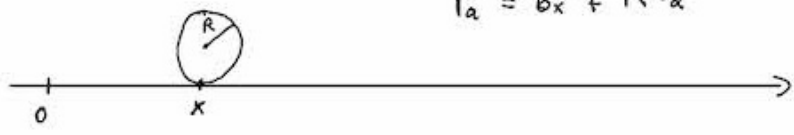


## Course Material

### Week 3: Extended Kalman Filters

3.1.	Examples for the Action in the EKF.....	2
3.2.	Examples for the Perception in the EKF.....	3
3.3.	The EKF is a weight mean .....	4
3.4.	The use of the EKF in robotics.....	5
3.5	Simultaneous Localization and Mapping (SLAM) .....	6
3.6	Observability in robotics.....	8
3.7.	Observability Rank Criterion .....	10
3.8.	Applications of the Observability Rank Criterion .....	11

### 3.1. Examples for the Action in the EKF

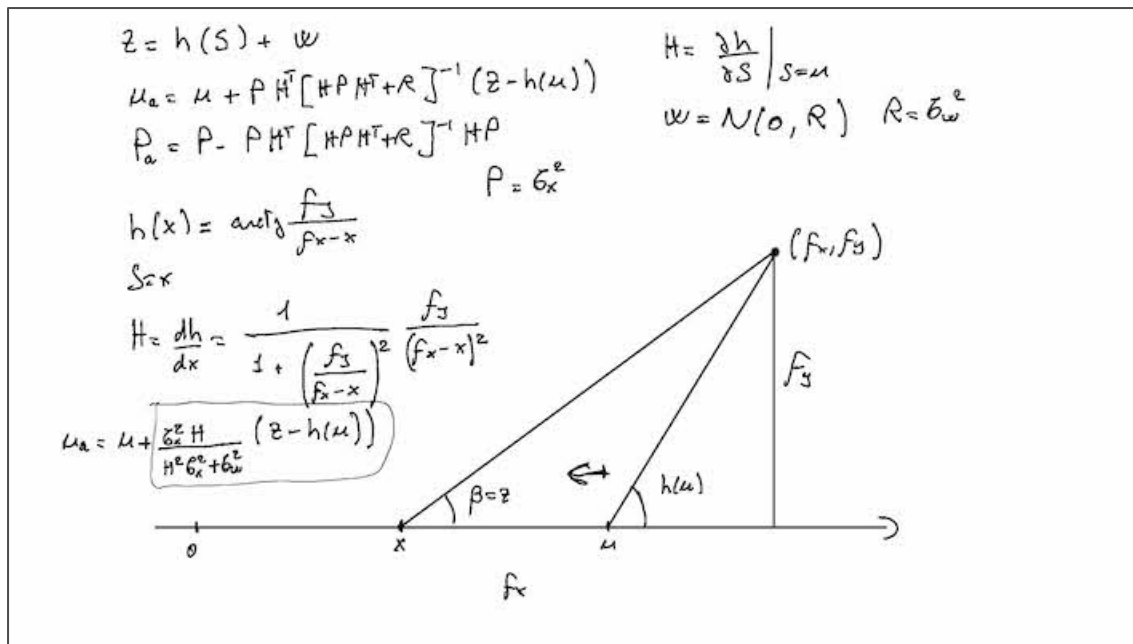
$$\begin{aligned}
 & \begin{cases} \mu_a = f(\mu, u^m) \\ P_a = F_x P F_x^T + F_u Q F_u^T \end{cases} & x = N(\mu, \sigma_x^2) & P = \sigma_x^2 \\
 & & d = N(d^m, \sigma_d^2) & Q = \sigma_d^2 \\
 & S = x & F_x = 1 & F_u = R \\
 & x_{i+1} = x_i + R d_{i+1} & \mu_a = \mu + R d^m & P_a = \sigma_x^2 + R \sigma_d^2 \\
 & & & \text{CHONG-KLS/AN}
 \end{aligned}$$


Screen 1: Encoders in 1D

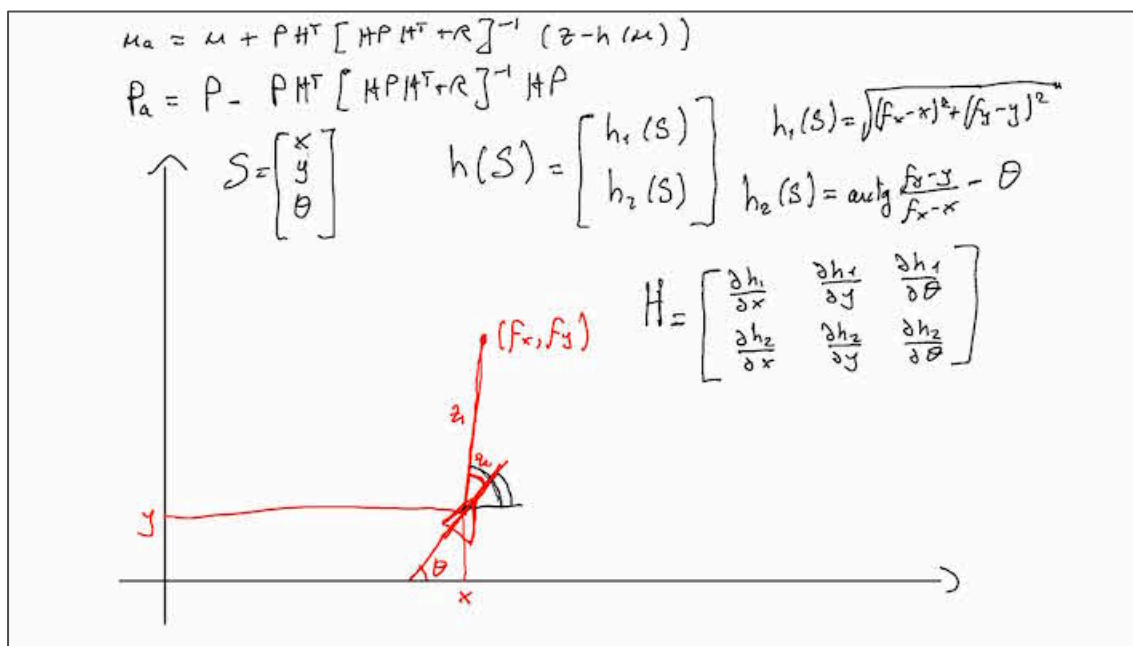
$$\begin{aligned}
 & S = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} & S_{i+1} = f(S_i, u_{i+1}) & \delta p = \frac{s_R + s_L}{2} \\
 & & \begin{cases} x_{i+1} = x_i + \delta p \cos \theta_i \\ y_{i+1} = y_i + \delta p \sin \theta_i \\ \theta_{i+1} = \theta_i + \delta \theta \end{cases} & \delta \theta = \frac{s_R - s_L}{B} \\
 & u^m = \begin{bmatrix} s_R^m \\ s_L^m \end{bmatrix} & u = N\left(\begin{bmatrix} s_R^m \\ s_L^m \end{bmatrix}, Q\right) & Q = \begin{bmatrix} \kappa^R |s_R| & 0 \\ 0 & \kappa^L |s_L| \end{bmatrix} \\
 & & & \text{CHONG-KLS/AN} \\
 & \begin{cases} \mu_a = f(\mu, u^m) \\ P_a = F_x P F_x^T + F_u Q F_u^T \end{cases} & F_{x3 \times 3} = \begin{bmatrix} 1 & 0 & -\delta p \sin \theta \\ 0 & 1 & \delta p \cos \theta \\ 0 & 0 & 1 \end{bmatrix} & F_{u3 \times 2} = \begin{bmatrix} \frac{\cos \theta}{2} & \frac{\cos \theta}{2} \\ \frac{\sin \theta}{2} & \frac{\sin \theta}{2} \\ \frac{1}{B} & -\frac{1}{B} \end{bmatrix}
 \end{aligned}$$

Screen 2: Encoders in 2D

### 3.2. Examples for the Perception in the EKF



Screen 1: Bearing of a point feature when the robot moves in 1D



Screen 2: Bearing and distance of a point features in 2D

### 3.3. The EKF is a weight mean

$$\begin{aligned}
 & x \quad z_1 \quad z_2 \quad z_1 = \mathcal{N}(x, \sigma_1^2) \quad z_2 = \mathcal{N}(x, \sigma_2^2) \\
 & \hat{x} = p_1 z_1 + p_2 z_2 \quad p_1 + p_2 = 1 \quad p_1 \propto \frac{1}{\sigma_1^2} \quad p_2 \propto \frac{1}{\sigma_2^2} \\
 & \boxed{\hat{x} = \frac{\frac{1}{\sigma_1^2} z_1 + \frac{1}{\sigma_2^2} z_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_2^2 z_1 + \sigma_1^2 z_2}{\sigma_1^2 + \sigma_2^2}} \\
 & \sigma^2 \quad \frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \quad \boxed{\frac{1}{\sigma^2} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}
 \end{aligned}$$

Screen 1: Simple weight mean

$$\begin{aligned}
 \mu_a &= \mu + P H^T [H P H^T + R]^{-1} (z - h(\mu)) \\
 P_a &= P - P H^T [H P H^T + R]^{-1} H P \\
 \mu &\rightarrow z_1 \\
 z &\rightarrow z_2 \\
 h &= \text{identity function} \quad H = 1 \\
 P &\rightarrow \sigma_1^2 \quad \hat{x} = z_1 + \sigma_1^2 \cdot 1 [\sigma_1^2 + \sigma_2^2]^{-1} (z_2 - z_1) = \\
 R &\rightarrow \sigma_2^2 \quad z_1 + \frac{\sigma_1^2 z_2 - \sigma_1^2 z_1}{\sigma_1^2 + \sigma_2^2} = \frac{z_1 \sigma_2^2 + z_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} \\
 \mu_a &\rightarrow \hat{x} \\
 P_a &\rightarrow \hat{\sigma}^2 \quad \hat{\sigma}^2 = \sigma_1^2 - \sigma_1^2 [\sigma_1^2 + \sigma_2^2]^{-1} \sigma_1^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}
 \end{aligned}$$

Screen 2: EKF perception for a special case: 1D and  $h$  is the identity function

### 3.4. The use of the EKF in robotics

$$\begin{cases} \mu_a = f(u, u^m) \\ P_a = F_x P F_x^T + F_u Q F_u^T \end{cases} \quad \text{ACTION}$$

$$\begin{cases} \mu_a = \mu + P H^T [H P H^T + R]^{-1} (z - h(\mu)) \\ P_a = P - P H^T [H P H^T + R]^{-1} H P \end{cases} \quad \text{PERCEPTION}$$

- 1) Introduce a suitable frame and the state to be estimated
- 2) Derive the analytical expression of the functions "f" and "h"
- 3) Compute the Jacobians  $F_x = \frac{\partial f}{\partial s}$   $F_u = \frac{\partial f}{\partial u}$   $H = \frac{\partial h}{\partial s}$

- ▶ LOCALIZATION
- ▶ SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)
- ▶ COOPERATIVE LOCALIZATION
- ▶ SELF-CALIBRATION

Screen 1: Basic steps to implement an EKF and important estimation problems in robotics

### 3.5 Simultaneous Localization and Mapping (SLAM)

Diagram illustrating the 2D case with  $N$  point features. The state vector  $S$  is defined as:

$$S = [x, y, \theta, f_1^x, f_1^y, \dots, f_N^x, f_N^y]^T$$

The function  $f$  maps the state to the next pose, and the function  $h$  maps the state to the feature measurements. The Jacobian matrices  $F_x$  and  $F_u$  are shown, along with the measurement Jacobian  $H$ .

$$F_x = \begin{bmatrix} F_{3 \times 3}^R & 0_{3 \times 2N} \\ 0_{2N \times 3} & I_{2N \times 2N} \end{bmatrix}$$

$$F_u = \begin{bmatrix} F_{3 \times 3}^R & 0_{3 \times 2N} \\ 0_{2N \times 3} & I_{2N \times 2N} \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial \theta} & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$h(S) = h(x, y, \theta, f_1^x, f_1^y, \dots, f_N^x, f_N^y)$$

$$F_{x(3+2N) \times 3} = \begin{bmatrix} F_{3 \times 3}^R & 0_{3 \times 2N} \\ 0_{2N \times 3} & I_{2N \times 2N} \end{bmatrix}$$

$$F_{u(3+2N) \times 3} = \begin{bmatrix} F_{3 \times 3}^R & 0_{3 \times 2N} \\ 0_{2N \times 3} & I_{2N \times 2N} \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial \theta} & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

Screen 2: Basic equations for the 2D case with  $N$  point features: the function  $f$  and  $h$  and their Jacobians

Diagram illustrating the matrices to implement the EKF and computational cost of the action.

$$S = [s^R, f^1, f^2, \dots, f^N]$$

$$F_x = \begin{bmatrix} F_x^R & 0 \\ 0 & I \end{bmatrix} \quad F_u = \begin{bmatrix} F_u^R \\ 0 \end{bmatrix} \quad P = \begin{bmatrix} P_R & P_{Rn} \\ P_{Rn}^T & P_n \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{\partial h}{\partial s^R} & 0 & 0 & \dots & \frac{\partial h}{\partial f^N} & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \mu_a = f(\mu, u) & \text{cost independent of } N \\ P_a = F_x P F_x^T + F_u Q F_u^T = \begin{bmatrix} F_x^R P_R F_x^{RT} & F_x^R P_{Rn} F_x^{RT} \\ P_{Rn}^T F_x^T & P_n \end{bmatrix} + \begin{bmatrix} F_u^R Q F_u^{RT} & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix}$$

$\hookrightarrow$  cost depend linearly on  $N$

$O(N)$

Screen 1: Matrices to implement the EKF and computational cost of the action

$$\mu_a = \mu + PH^T [HPH^T + R]^{-1} (z - h/\mu) \quad \text{cost is } O(N)$$

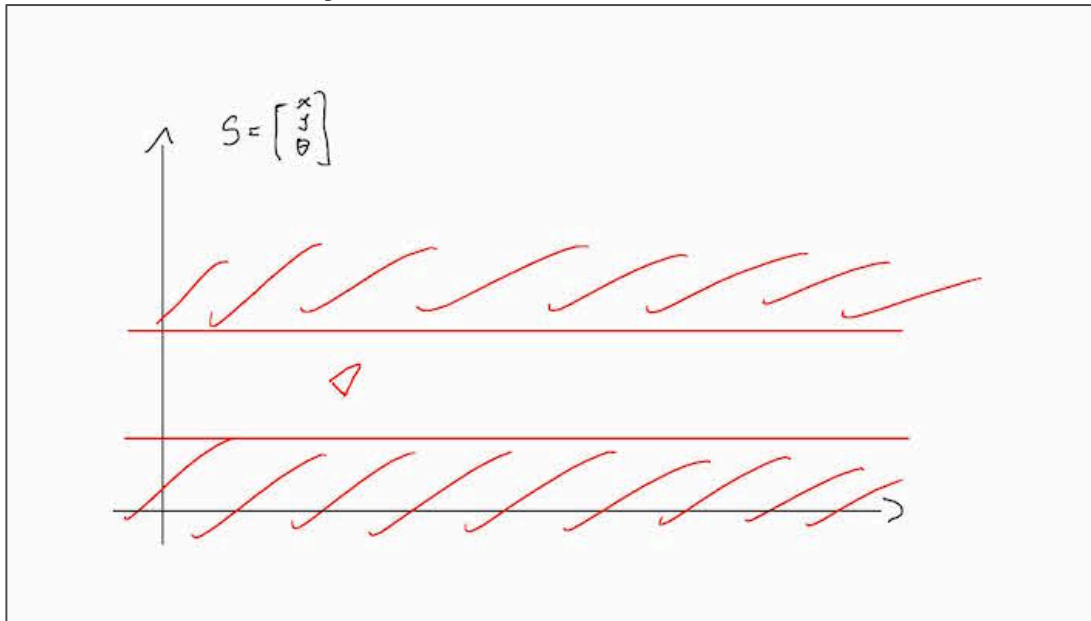
$$P_a = P - PH^T [HPH^T + R]^{-1} HP \quad \text{cost is } O(N^2)$$

$$H = [H^R, \emptyset \emptyset, \dots, H^J, \emptyset \emptyset]$$

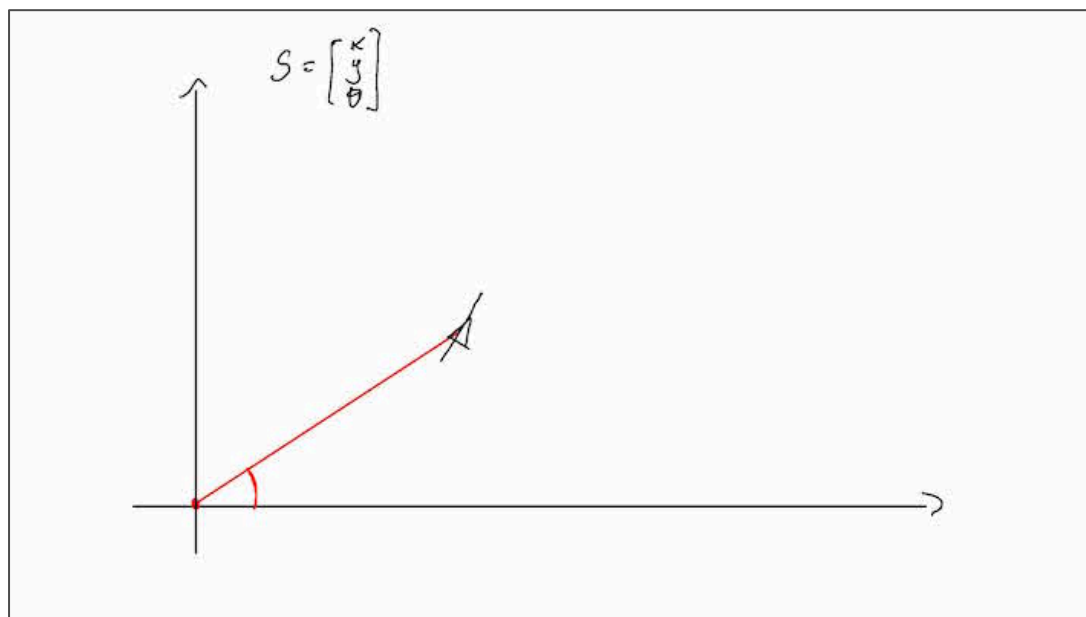
$$\begin{bmatrix} P_R & P_{Rn} \\ P_{Rn}^T & P_n \end{bmatrix} \begin{bmatrix} H^{RT} r \\ 0 \\ 0 \\ H^J r \\ 0 \end{bmatrix} \quad O(N)$$

Screen 2: Computational cost of the perception

### 3.6 Observability in robotics

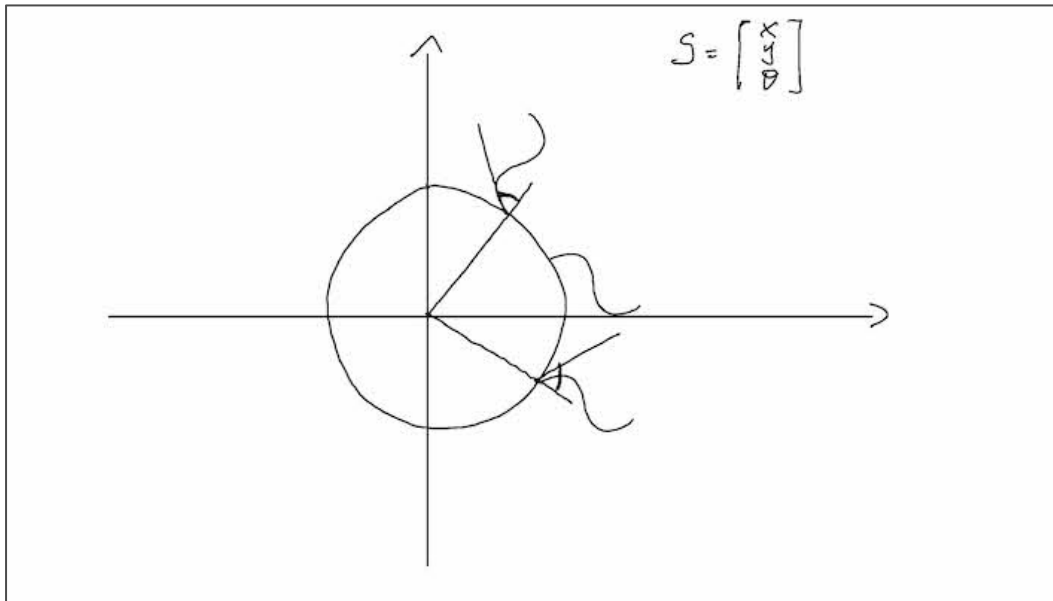


Screen 1: Uniform hallway: intuitively, we cannot estimate the x-coordinate

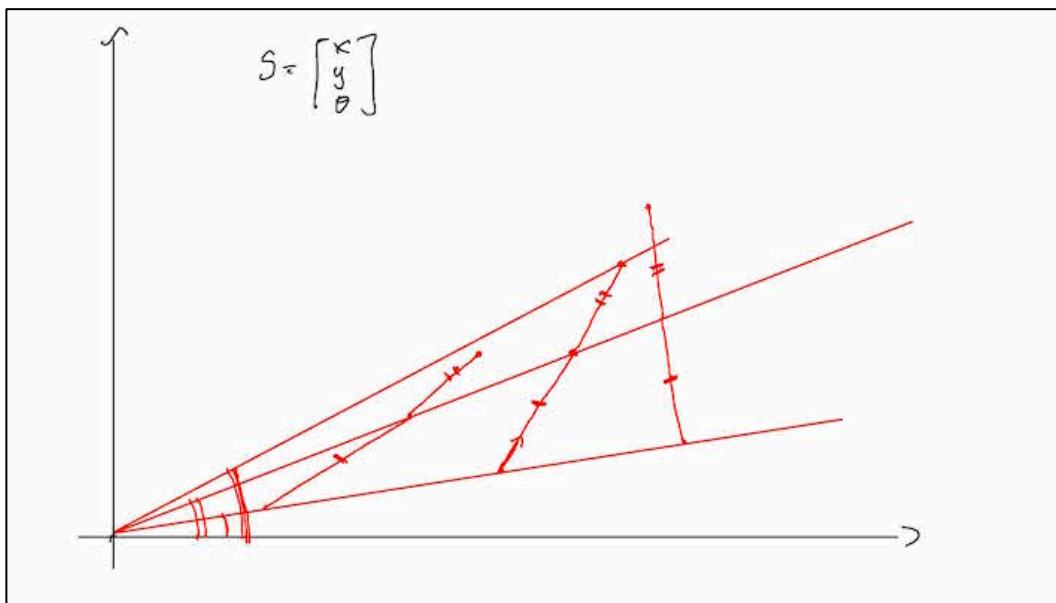


Screen 2: 2D case with range sensor that provides the distance from the origin





Screen 3: The 2 trajectories (the upper and the lower) provide the same measurements



Screen 4: All the three trajectories provide a different third bearing measurement

### 3.7. Observability Rank Criterion

OBSERVABILITY RANK CRITERION  
HERMAN KREMER 1977

$$\begin{cases} x_{i+1} = x_i + \delta s \cos \theta_i \\ y_{i+1} = y_i + \delta s \sin \theta_i \\ \theta_{i+1} = \theta_i + \delta \theta \end{cases} \quad \begin{aligned} \delta s &= v \delta t \\ \delta \theta &= \omega \delta t \end{aligned}$$

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases}$$

$$\dot{S} = f_0(S) + \sum_{i=1}^m u_i f_i(S)$$

$$f_0(S) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad m=2 \quad \begin{aligned} u_1 &= v \\ u_2 &= \omega \end{aligned}$$

$$f_1(S) = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} \quad f_2(S) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Screen 1: Continuous time model for a wheeled robot in 2D (the unicycle)

$$\dot{S} = f_0(S) + \sum_{i=1}^m u_i f_i(S) \quad h(S)$$

$$L^0 h = h$$

$$m+1 \text{ first order} \quad L_0^1 h = (D \cdot h) \cdot f_0 \quad \dots \quad L_i^1 h = (D \cdot h) \cdot f_i$$

$$(m+1)^2 \text{ second order} \quad L_{ij}^2 h = (D L_i^1 h) \cdot f_j$$

$$\Omega^0 = \text{span} \{ D h \}$$

$$\Omega^1 = \text{span of the gradients of all the Lie derivatives up to order 1}$$

Screen 2: Lie derivatives and the observable codistribution

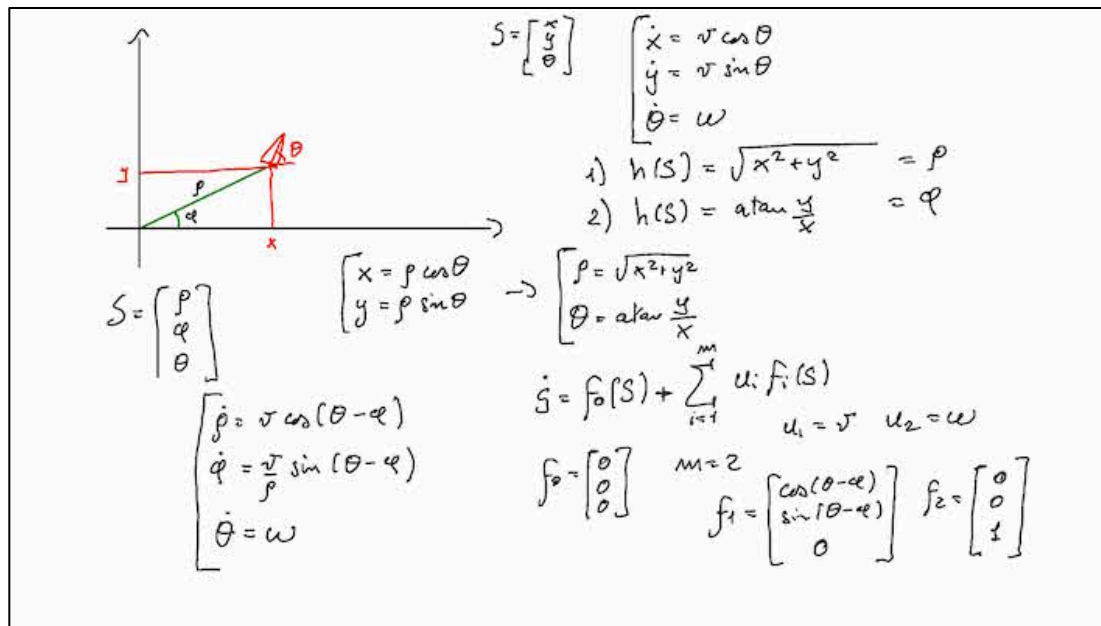
$$\Omega^0 \subset \Omega^1 \subset \Omega^2 \quad \dots$$

$$n = \text{dimension of } S \quad \dim(\Omega^k) \leq n$$

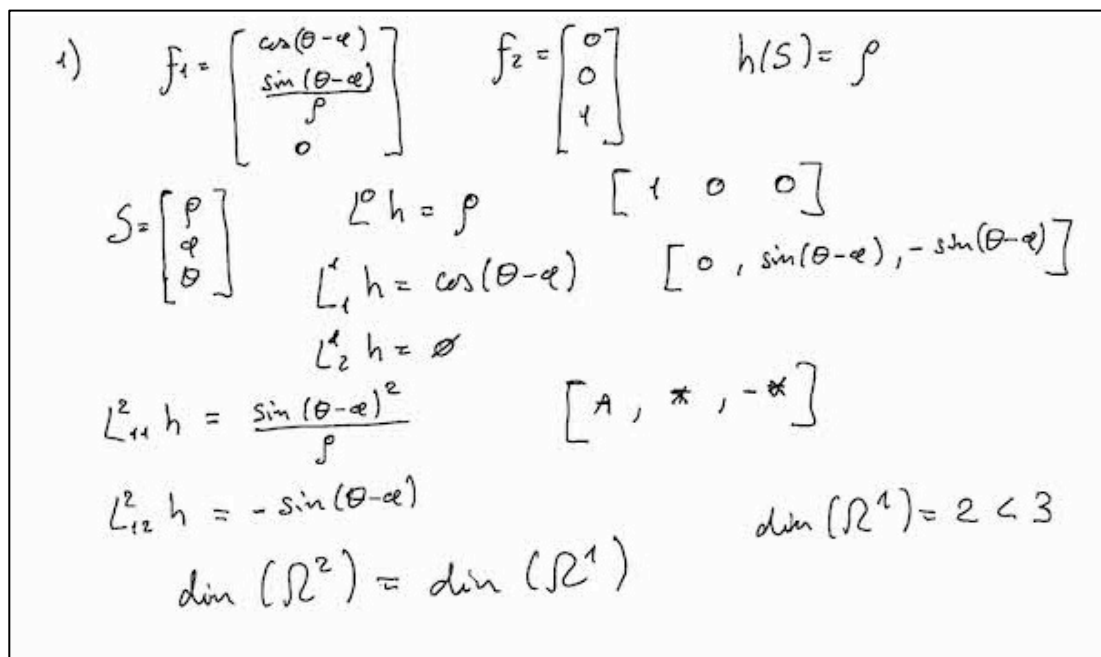
$$\dim(\Omega^j) = \dim(\Omega^{j+1}) \leq n$$

Screen 3: Dimension of the observable codistribution

### 3.8. Applications of the Observability Rank Criterion



Screen 1: 2D robot (satisfying the unicycle dynamics) in polar coordinates



Screen 2: The case when the sensor provides the distance from the origin. The dimension of the observable codistribution is  $2 < 3$  and the system is non-observable

$$\begin{aligned}
 2) \quad f_1 &= \begin{bmatrix} \cos(\theta - \varphi) \\ \frac{\sin(\theta - \varphi)}{\rho} \\ 0 \end{bmatrix} \quad f_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad h(S) = \varphi \\
 L^0 h &= \varphi \quad [0 \quad 1 \quad 0] \\
 L^1_1 h &= \frac{\sin(\theta - \varphi)}{\rho} \quad \left[ -\frac{\sin(\theta - \varphi)}{\rho^2}, -\frac{\cos(\theta - \varphi)}{\rho}, \frac{\cos(\theta - \varphi)}{\rho} \right] \\
 L^1_2 h &= \emptyset \\
 L^2_{12} h &= \frac{\cos(\theta - \varphi)}{\rho} \quad \left[ -\frac{\cos(\theta - \varphi)}{\rho^2}, \frac{\sin(\theta - \varphi)}{\rho}, -\frac{\sin(\theta - \varphi)}{\rho} \right] \\
 \dim(\Omega^2) &= 3
 \end{aligned}$$

Screen 3: The case when the sensor provides the bearing of the robot. The dimension of the observable codistribution is 3 and the system is observable