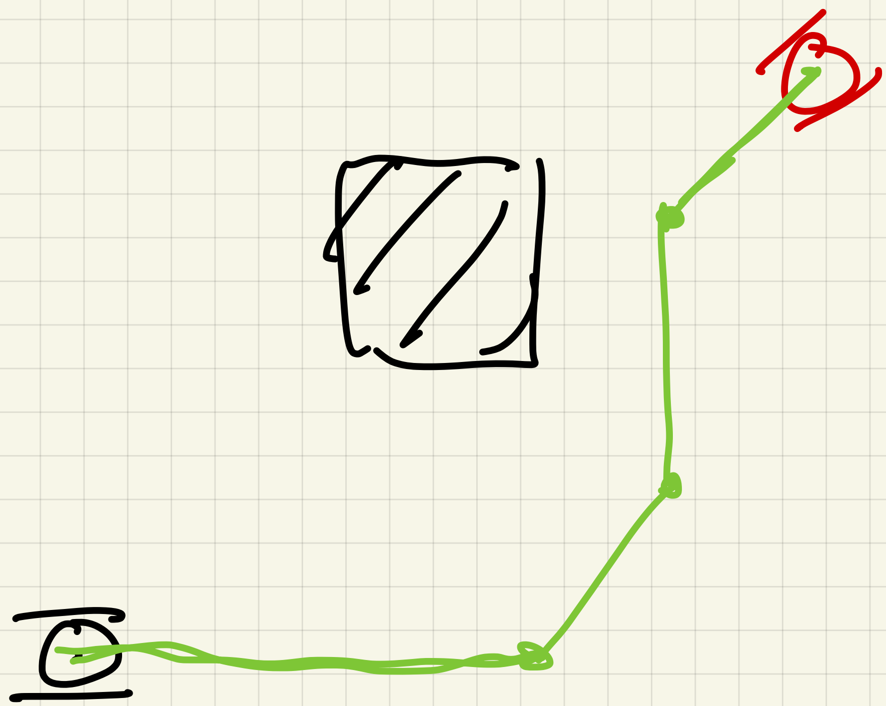
linearize about $z = 0$

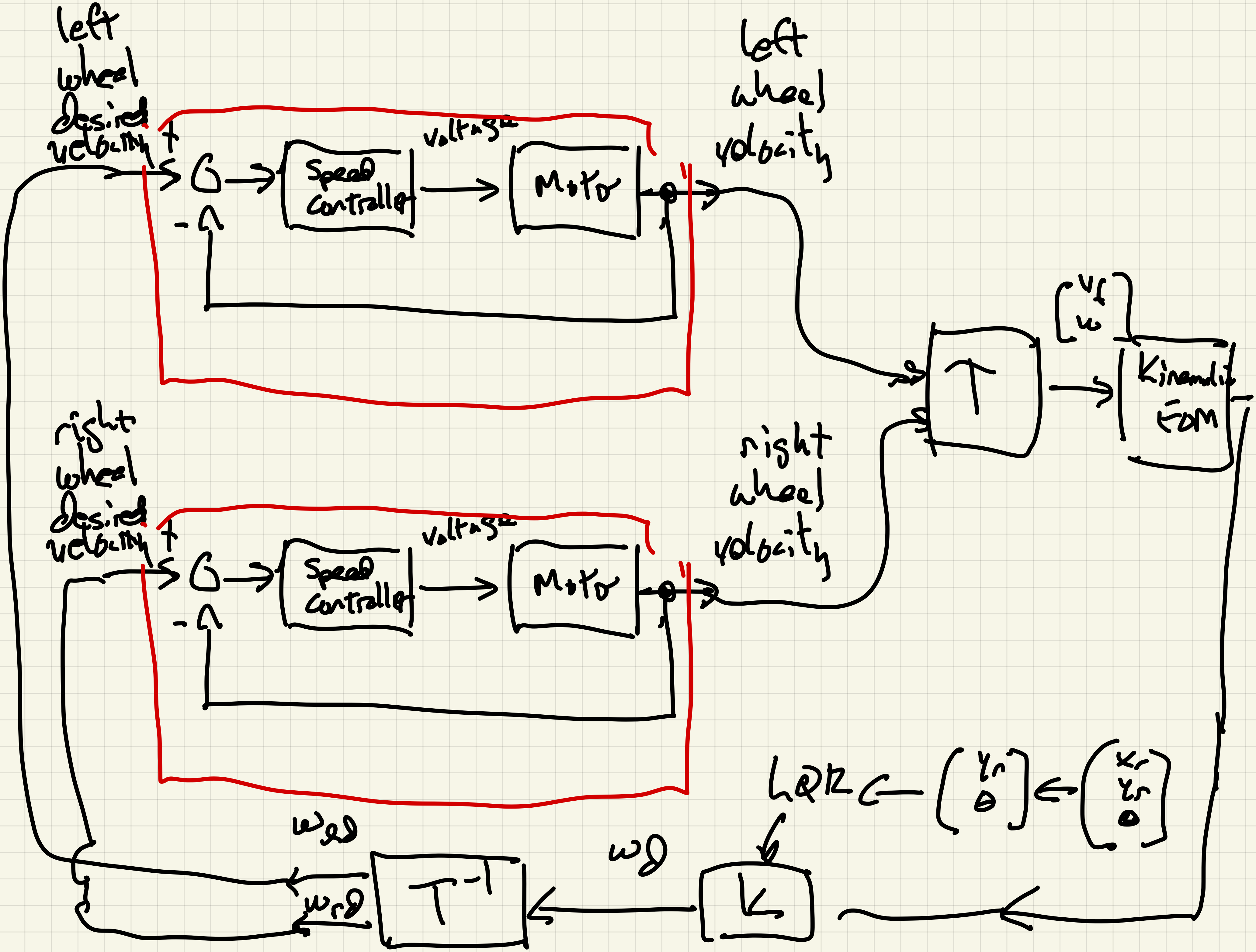
$$\dot{z} = \begin{bmatrix} 0 & v_f \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \omega$$

$Q = \begin{bmatrix} 0 & v_f \\ 1 & 0 \end{bmatrix}$ rank $= 2$ $\Rightarrow$ controllable!

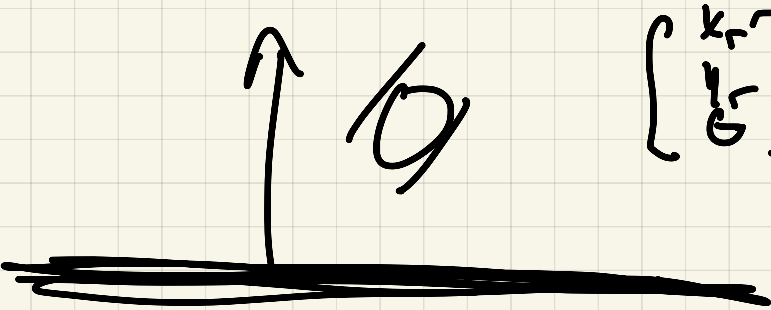


ω_{ld}
speed controller
ω_l
Robot Kinematics (velocity)
T
v_f
ω
Robot Kinematics (line)
$\dot{z} = f(z, \omega)$
z
ω_{rd}
speed controller
ω_r
ω_{ld}
T^{-1}
v_f
ω_d
$-K$
ω_{rd}
eigenvalue placement



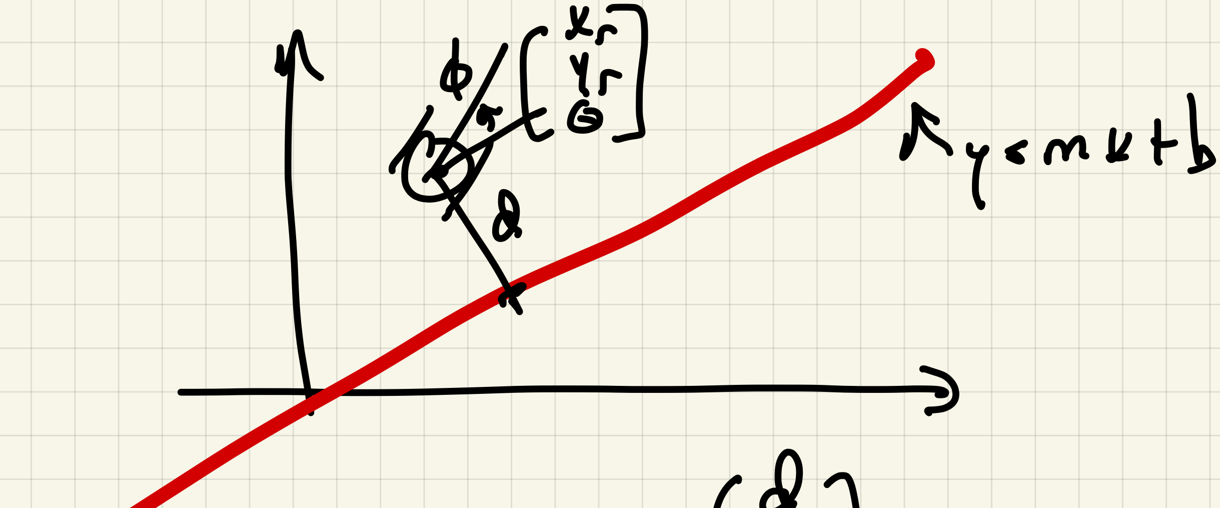


left wheel desired velocity
+
-
Speed Controller
voltage
Motor
left wheel velocity
right wheel desired velocity
+
-
Speed Controller
voltage
Motor
right wheel velocity
T
$\begin{bmatrix} v_f \\ \omega \end{bmatrix}$
Kinematic EOM
LQR
$\begin{bmatrix} y_r \\ \theta \end{bmatrix}$
$\begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix}$
K
ω_d
T^{-1}
ω_{ld}
ω_{rd}

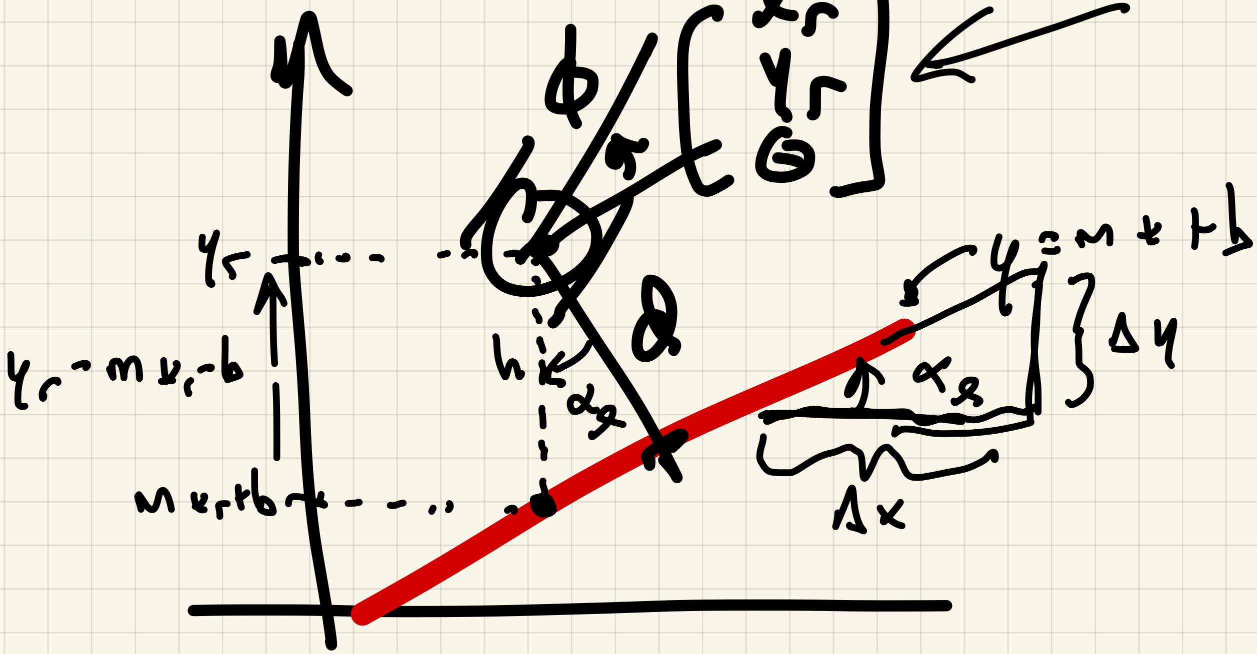


$$\begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} y_r \\ \theta \end{bmatrix}$$

Different line



$$\text{new state} = \begin{bmatrix} d \\ \phi \end{bmatrix}$$



Recall

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_f \cos\theta \\ v_f \sin\theta \\ \omega \end{bmatrix}$$

$$m = \frac{\Delta y}{\Delta x}$$

$$m = \tan \alpha_e$$

$$\alpha_e = \operatorname{atan}(m)$$

$$h = y_r - m x_r - b \Rightarrow d = (y_r - m x_r - b) \cos \alpha_e$$

$$\phi = \theta - \alpha_e$$

$\underline{\text{goal}}$: $\dot{z} = f(z, v)$

$$z = \begin{bmatrix} d \\ \phi \end{bmatrix}$$

$$\dot{z} = \begin{bmatrix} \dot{d} \\ \dot{\phi} \end{bmatrix} \Rightarrow$$

$$\dot{d} = \cos\alpha_e (\dot{y}_r - m \dot{x}_r)$$
$$= \cos\alpha_e (v_f \sin\theta - m v_f \cos\theta)$$
$$= v_f \cos\alpha_e (\sin(\phi + \alpha_e) - m \cos(\phi + \alpha_e))$$

$$\dot{d} = v_t \cos\alpha_e \left(\sin(\phi + \alpha_e) - m \cos(\phi + \alpha_e) \right)$$

$$\dot{\phi} = \dot{\theta} = \omega$$

<u>Intuitively</u> $z = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ shall be an equilibrium point

for input $= \begin{bmatrix} v_t \\ 0 \end{bmatrix}$

set $d = \phi = \omega = 0$, v_t = constant

$\dot{\phi} = 0$ ✓

<u>Recall</u>:

$$\sin(\phi + \alpha_e) = \sin\phi \cos\alpha_e + \cos\phi \sin\alpha_e$$
$$\cos(\phi + \alpha_e) = \cos\phi \cos\alpha_e - \sin\phi \sin\alpha_e$$

$$\Rightarrow \dot{d} = v_t \cos\alpha_e \Big(\cancelto{0}{\sin\phi} \cos\alpha_e + \cancelto{1}{\cos\phi} \sin\alpha_e$$
$$- m \cancelto{1}{\cos\phi} \cos\alpha_e + m \cancelto{0}{\sin\phi} \sin\alpha_e \Big)$$

$$= v_t \cos\alpha_e \left(\sin\alpha_e - m \cos\alpha_e \right)$$

$$\dot{d} = v_t \cos\alpha_l \left(\sin\alpha_l - m \cos\alpha_l \right)$$

$$m = \frac{\Delta x}{\Delta y} = \tan\alpha_l = \frac{\sin\alpha_l}{\cos\alpha_l}$$

$$\dot{d} = v_t \cos\alpha_l \left(\cancel{\sin\alpha_l} - \frac{\cancel{\sin\alpha_l}}{\cancel{\cos\alpha_l}} \cancel{\cos\alpha_l} \right) = 0 \;!!$$

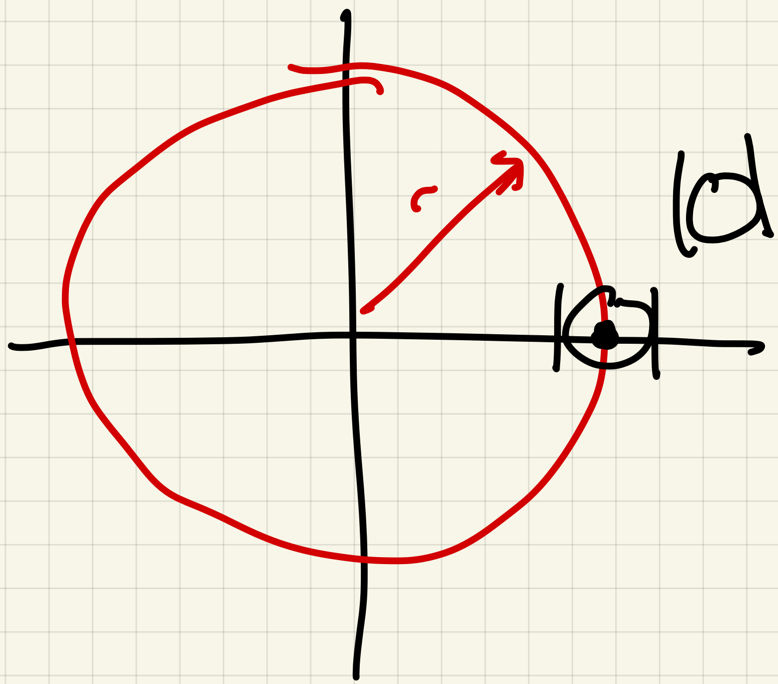
$$\dot{d} = v_t \cos\alpha_l \left(\sin(\phi + \alpha_l) - m \cos(\phi + \alpha_l) \right)$$

$$\dot{\phi} = \dot{\theta} = \omega$$

Linearize about $x_e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, check controllability, do state feedback.

What if trajectory is not a line?

Try a circle



Is there an equilibrium trajectory?

Start at

$$x_0 = \begin{bmatrix} r \\ 0 \\ \pi/2 \end{bmatrix}$$

make $v_f \neq 0$ constant

Is there a constant ω that keeps robot on circle?

Yes! what is it?

circumference of circle: $2\pi r$

forward velocity is v_f

distance = rate × time

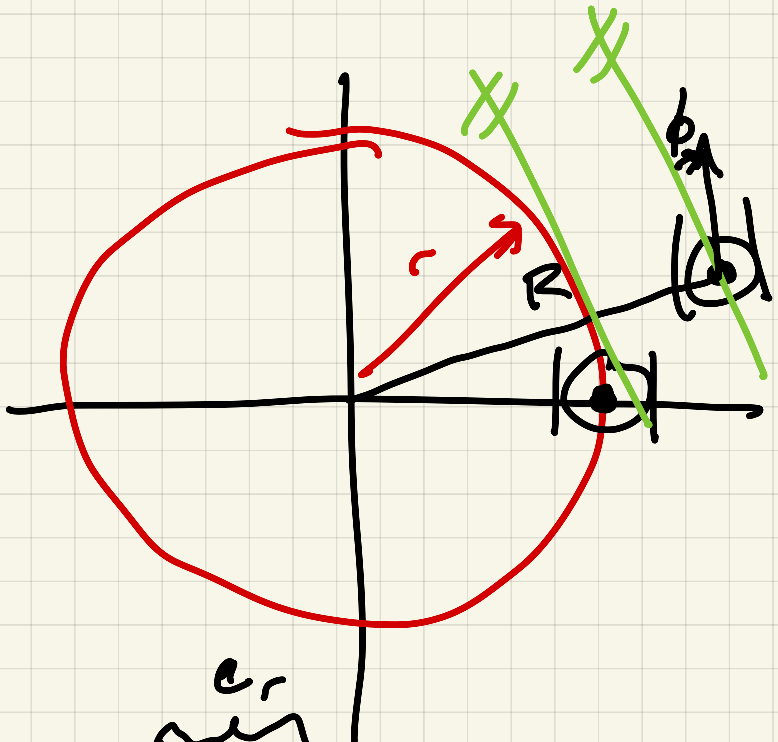
$\Rightarrow$ time to complete the circle is $t_c = \frac{2\pi r}{v_f}$

$$\dot{\theta} = \omega = \frac{2\pi}{2\pi r / v_f} = \frac{v_f}{r}$$

Equilibrium trajectory

initial condition x_0 constant input $= \begin{bmatrix} v_f \\ v_f / r \end{bmatrix}$

$(R-r) \to 0$



$$z = \begin{bmatrix} \overbrace{(R-r)}^{e_r} \\ \phi \end{bmatrix} = \begin{bmatrix} e_r \\ \phi \end{bmatrix}$$

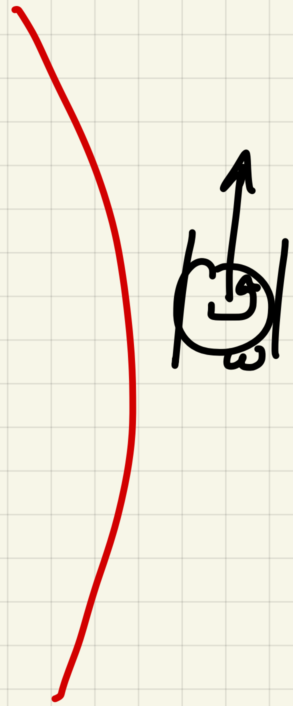
$\dot{e}_r = ?$

$\dot{\phi} = ?$

$f(z, u)$

Ad Hoc

v_f constant



Conceptually

If robot is outside circle, turn left
inside circle, turn right

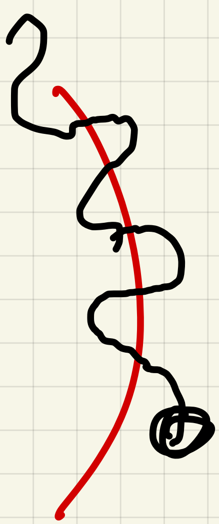
P controller?

$$\omega = k_p(R-r)$$

tune k_p by trial and error

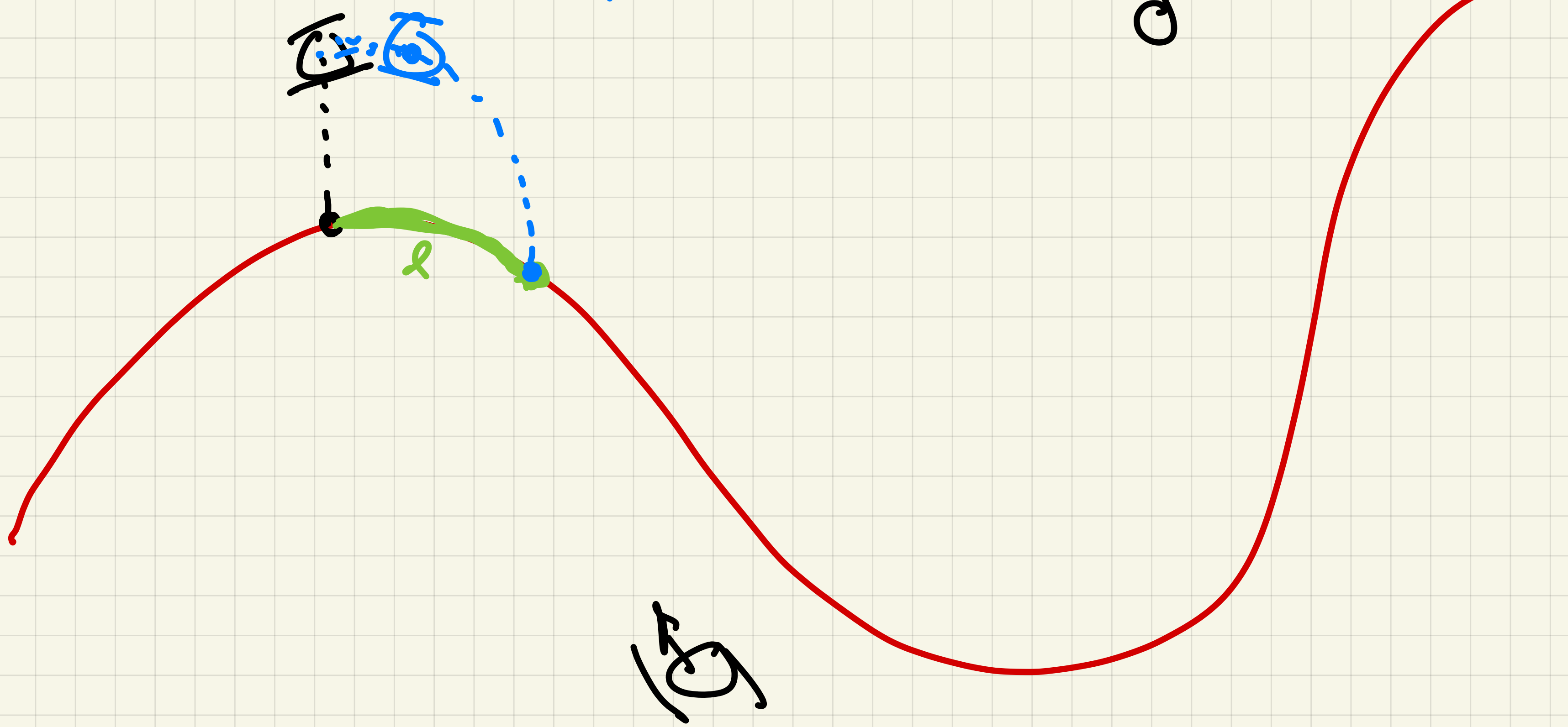
There might be overshoot
⇒ add a D term

$$\omega = k_p(R-r) + k_d\dot{R}$$

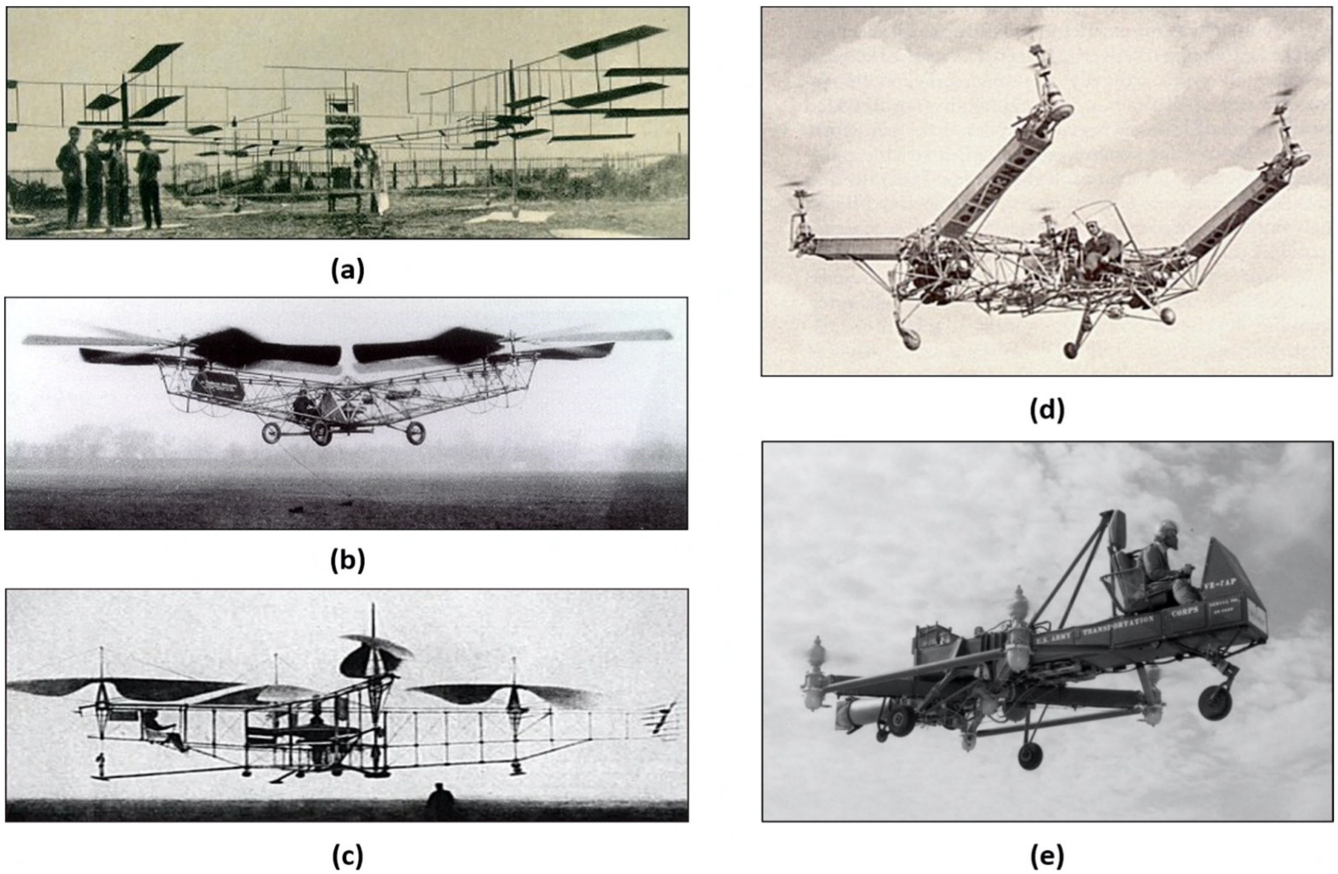


Pure Pursuits

1. project robot position onto path
2. look ahead on the path by some distance l
3. calculate arc that hits lookahead point
4. start following arc

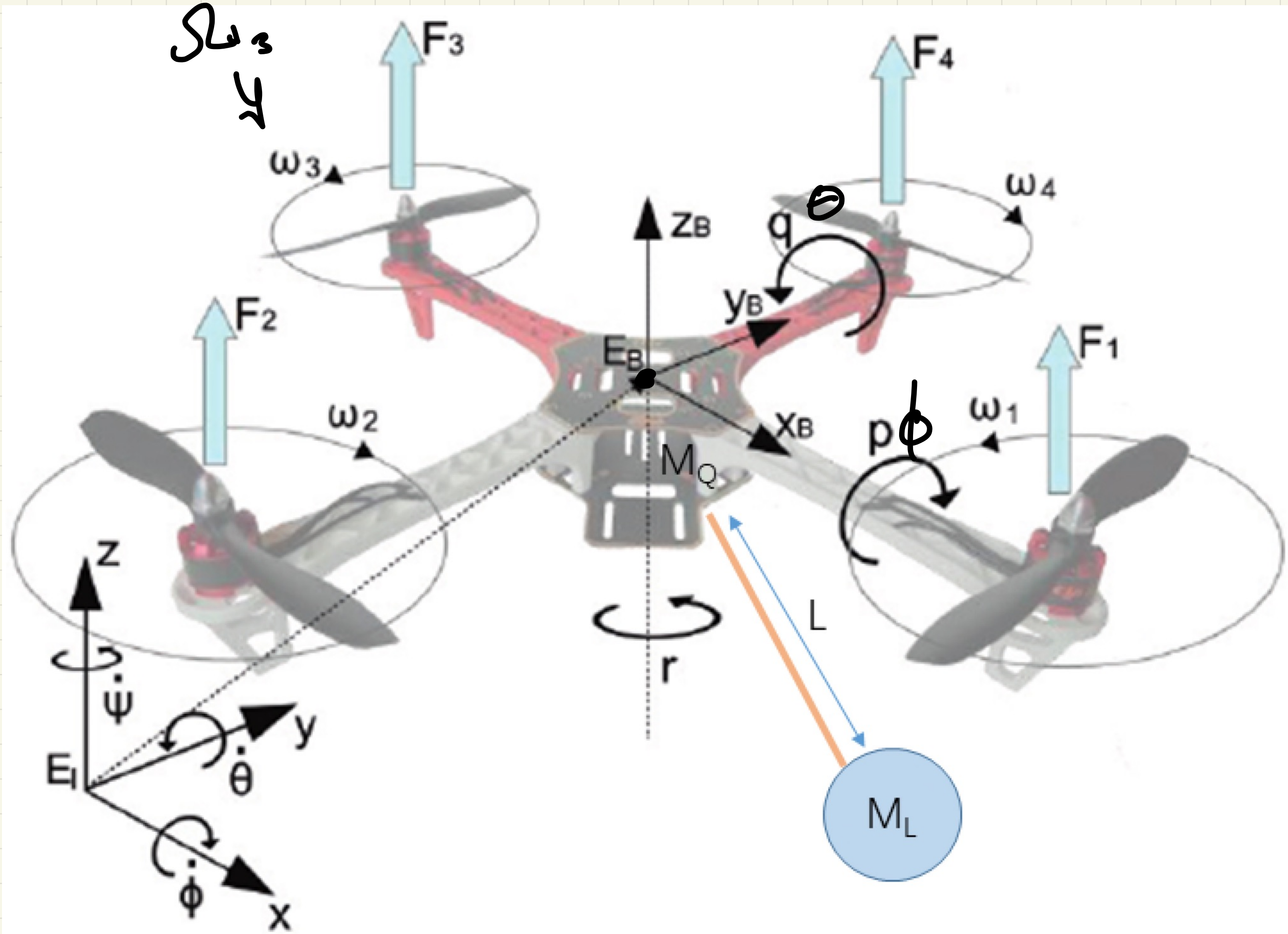


4|13



Source: Kim, Jinho, S. Andrew Gadsden, and Stephen A. Wilkerson. "A comprehensive survey of control strategies for autonomous quadrotors." Canadian Journal of Electrical and Computer Engineering 43, no. 1 (2019): 3-16.

Fig. 3. History of quadrotor: (a) Brèguet-Richet Gyroplane No. 1; (b) Oehmichen No.2; (c) Bothezat helicopter; (d) Convertawings Model A; (e) Curtiss-Wright VZ-7



pose
x
y
z
ϕ
θ
ψ

propeller speeds
ω_1
ω_2
ω_3
ω_4

derivatives
?

sliding mode controller

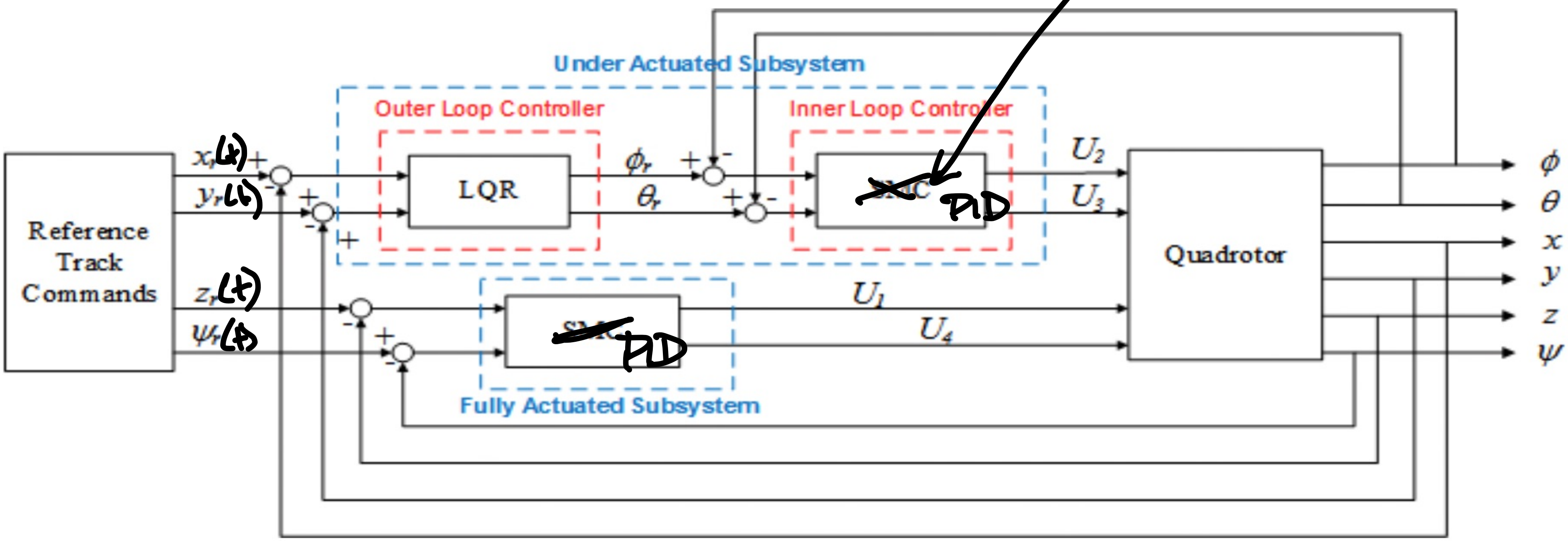
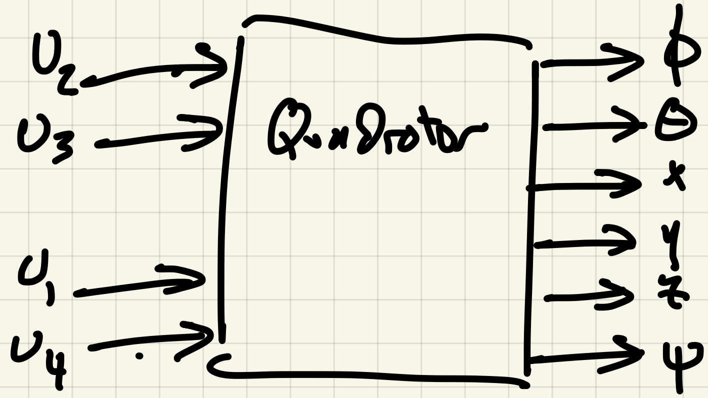


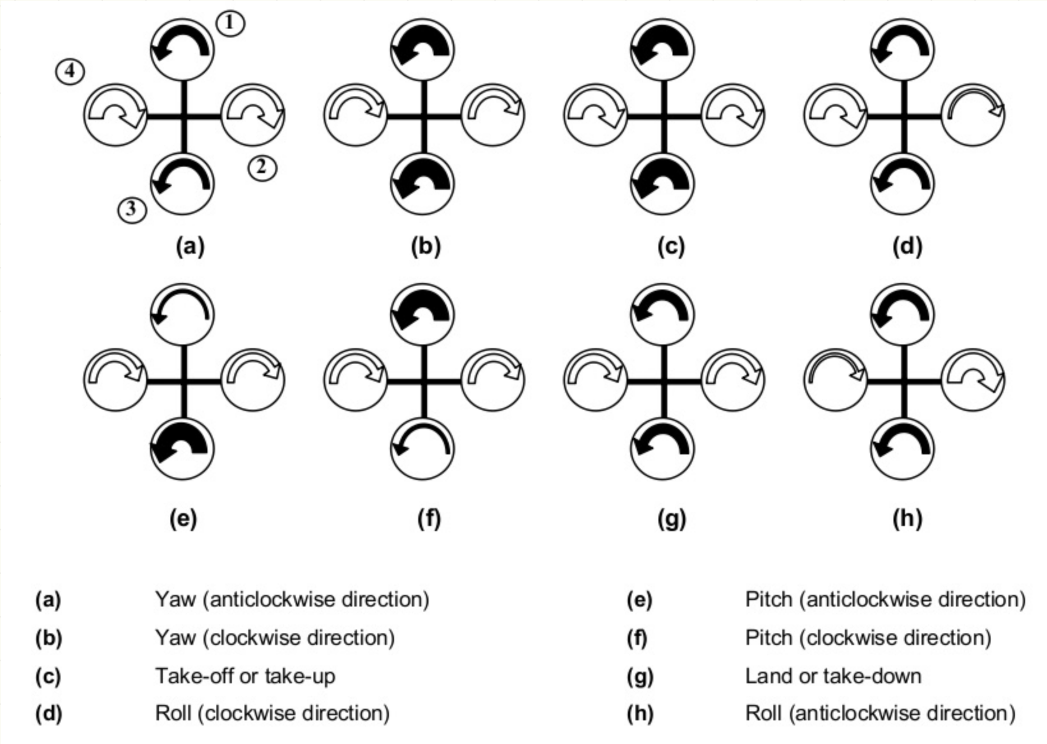
Fig. 2. UAV control system block diagram.

Ghamry, Khaled A., and Youmin Zhang. "Formation control of multiple quadrotors based on leader-follower method." In 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1037-1042. IEEE, 2015.



U's are kind of like thrust thrust constant

Single Propeller $T \approx b\omega^2$



(a, b) conservation of momentum

$$U_4 = d\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right)$$

(c, g) up and down

$$U_1 = b\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right)$$

(e, f) roll, tilt about x axis

ϕ

$$U_3 = b\left(\omega_1^2 - \omega_3^2\right)$$

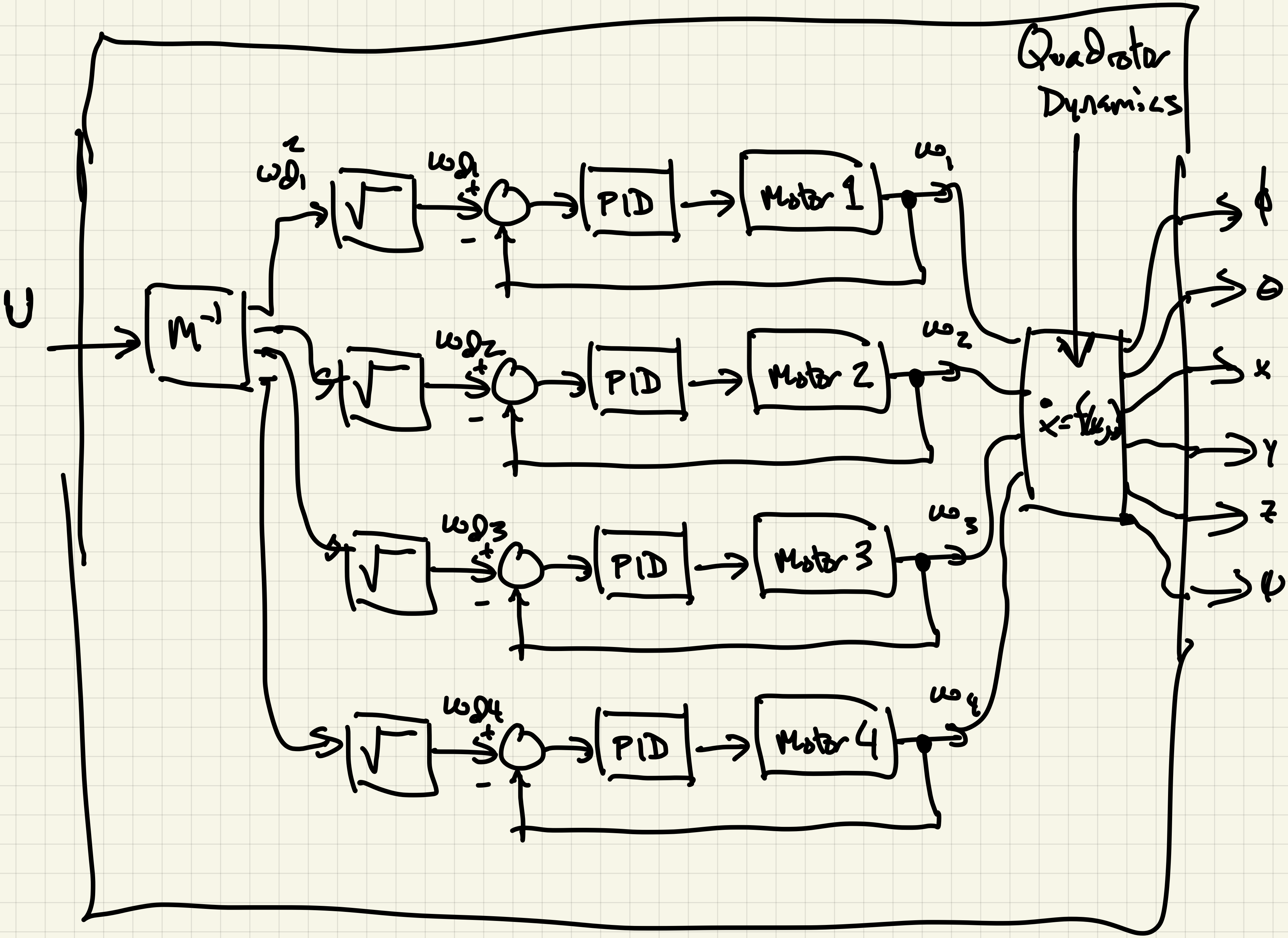
(d, h) pitch, tilt about y axis

θ

$$U_2 = b\left(-\omega_2^2 + \omega_4^2\right)$$

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \overbrace{\begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ b & 0 & -b & 0 \\ -d & d & -d & d \end{bmatrix}}^{M} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

$$\Rightarrow \omega^2 = M^{-1} U$$


Quadrotor Dynamics
U
M^-1
ωd1²
ωd1
PID
Motor 1
ω1
ωd2
PID
Motor 2
ω2
ωd3
PID
Motor 3
ω3
ωd4
PID
Motor 4
ω4
ẋ=f(x,u)
ϕ
θ
x
y
z
ψ

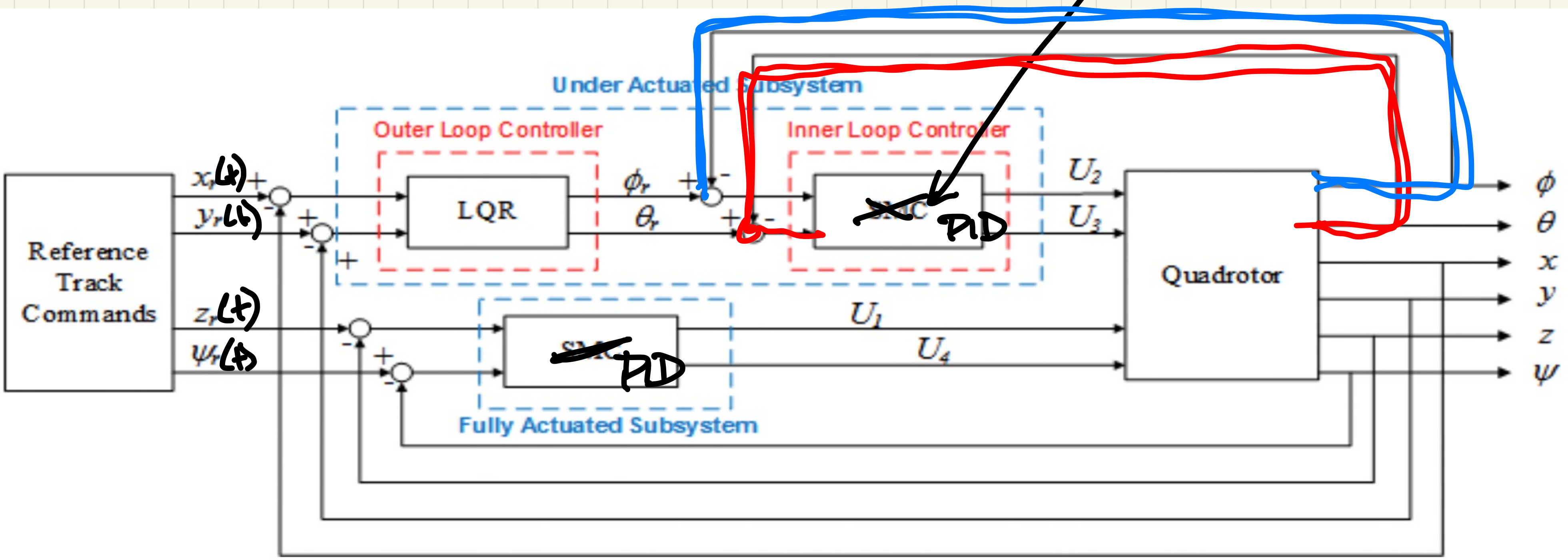
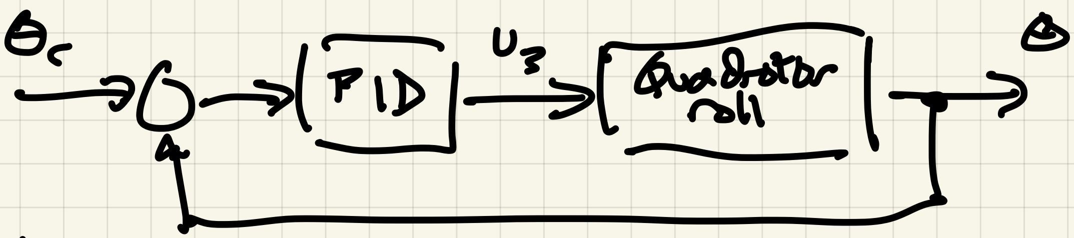
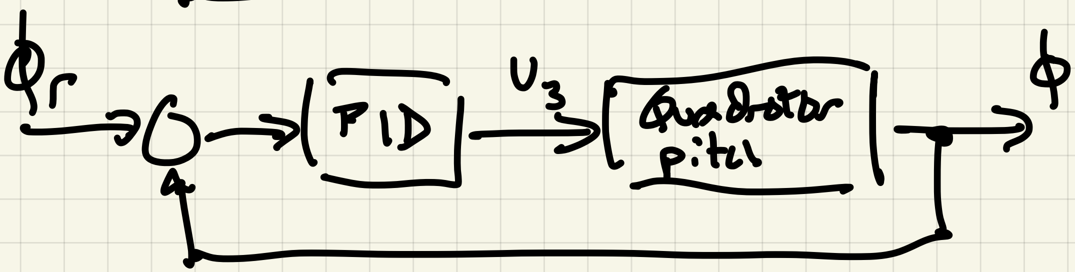


Fig. 2. UAV control system block diagram.

Ghamry, Khaled A., and Youmin Zhang. "Formation control of multiple quadrotors based on leader-follower method." In 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1037-1042. IEEE, 2015.

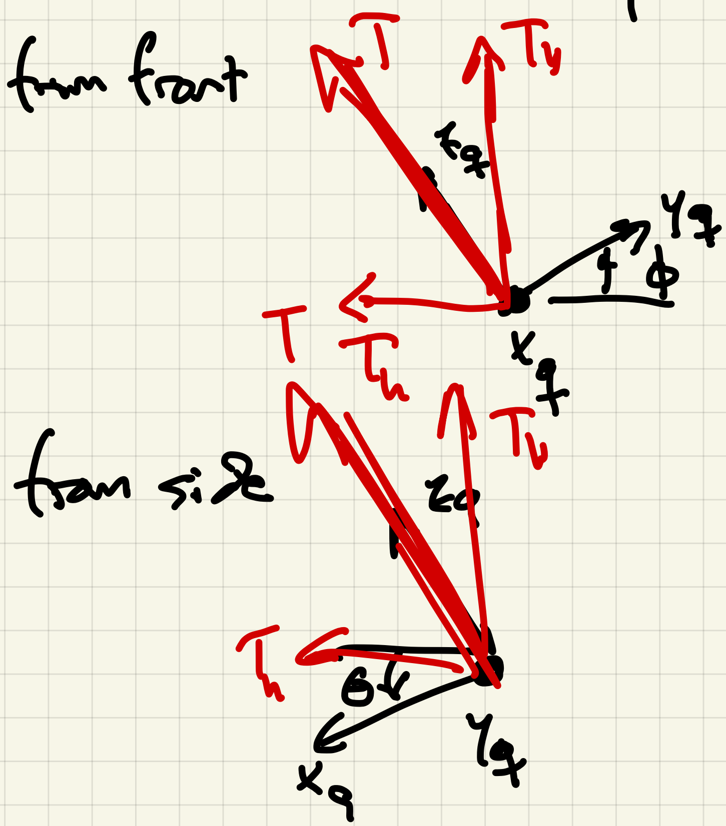




Outer Loop (LQR)

Two Things

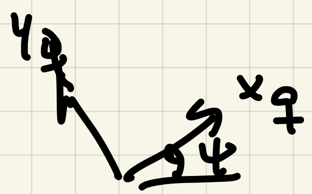
① How $\phi, \theta \rightarrow x, y$



ϕ causes acceleration in the (sort of) $-y_q$ direction

θ cause acceleration in the (sort of) $+x_q$ direction

(2) x_r, y_r and x_q, y_q are not the same thing!



Rotation difference between r and q coordinates

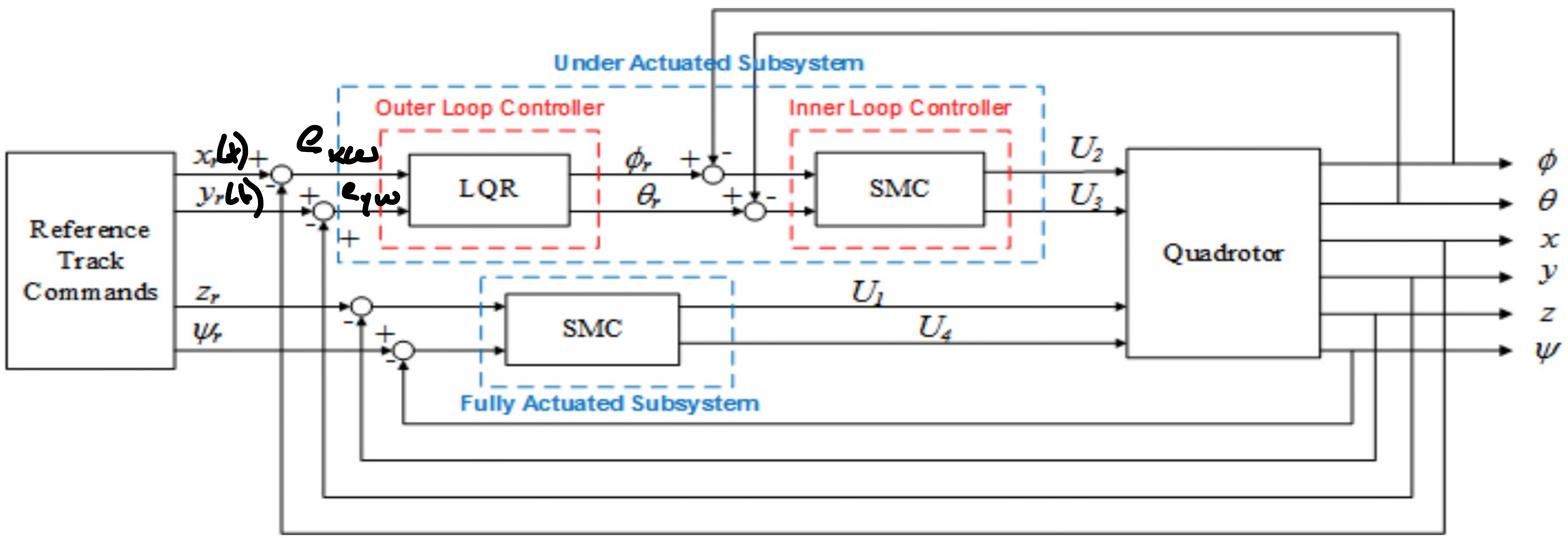
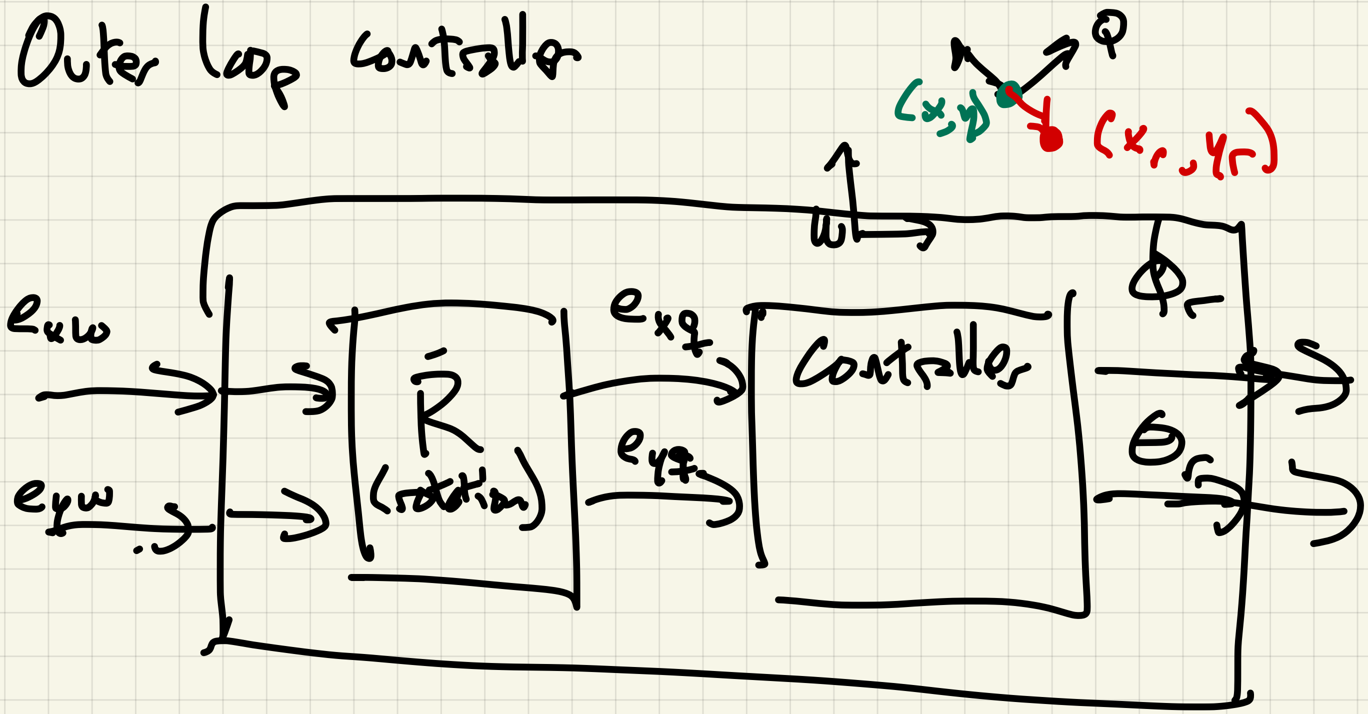


Fig. 2. UAV control system block diagram.

Ghamry, Khaled A., and Youmin Zhang. "Formation control of multiple quadrotors based on leader-follower method." In 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1037-1042. IEEE, 2015.

Outer loop controller



3 Take Home Messages

1. Loops within loops
2. Decoupling (combination of intuition and math)
3. control method can be different