Mobile Robots Lab
Localization using magnets

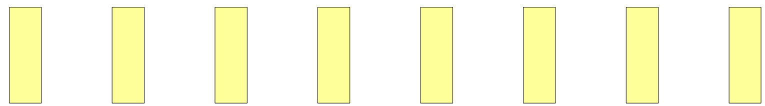
Presentation

The robot and the magnet sensor

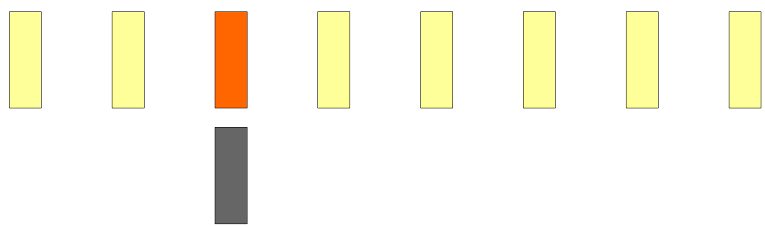
The reed switch

A normally open reed switch.
In a sufficiently intense magnetic field, the switch closes.

The magnet sensor

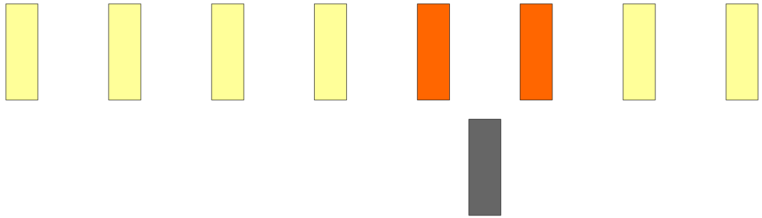


No magnet in the vicinity of the reed switches: all are open



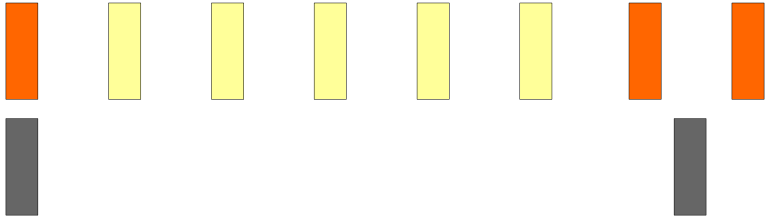
A magnet is right under reed switch 3, which is closed.

The magnet sensor



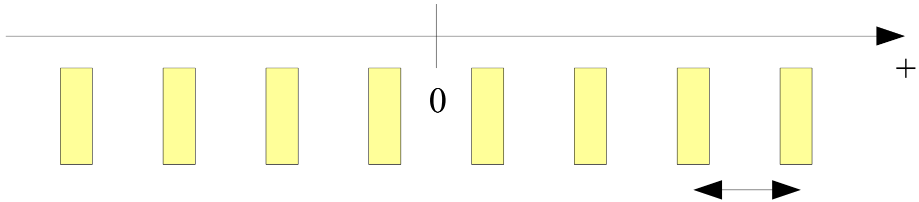
A magnet is right under switches 5 and 6, which are closed.

The measurement vector is a 2 vector



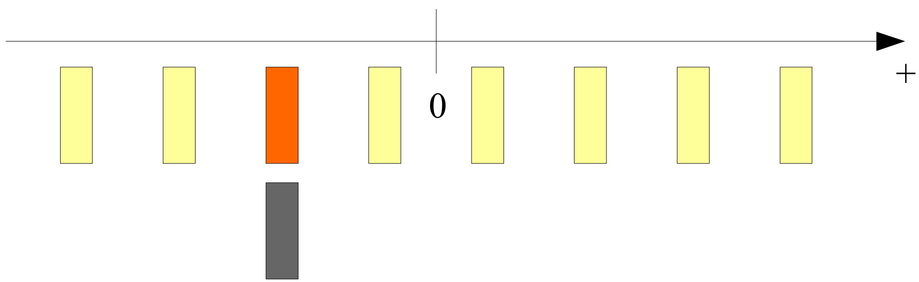
A magnet is right under switche 1, another under 7 and 8

The magnet sensor measurements



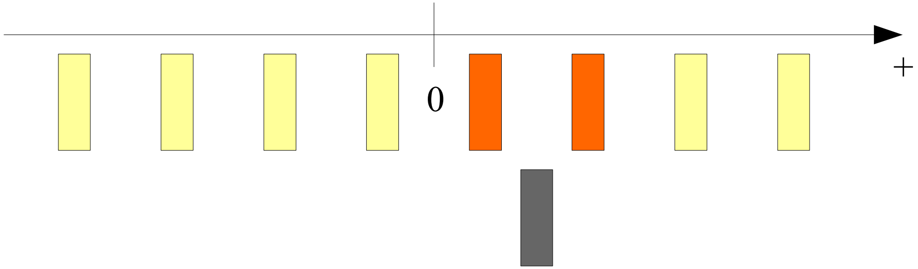
No measurement

This 8 binary states becomes a measurement when we define a zero position in the center and then we multiply by a constant Δ

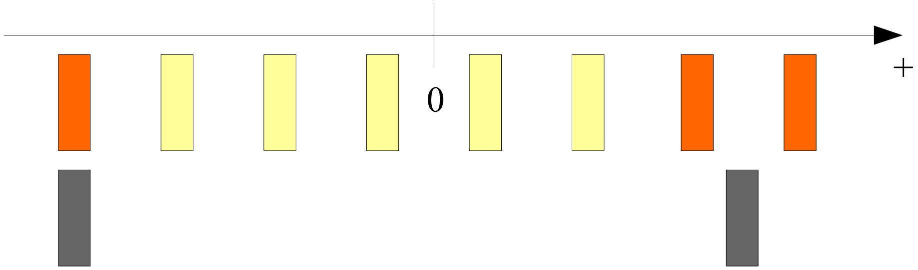


Measurement is -1.5 Δ

The magnet sensor measurements



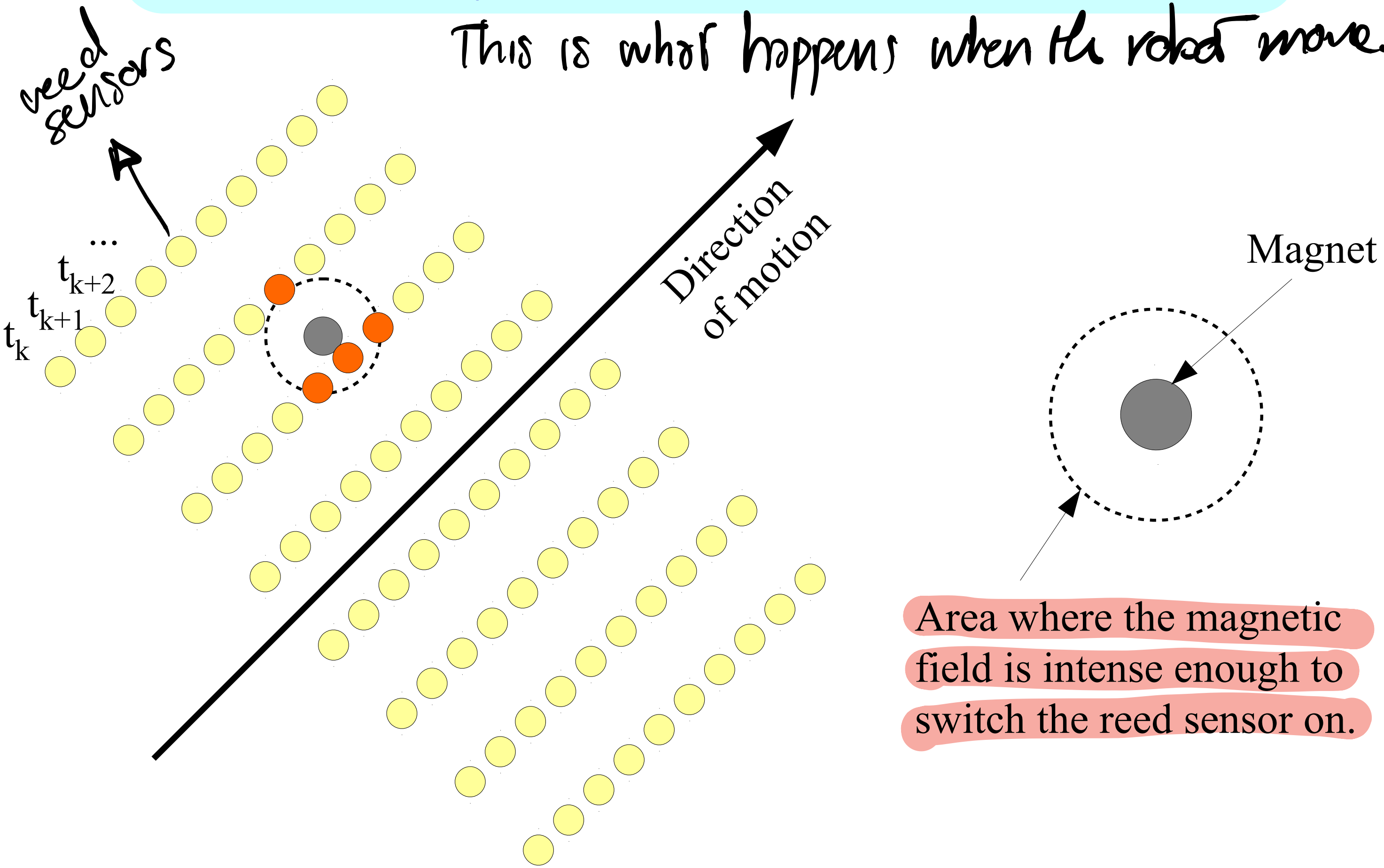
Measurement is $+\Delta$



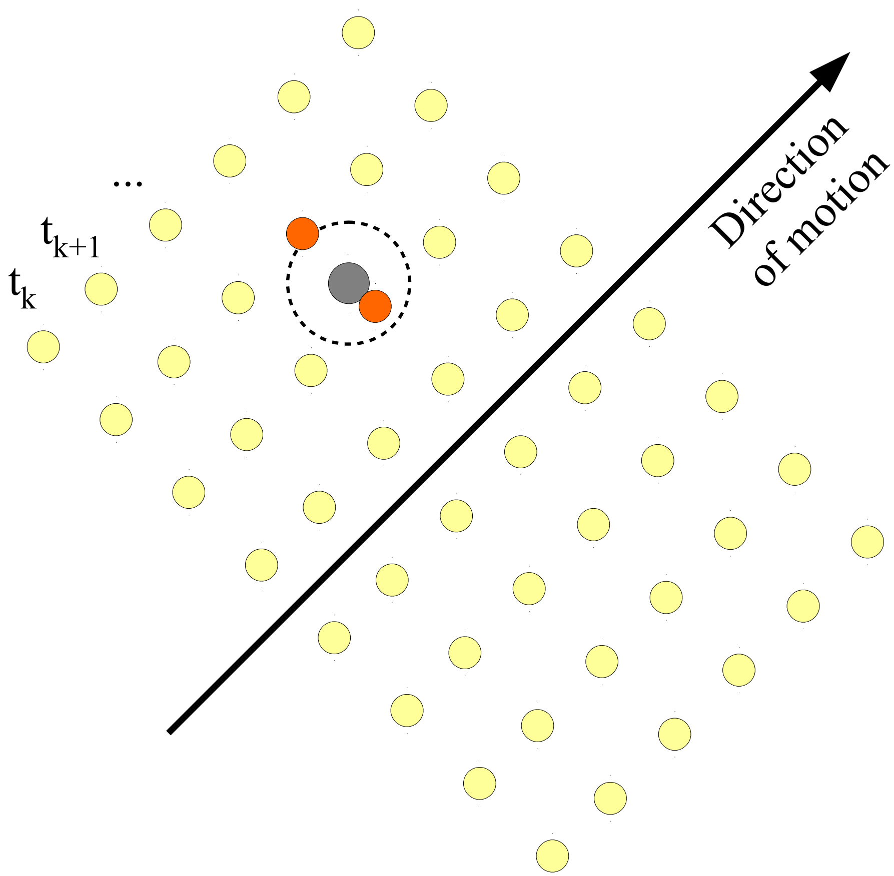
Two measurements: -3.5Δ and $+3\Delta$

Sensor passing over a magnet

This is what happens when the robot moves



Sensor passing over a magnet
(lower sampling rate)



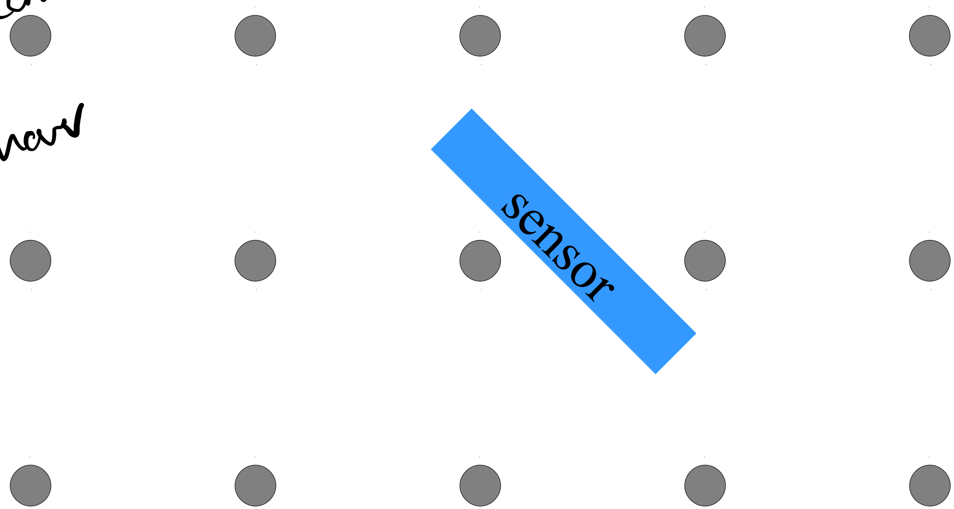
About the magnet sensor

- The system design parameters are:
  - Magnet field intensity.
  - Inter-magnet distance
  - Reed switch sensitivity.
  - Reed switch number/spacing and sensor length.
- The sensor has been designed in such a way that:
  - When passing over a magnet, either one or two reed switches are activated.
  - When the robot moves, the sensor “cannot avoid crossing magnets”

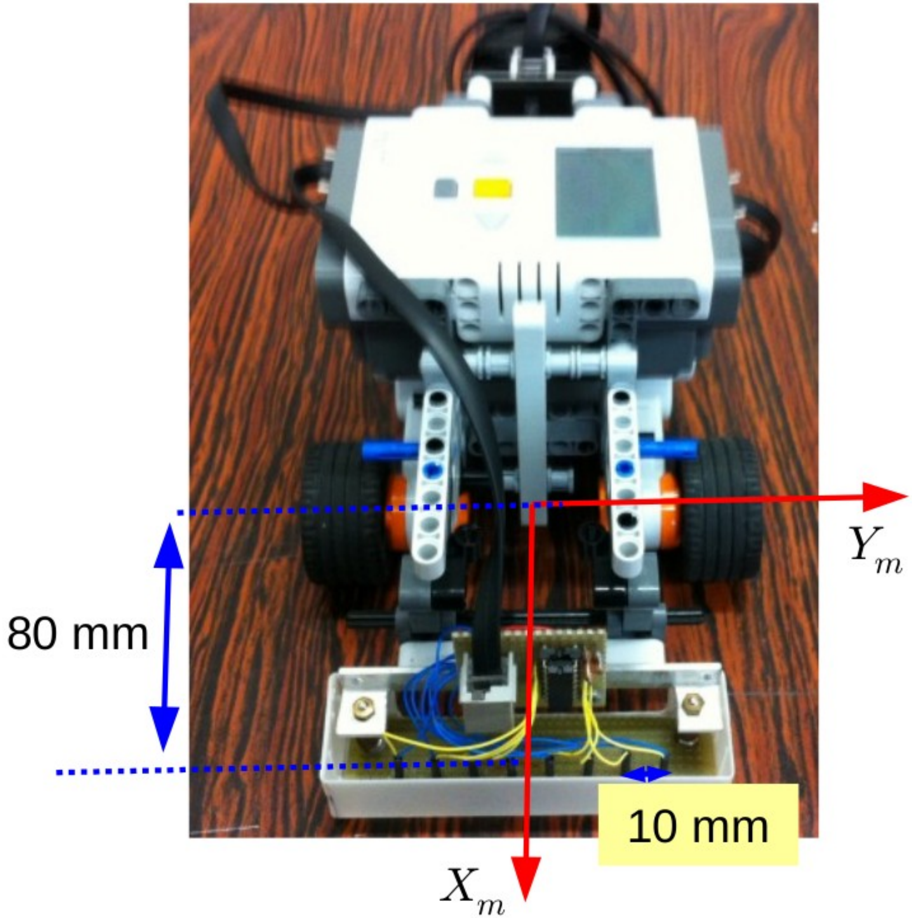
System characteristics

- 8 reed switches.
- 10 mm interval between reed switches
- 55 mm interval between magnets, arranged in a regular square pattern.

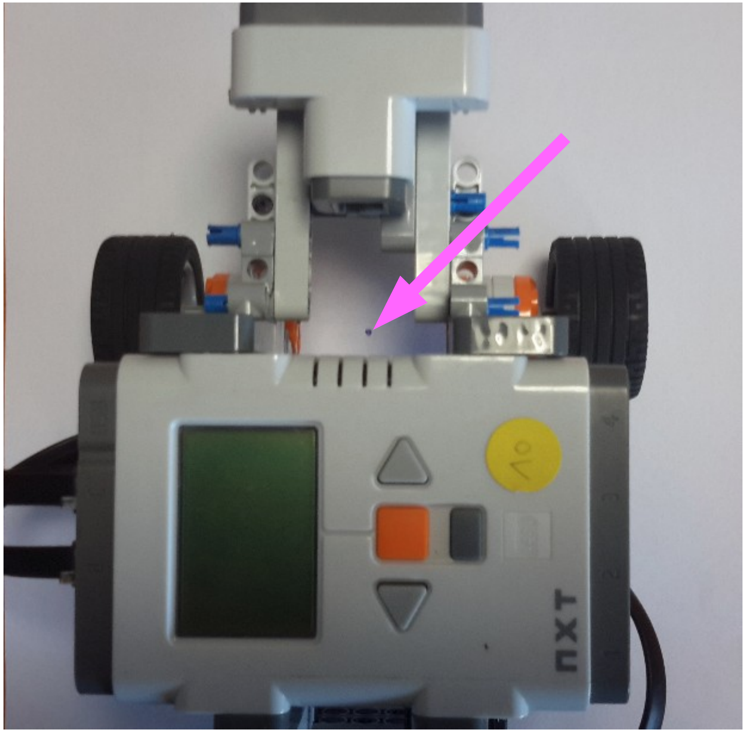
magnets are not signed beacons so we can't know which magnet has activated the sensor.



Robot and sensor



Initial positioning of the robot



Initial state on the axle of the two wheels → not very precise.

The center point of the fixed wheel axle is put above a dot painted at (0,0).

Robot characteristics

- Encoders:
  - 360 dots per wheel revolution.
  - "Dumbed down" to 45 dots per revolution to make the problem a bit more challenging.
- Recording frequency:
  - 20 Hz. → look at the data at 20Hz when determining the noise variance
  - We will work at 5 Hz.

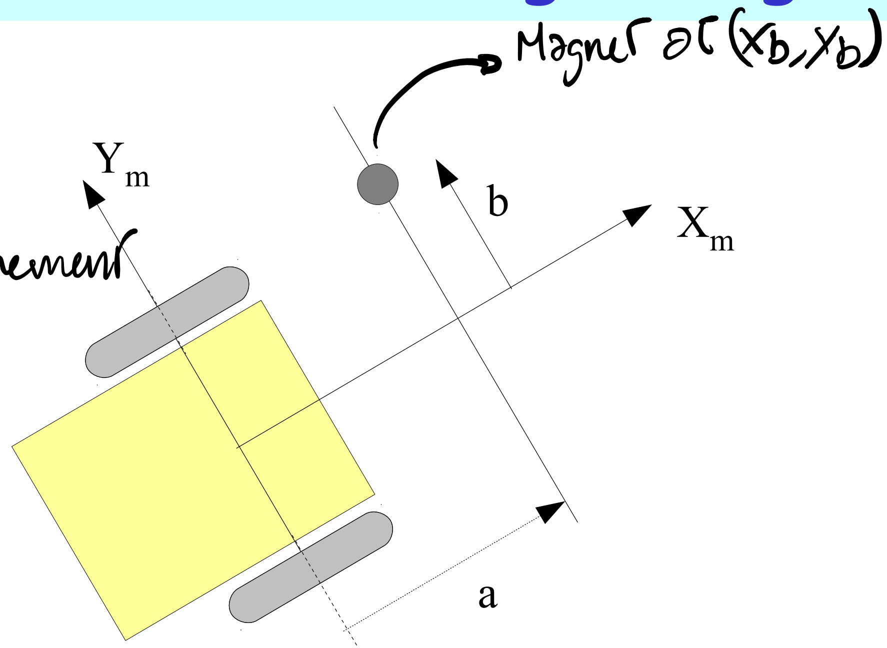
There is a file which contains the frequency parameter

change it to 1 → to see at 20Hz

change it to 4 → to work at 5 Hz

The robot detecting a magnet

It's a big error to think that the noise measurement is determined by trial and error.



- Express in plain English what the sensor measures.
- Write the measurement equation

(1) The sensor is showing wetter the magnetic field surrounding it is over a certain treshold.
IT MEASURES the coordinates of the magnetic field w.r.t. the robot frame, which is located at the center of the wheel axle.

The a coordinate is always 80, because the sensors are always at distance a.

In the equations of the Kalman filter we have $y-\hat{y}$

$\rightarrow \underset{2\times1}{y} = 80$ but usually $\hat{y} \neq 80$

$$y = g(x) = \begin{bmatrix} a \\ b \end{bmatrix}$$

Surely we're not measuring angles. And we're also not measuring distances
$\rightarrow$ We're measuring <u>coordinates</u>.

${}^{o}M$ IS KNOWN
${}^{m}M$ is measured
$\rightarrow = g(x)$

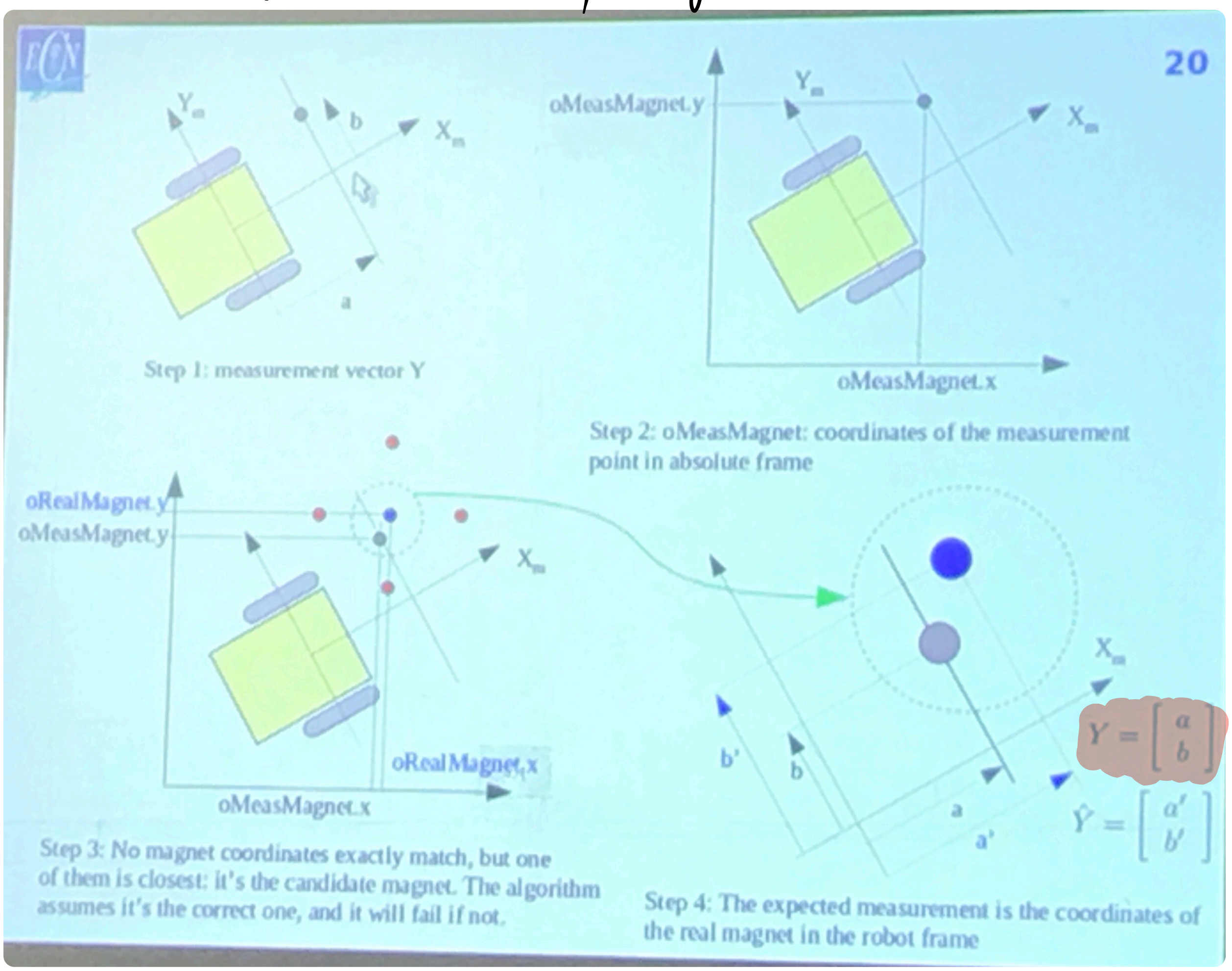
$$\Rightarrow {}^{o}T_{m} = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ \emptyset & \emptyset & 1 \end{bmatrix} = \left[\begin{array}{c|c} R & P \\ \hline 0\ 0\ 0 & 1 \end{array}\right]$$

We're actually going to use the inverse

$$ {}^{m}T_{o} = \left[\begin{array}{c|c} R^T & -R^T P \\ \hline 000 & 1 \end{array}\right]$$

$$\rightarrow {}^{o}M = \begin{bmatrix} x_M \\ y_M \\ 1 \end{bmatrix} \qquad \rightarrow {}^{m}M = {}^{m}T_{o}\,{}^{o}M$$

x_M and y_M will be multiples of 55.



Take the measurement, express it in absolute coordinates, determine the candidate magnet and project its coordinates onto the robot frame. → this is the expected measurement

Let z be a gaussian vector $z^T P^{-1} z$ follows a χ^2-law.
The shape of the χ^2 distribution depends on the dimension of z (number of lines). The threshold is obtained in Matlab with the "chi2inv" function : chi2inv(p, v).

$p = 0{,}95$ to $0{,}99$ tipically , $v = \dim(y)$

We use $z = y - \hat{y}$, $P = CPC^T + Q_y$

The value of the threshold depends on the dimension of the output vector. NO TRIAL & ERROR

$$\begin{bmatrix} v \\ w \end{bmatrix} = JTC \begin{bmatrix} \dot{q}_r \\ \dot{q}_L \end{bmatrix}$$

↳ Joint To Cartesian

$$\begin{bmatrix} \Delta D \\ \Delta \theta \end{bmatrix} = JTC \begin{bmatrix} \Delta q_R \\ \Delta q_L \end{bmatrix}$$

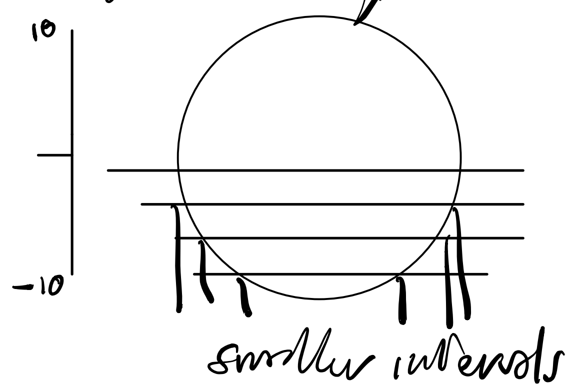
X = evolutionModel (X, u)

↳ $x_{k+1} = f(x, u)$

Compute $C = \frac{\partial g_\theta}{\partial X} = \frac{\partial}{\partial X}\begin{bmatrix} \cos\theta({}^o x_m - x) + \sin\theta({}^o y_m - y) \\ \cos\theta({}^o y_m - y) - \sin\theta({}^o x_m - x) \end{bmatrix}$

$$= \begin{bmatrix} -\cos\theta & -\sin\theta & \left(-\sin\theta({}^o x_{mag} - x) + \cos\theta({}^o y_{mag} - y)\right) \\ \sin\theta & -\cos\theta & \left(-\sin\theta({}^o y_{mag} - y) - \cos\theta({}^o x_{mag} - x)\right) \end{bmatrix}$$

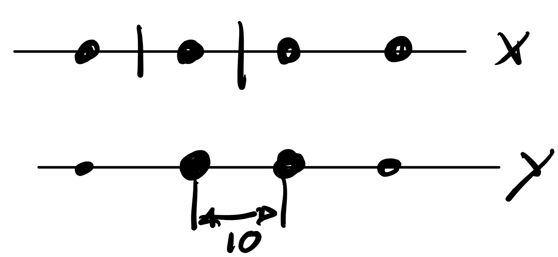
The distance between the revolutions vary because of the frequency



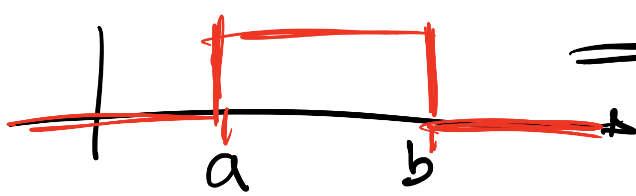
$$\sigma_x^2 = \frac{20^2}{12} \quad \text{eg.}$$

Along y is completely different:

$$\sigma_y^2 = \frac{10^2}{12} \quad \text{fig } 7$$



Uniformed Noise Variance:



$$\Longrightarrow \sigma^2 = \frac{(b-a)^2}{12}$$

If we set δ tuning very low, P increases too slowly

→ $\underbrace{CPC^T}_{\text{variance of } \hat{y}}$ is also too low so you are trusting your $\hat{y}$ much more than you trust y, so each y that has a discrepancy from $\hat{y}$ gets REJECTED

→ $CPC^T + \underbrace{Q_y}_{\text{variance of } y}$ is also too low, so its inverse is too high → an high gain that goes to multiply the mahalanobis distance. So you accept pretty anything, and that is why all the red dots are below threshold

Mahalanobis distance =

$$\rightarrow \underbrace{(y-\hat{y})^T}_{\text{innov}} \underbrace{(CPC^T + Q_y)^{-1}}_{\text{variances of } \hat{y} \quad y} \underbrace{(y-\hat{y})}_{\text{innov}}$$

⇒ To compute the threshold for the system use chi2inv(% of acceptable measurements, dimension of y)

LAB 2

New state: $\begin{bmatrix} x \\ y \\ \theta \\ r_c \\ r_m \end{bmatrix}$

New evolution model:

$$\boxed{r_{k+1} = r_k + \underset{\alpha_k}{\text{some noise}}}$$

$$Q_\alpha = \left[\begin{array}{c|cc} \varnothing_{3\times3} & \multicolumn{2}{c}{\varnothing_{3\times2}} \\ \hline \varnothing_{2\times3} & \sigma^2 & 0 \\ & 0 & \sigma^2 \end{array}\right]$$

$\longrightarrow$ controls the behaviour of the wheel radius

So:

σ^2 small $\longrightarrow$ r is low-pass filtered (low noise) but the radius will only be able to track slow variation.
(NOT ILLUSTRATED in the lab)

Also $\underline{P_{init}}$ is affected:

$$P_{init} = \begin{bmatrix} \sigma_x^2 & & & & \varnothing \\ & \sigma_y^2 & & & \\ & & \sigma_\theta^2 & & \\ & & & \sigma^2_{r\,init} & \\ \varnothing & & & & \sigma^2_{r\,init} \end{bmatrix}$$

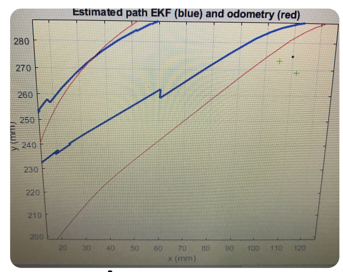
the $\sigma^2_{r\,init}$ has to be of the order of magnitude of the initial error.

When I need to identify a parameter constant or slowly evolving, we add it to the state vector and make it evoluting adding some noise.

→ (e.g.) Car Tyres which lower their pressure based on velocity, temperature ...

TO-DO-LIST point 6

b) Two loops has a sharp correction at (60, 260), why?



Looking at figure 3-4 we can clearly see the sudden variation on x's standard deviation: we also see that from $t = 14s$ to $t = 17s$ there are no measurements available → The robot goes blind for 3 seconds and when the measurement comes it suddently corrects the trajectory.

Figure 3 → standard deviation in absolute frame

Figure 4 → " " in robot frame

c) In diag 45 degree we have two diverging lines, why?

Variant of Kalman filter equations

- The program uses a slight variant of the Kalman filter equations, where the input u is assumed to be disturbed by noise:

$$X_{k+1} = f(X_k, U_k^*) + \alpha_k$$

$U_k^* = U_k + \beta_k$ The measured input is affected by an additive noise

$$P_{k+1} = A_k\, P_k\, A_k^T + B_k\, Q_\beta\, B_k^T + Q_\alpha$$

The error propagation equation is the only change.

$$A_k = \frac{\partial f}{\partial X} \qquad B_k = \frac{\partial f}{\partial U}$$

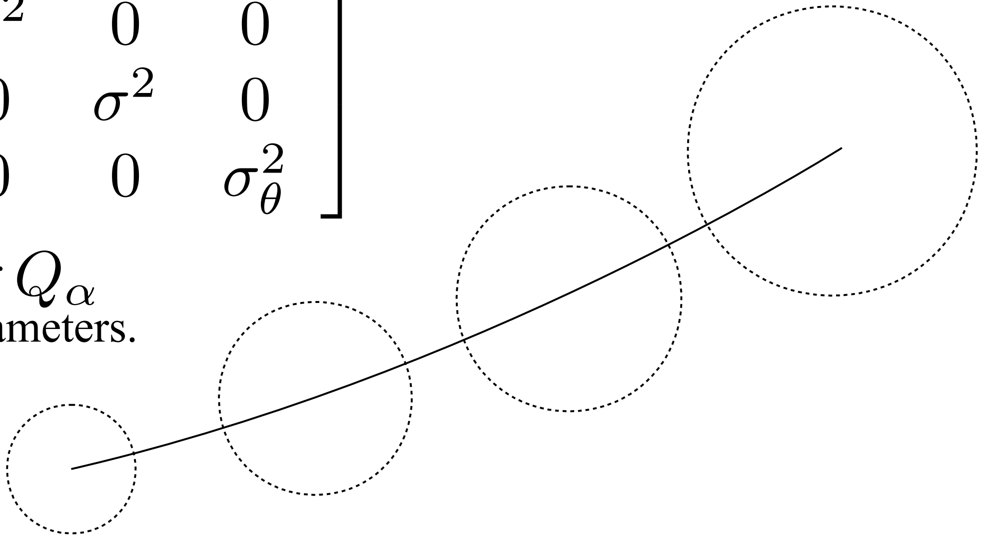
See paragraph 5.2 of the “book” form of the localization class material for the equations.

Evolution of uncertainty (standard form of the equations)

$$P_{k+1} = A_k \, P_k \, A_k^T + Q_\alpha$$

$$Q_\alpha = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

A logical for for Q_α
Two tuning parameters.



Assuming the initial uncertainty is the same in x and y, the uncertainty ellipse in the x-y plane starts as a circle and remains a circle during successive odometry phases (no update phase).

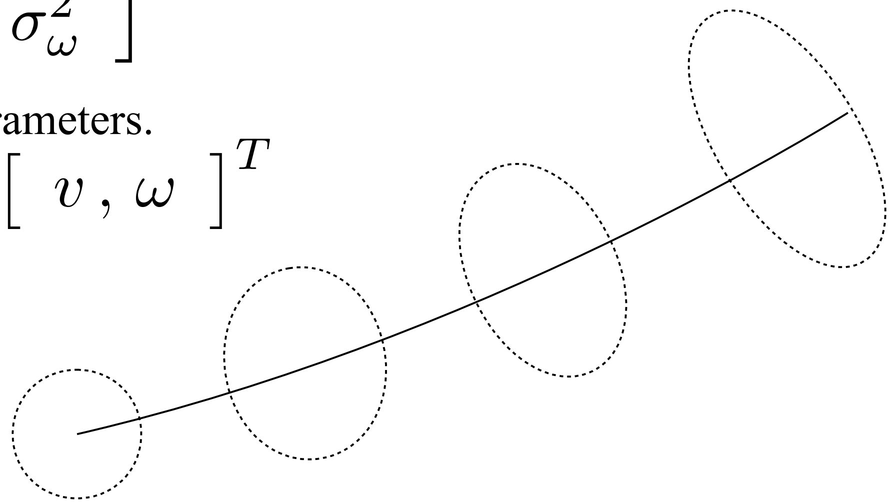
Evolution of uncertainty (form with noisy input)

$$P_{k+1} = A_k \, P_k \, A_k^T + B_k \, Q_\beta \, B_k^T$$ (Q_α has been set to zero)

$$Q_\beta = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix}$$

Still two tuning parameters.

Remember: $u = \left[\; v \; , \; \omega \; \right]^T$



Now the uncertainty ellipse orients with the motion of the robot. The result is closer to the way errors actually evolve during odometry.

$$\underset{n\times 1}{3\times 1}\ \hat{X}_{K+1/K} = f(\hat{X}_{K|K}, U_K) = \begin{bmatrix} x_K + \Delta D_K \cos\vartheta_K \\ y_K + \Delta D_K \sin\vartheta_K \\ \vartheta_K + \Delta\vartheta_K \end{bmatrix}$$

ODOMETRY

→ Estimate of the state vector 3

$$\underset{p\times 1}{2\times 1}\ U_K = \begin{bmatrix} \Delta D_K \\ \Delta\vartheta_K \end{bmatrix}$$

This is the input of the system measured from encoders (sensors) 1

$$\underset{\substack{3\times 3\\ n\times n}}{A_K} = \partial f/\partial X, \quad \underset{\substack{3\times 2\\ n\times p}}{B_K} = \partial f/\partial U$$

} linear approx of system equation. 2

measurements

$$\underset{\substack{2\times 1\\ m\times 1}}{Y} = g_B(x) = \begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow \underset{\substack{2\times 2\\ m\times m}}{Q_r} = \text{quality of the measurements}$$

5

Gain

$$\underset{\substack{2\times 3\\ m\times n}}{G_K} = \frac{\partial g_B}{\partial x}$$

6

$$\underset{\substack{3\times 3\\ n\times n}}{P_K} = \text{quality of the state estimate}$$

4

STEPS OF THE ALGORITHM

↓

After computing the gain update X and P.

The input noise in the lab

$$\begin{cases} v = (r_r\dot{q}_r + r_l\dot{q}_l)/2 \\ \omega = (r_r\dot{q}_r - r_l\dot{q}_l)/e \end{cases}$$

There is a linear relation between v, w and the rotation speed of the wheels.

So Q_β can be written as a function of the covariance matrix of errors on $\dot{q}_r$ and $\dot{q}_l$.

$$Q_\beta = K\, Q_{\dot{q}}\, K^T \qquad \text{with} \qquad K = \begin{bmatrix} r_r/2 & r_l/2 \\ r_r/e & -r_l/e \end{bmatrix}$$

A reasonable form for $Q_{\dot{q}}$:

$$Q_{\dot{q}} = \begin{bmatrix} \sigma_{\dot{q}}^2 & 0 \\ 0 & \sigma_{\dot{q}}^2 \end{bmatrix}$$

Now the number of tuning parameters for the input noise is 1.

REPORT QUESTIONS

① If overestimated → initial uncertainty high → P high → great attention on first data in order to correct trajectory.

If underestimated → when given wrong initial position it drift and diverge, because $y-\hat{y}$ will be very big and so won't be under mahalanobis threshold

②

OBSERVABILITY

We suppose to make a snapshot of the system w/ only one magnet → the one in (0,0).
So we need to modify ${}^{m}P_{mag}$ → matrix of the coordinates of the magnet in robot frame.

$$
{}^{m}P_{mag} = \begin{bmatrix} -x\cos\vartheta - y\sin\vartheta \\ -y\cos\vartheta + x\sin\vartheta \end{bmatrix}
$$

We need to compute the new C matrix: $\frac{\partial s_B}{\partial X}$

$$
\rightarrow \quad C = \begin{array}{c} \begin{bmatrix} -\cos\vartheta & -\sin\vartheta & +x\sin\vartheta - y\cos\vartheta \\ \sin\vartheta & -\cos\vartheta & y\sin\vartheta + x\cos\vartheta \end{bmatrix} \end{array} \begin{array}{c} dg_1 \\ dg_2 \end{array}
$$

$$
\begin{cases} \dot{x} = v\cos\vartheta \\ \dot{y} = v\sin\vartheta \\ \dot{\vartheta} = w \end{cases}
$$

$$
\begin{aligned}
dL_f g_1 &= d\big(+x\sin\vartheta \cdot \dot{\vartheta} - \dot{x}\cos\vartheta - \dot{y}\sin\vartheta - \cos\vartheta \cdot y\dot{\vartheta} \big) \\
&= d\big(xw\sin\vartheta - v\cos^2\vartheta - v\sin^2\vartheta - yw\cos\vartheta \big) \\
&= d\big(xw\sin\vartheta - yw\cos\vartheta - v \big)
\end{aligned}
$$

WRONG

We need to use two different magnets because otherwise we get $L_f g_1$ depending on w.

$$
{}^{m}P_{mag} = \begin{bmatrix} -x\cos\theta - y\sin\theta \\ -y\cos\theta + x\operatorname{sen}\theta \\ \cos\theta(1-x) - y\sin\theta \\ -y\cos\theta - \sin\theta(1-x) \end{bmatrix}
$$

$$
C = \begin{array}{c} \begin{bmatrix} -\cos\theta & -\sin\theta & +x\sin\theta - y\cos\theta \\ \operatorname{sen}\theta & -\cos\theta & y\operatorname{sen}\theta + x\cos\theta \\ -\cos\theta & -\sin\theta & -\sin\theta + x\sin\theta - y\cos\theta \\ +\sin\theta & -\cos\theta & y\sin\theta + x\cos\theta - \cos\theta \end{bmatrix} \end{array} \begin{array}{c} dg_1 \\ dg_2 \\ dg_3 \\ dg_4 \end{array}
$$

I try to compute the O matrix: if it's full rank the sys is obs and so i'm done.

$$
O = \begin{bmatrix} -\cos\theta & \operatorname{sen}\theta & -\cos\theta \\ -\sin\theta & -\cos\theta & -\sin\theta \\ x\sin\theta - y\cos\theta & y\operatorname{sen}\theta + x\cos\theta & -\sin\theta + x\sin\theta - y\cos\theta \end{bmatrix}
$$

dg_1

The $\operatorname{rank}(O) = 3$ so the system is observable

Doing the math:

$$\operatorname{rank}(O) = -\left(x\sin\theta - y\cos\theta\right)\cancel{\left[-\sin^2\theta - \cos^2\theta\right]} - \cancel{\left(y\sin\theta + x\cos\theta\right)}$$

$$\cancel{\left[+\sin\theta\cos\theta - \sin\theta\cos\theta\right]} + \left(-\sin\theta + x\sin\theta - y\cos\theta\right)$$

$$\cancel{\left[\cos^2\theta + \sin^2\theta\right]} = \cancel{y\cos\theta} - \cancel{x\sin\theta} - \sin\theta + \cancel{x\sin\theta}$$

$$\cancel{-y\cos\theta} = \underline{\underline{-\sin\theta}} \neq 0$$

```
% The variables and the evolution equation
syms x y theta v w
X = [x y theta].'
U = [v w].'
f = [ v*cos(theta) ; v*sin(theta) ; w ]
pause

% The general measurement equation for magnet
(x_mag,y_mag) is:
% g(X) = | cos(theta)(xmag-x)+sin(theta)(ymag-y) |
%        | cos(theta)(ymag-y)-sin(theta)(xmag-x) |

% Case when the two coordinates of each magnet in the
robot frame are
% measured. Should of course be observable. Let's practice
by checking it.
% Without loss of generality, the magnet coordinates are :
% M0=(0,0) and M1=(1,0)
% The four components of the measurement equation are
called
% g0x, g0y, g1x, g1y, with the obvious meaning.
```

```
g0x = -x*cos(theta) - y*sin(theta)
g0y = -y*cos(theta) + x*sin(theta)
g1x = cos(theta)*(1-x) - sin(theta)*y
g1y= -y*cos(theta) + (x-1)*sin(theta)
pause

% Gradient vectors of the four measurement components.

dg0x = jacobian(g0x,X).'
dg0y = jacobian(g0y,X).'
dg1x = jacobian(g1x,X).'
dg1y = jacobian(g1y,X).'
pause
```

WITH ONLY 1 MAG

$$C = \begin{array}{cc} \begin{bmatrix} -\cos\theta & -\sin\theta & +x\sin\theta - y\cos\theta \\ \text{sen}\,\theta & -\cos\theta & y\,\text{sen}\,\theta + x\cos\theta \end{bmatrix} & \begin{array}{c} \partial g_1 \\ \partial g_2 \end{array} \end{array}$$

$$\partial L_f g_1 = \partial\big(+x\,\text{sen}\,\theta \cdot \dot\theta - \dot x\cos\theta - \dot y \sin\theta - \cos\theta \cdot y\dot\theta \big)$$

$$= \partial\big(x\omega\,\text{sen}\,\theta - v\cos^2\theta - v\sin^2\theta - y\omega\cos\theta \big)$$

$$= \partial\big(\omega x\,\text{sen}\,\theta - \omega y\cos\theta - v \big)$$

$$\partial L_f g_1 = \begin{bmatrix} \omega\,\text{sen}\,\theta \\ -\omega\cos\theta \\ +\omega x\cos\theta + \omega y\,\text{sen}\,\theta \end{bmatrix}$$

ω multiplies everything so, when $\theta = 0$ or constant the system is not observable in this case

Now we try with $L_f g_2 = -y\cos\theta + x\,\mathrm{sen}\,\theta$

$$\partial L_f g_2 = \partial\left(-\cos\theta\,\dot{y} + \mathrm{sen}\,\theta\, y\dot{\theta} + \dot{x}\,\mathrm{sen}\,\theta + x\cos\theta\,\dot{\theta}\right)$$

$$= \partial\left(-v\,\cancel{\mathrm{sen}\,\theta\cos\theta} + \omega y\,\mathrm{sen}\,\theta + \cancel{v\cos\theta\,\mathrm{sen}\,\theta} + x\omega\cos\theta\right)$$

$$= \begin{bmatrix} \omega\cos\theta \\ \omega\,\mathrm{sen}\,\theta \\ \omega y\cos\theta - x\omega\,\mathrm{sen}\,\theta \end{bmatrix}$$

Same as before.