CENTRALE NANTES

Mobile Robots Lab
Localization using magnets
Presentation

Beware

- The robot is pretty much a toy.
- The sensor is extremely rudimentary.
- The problem is not a toy. It's actually a bit challenging and, should you have to solve it from scratch, it would take you some time.
- Much easier variants have been and are still used in the industry.

The robot

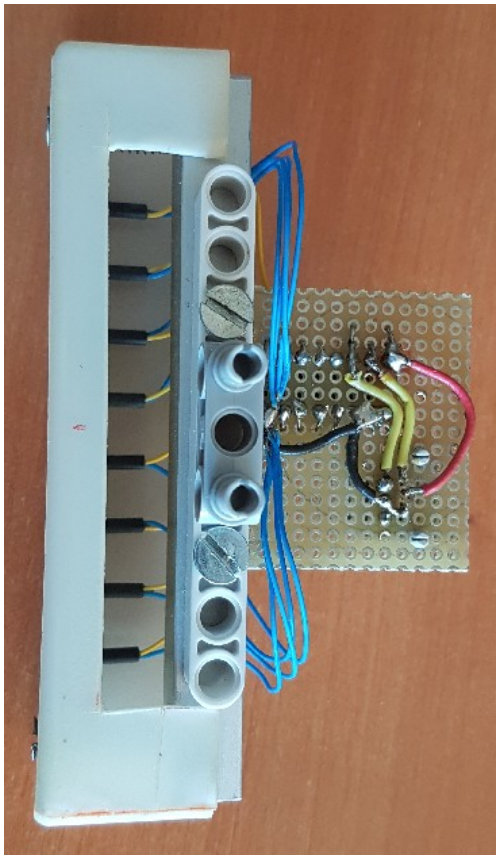
The robot and the sensor

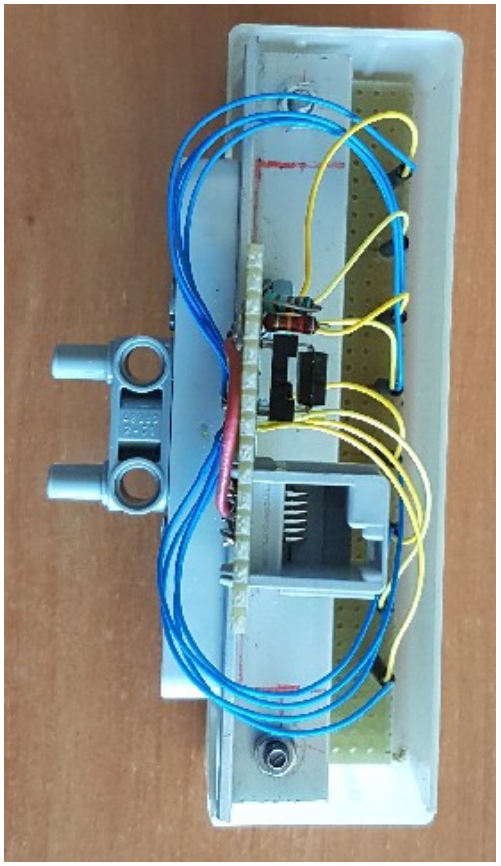
Robot characteristics

- 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.
  - We will work at 5 Hz.

The sensor and magnet setup

The magnet detector

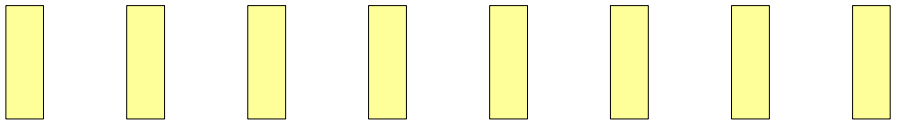




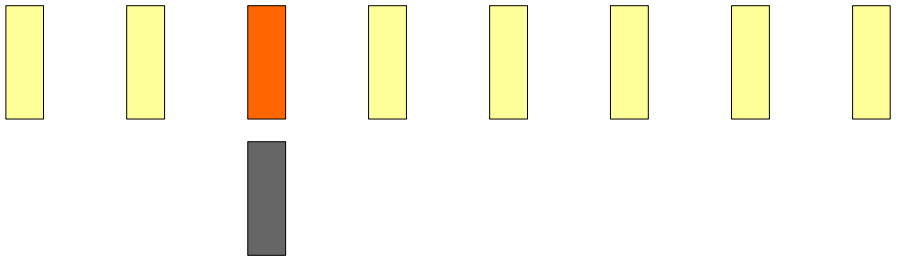
The reed switch

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

The magnet sensor

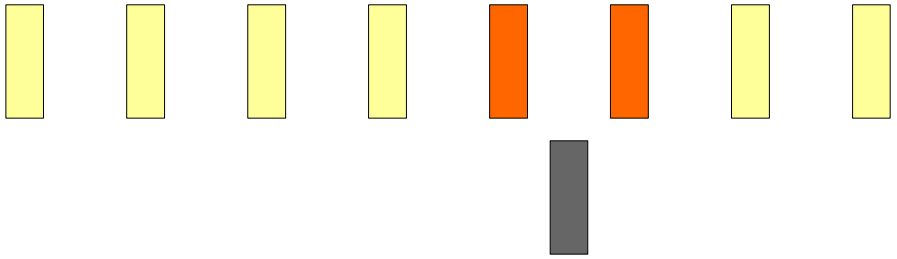


No magnet in the vicinity of the reed switches: all are open

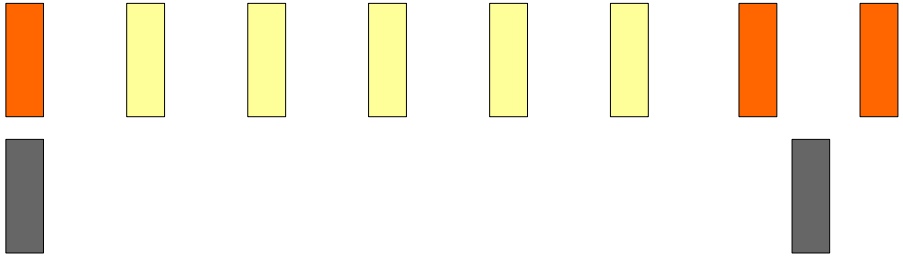


A magnet is right under reed switch 3, which is closed.

The magnet sensor

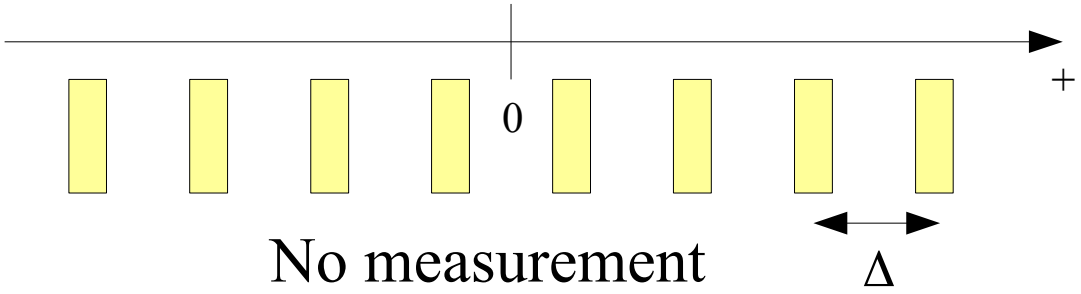


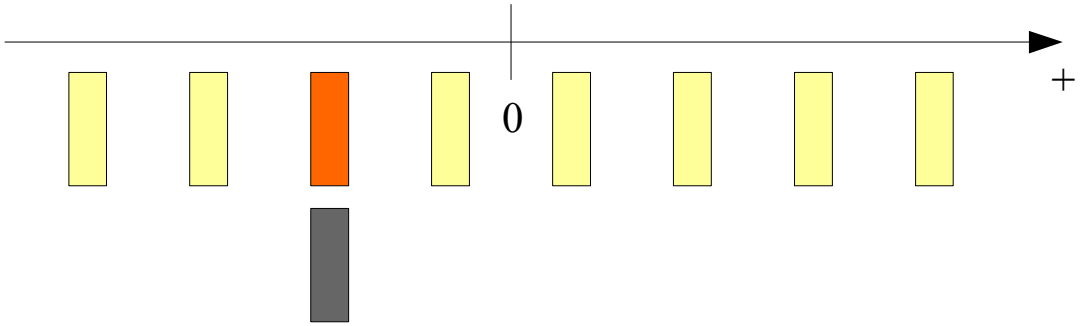
A magnet is right under switches 5 and 6, which are closed.



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

The magnet sensor measurements

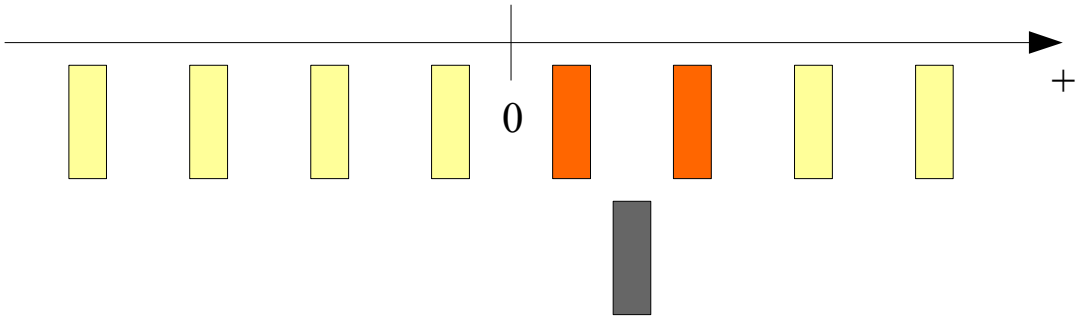




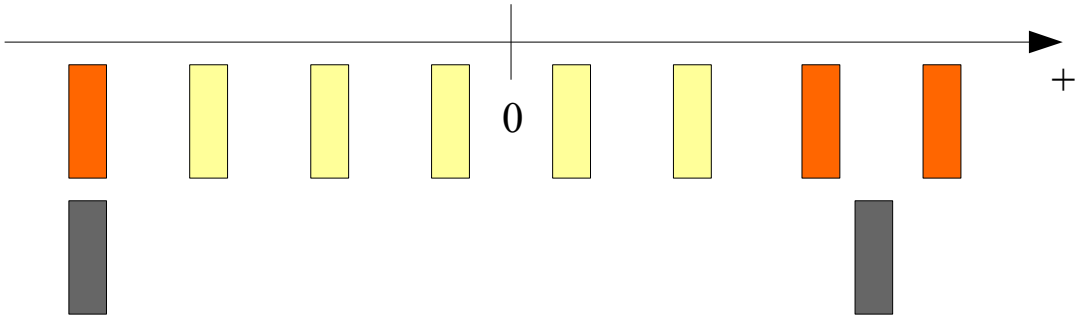
Measurement is -1.5 Δ

The resolution Δ is 10 mm

The magnet sensor measurements

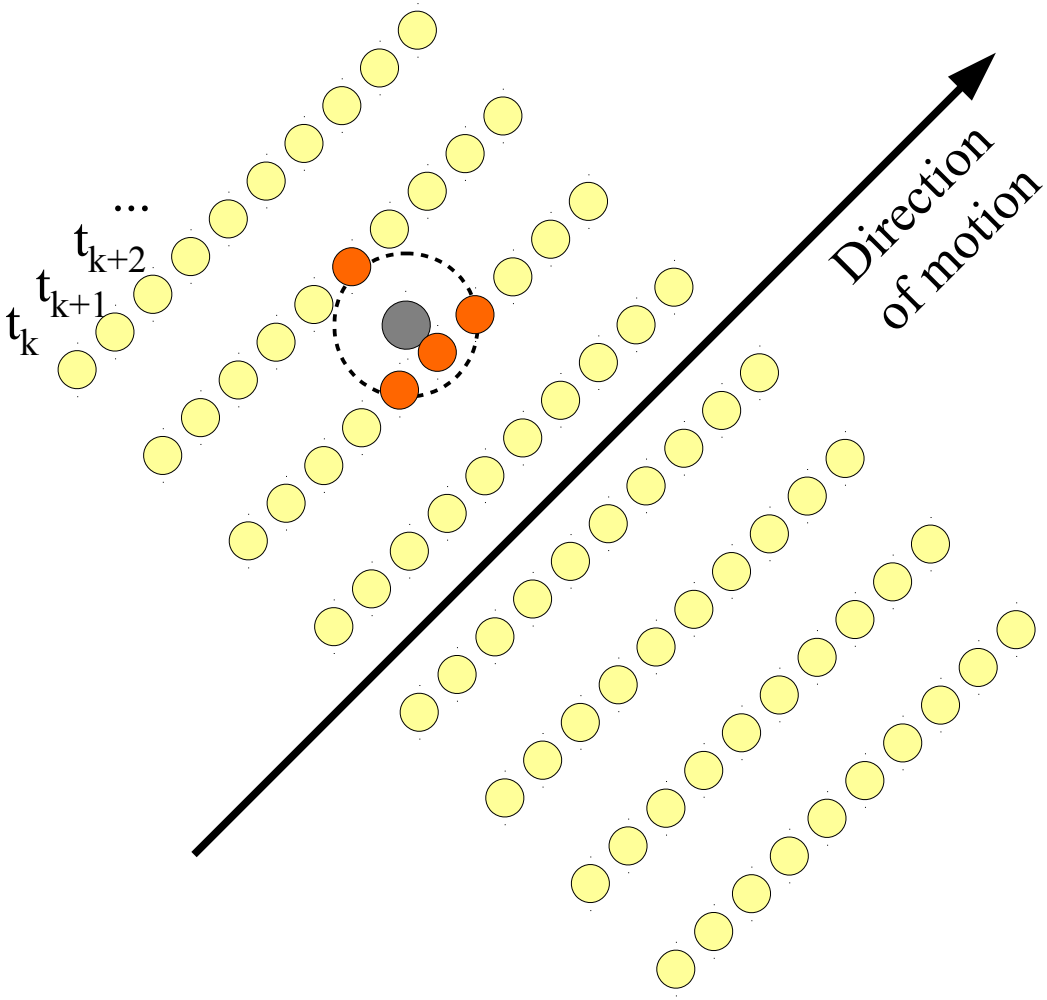


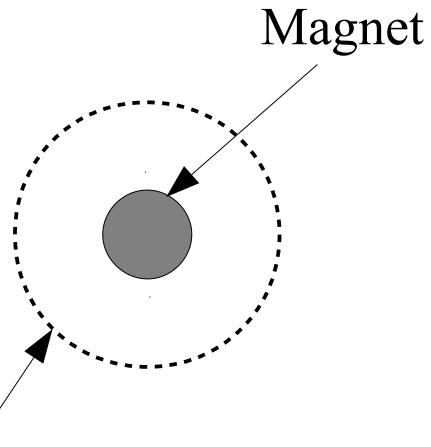
Measurement is +Δ



Two measurements: -3.5Δ and +3Δ

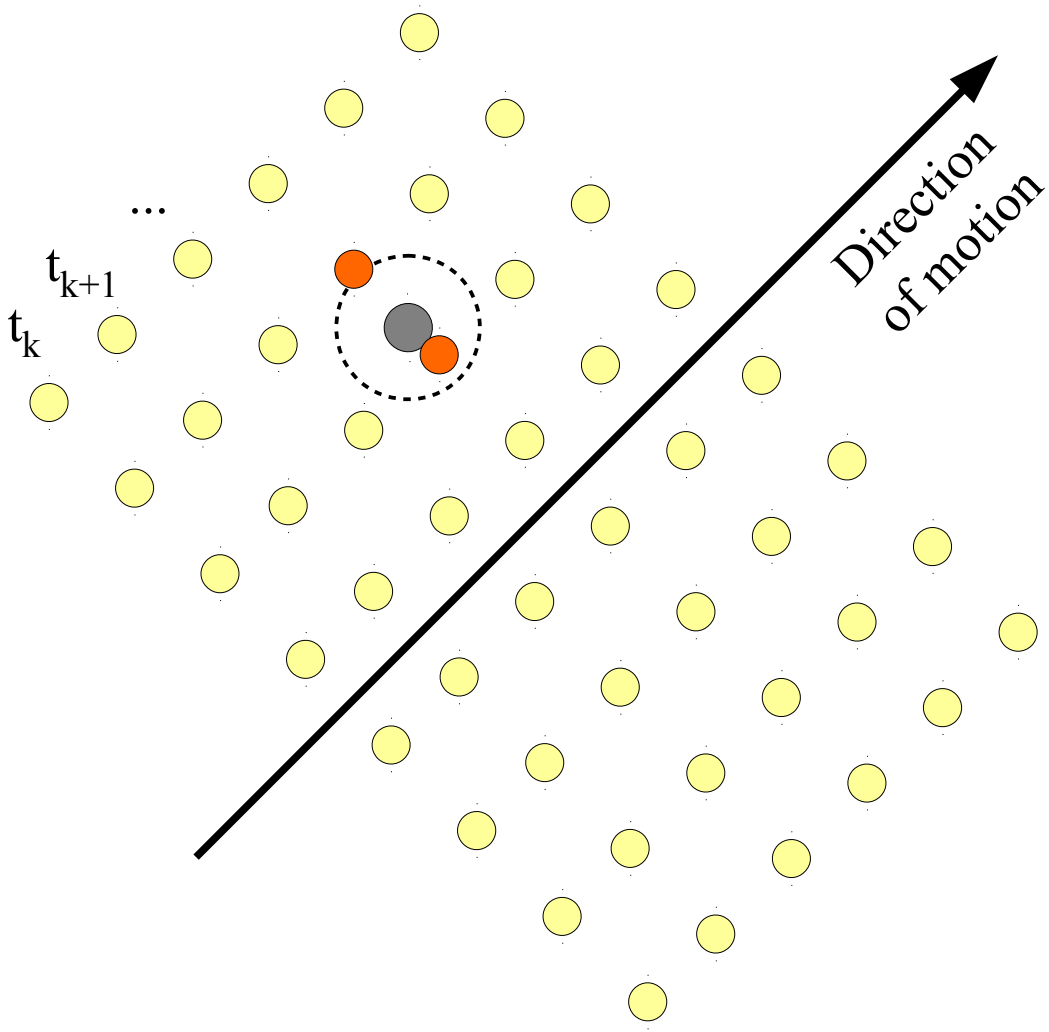
Sensor passing over a magnet





Area where the magnetic field is intense enough to switch the reed sensor on.

Sensor passing over a magnet
(lower sampling rate)

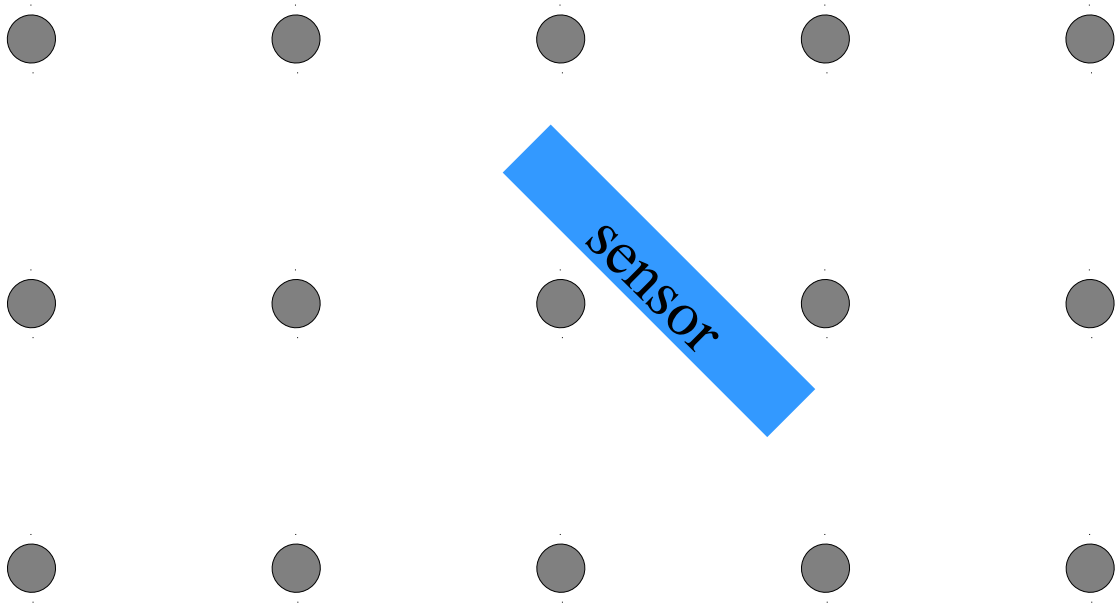


About the magnet sensor

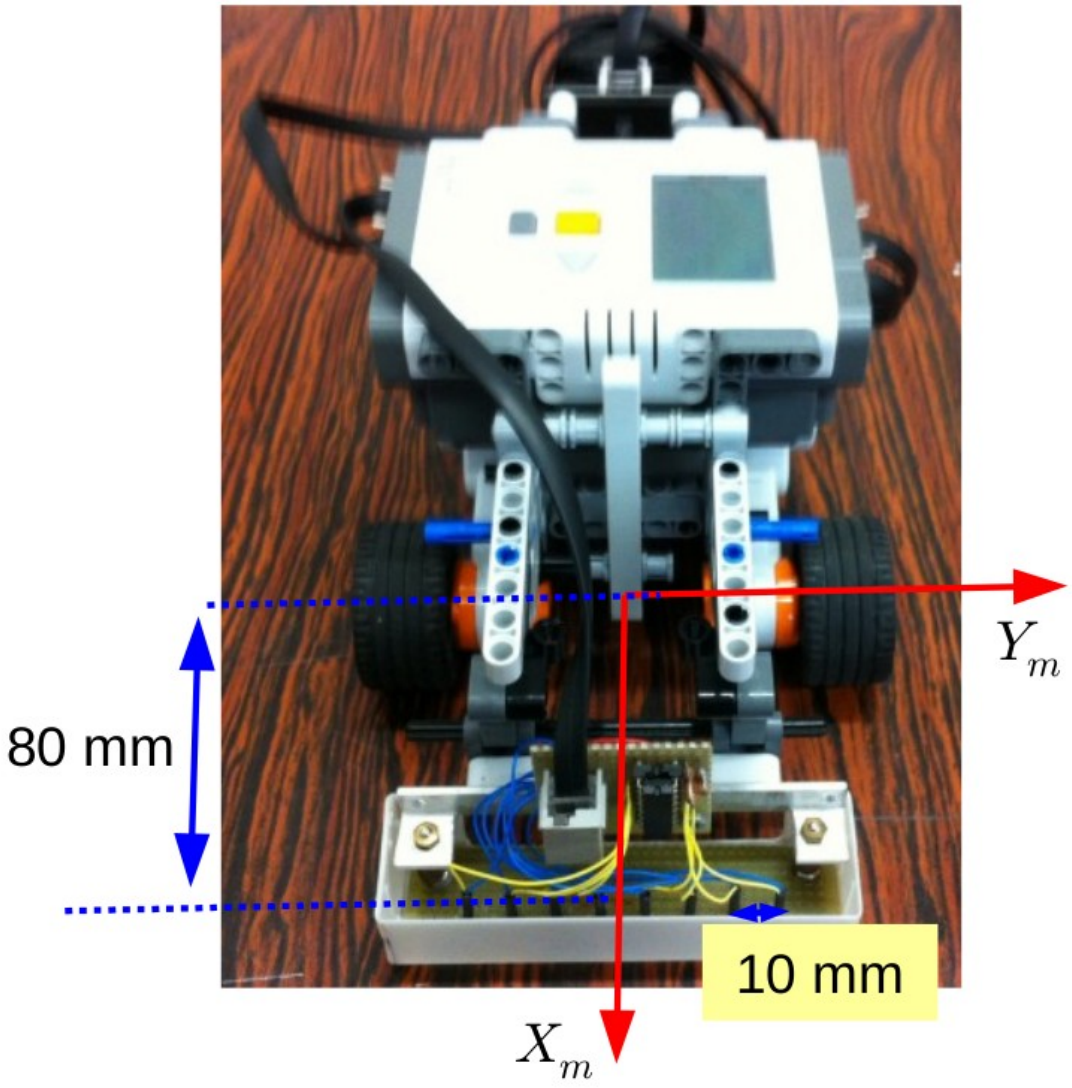
- The system design parameters are:
  - Magnet field intensity.
  - Inter-magnet distance
  - Reed switch sensitivity.
  - Reed switch number/spacing and sensor length.
- The sensor has been designed in such a way that:
  - When passing over a magnet, either one or two reed switches are activated.
  - 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.



Robot and sensor

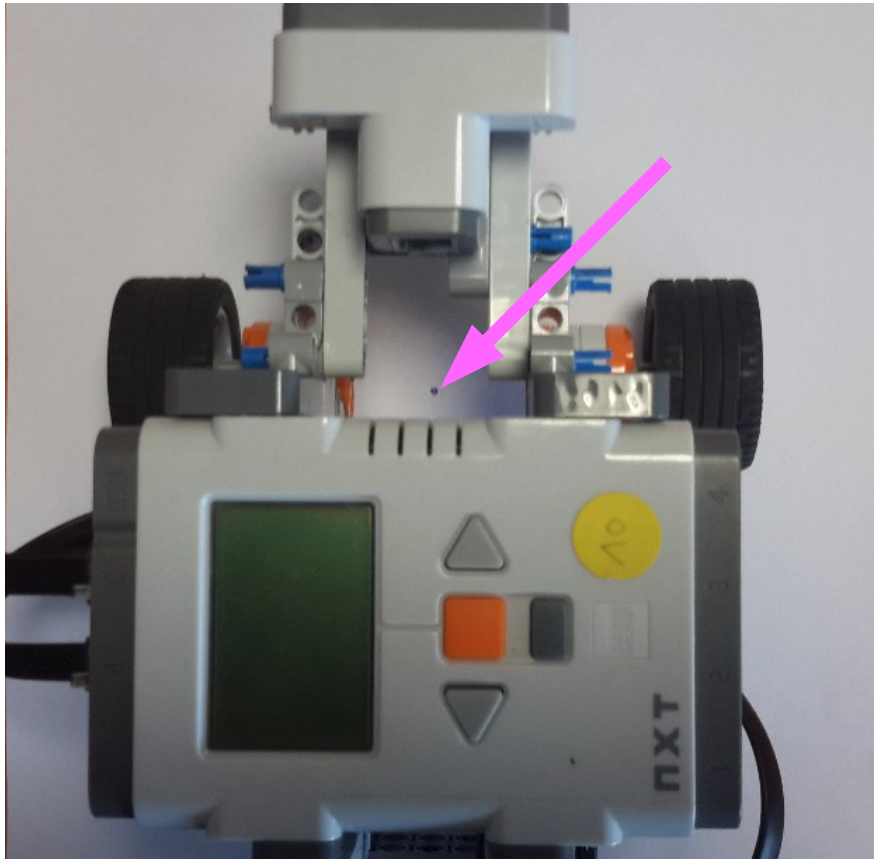


Information about the setup

- It is available in the file “RobotAndSensorDefinition.m”
  - Robot characteristics
  - Sensor characteristics
  - Magnet array
  - Frequency
- In the initial phases (determination of measurement noise variance), you can change the frequency to 20 Hz, but when you tune/run/test your Kalman filter, the frequency must be reset to 5 Hz.

Initial positioning of the robot

Initial positioning of the robot



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

Variant of Kalman filter equations

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.

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

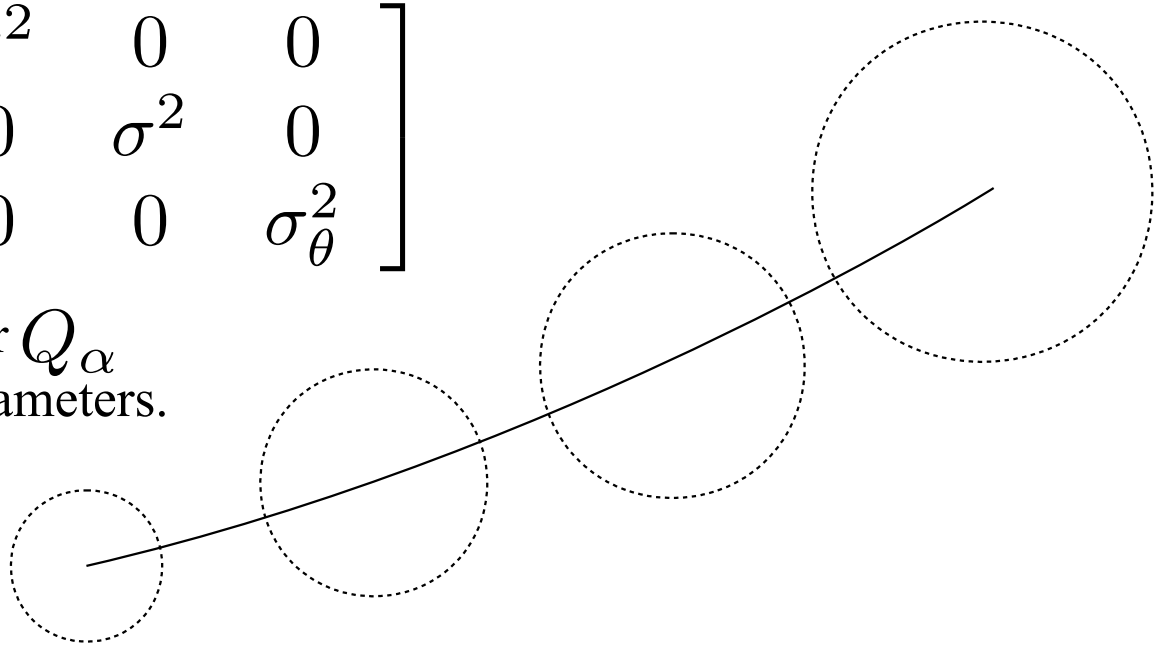
See paragraph 5.2 of the “book” form of the localization class material for the equations.

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).

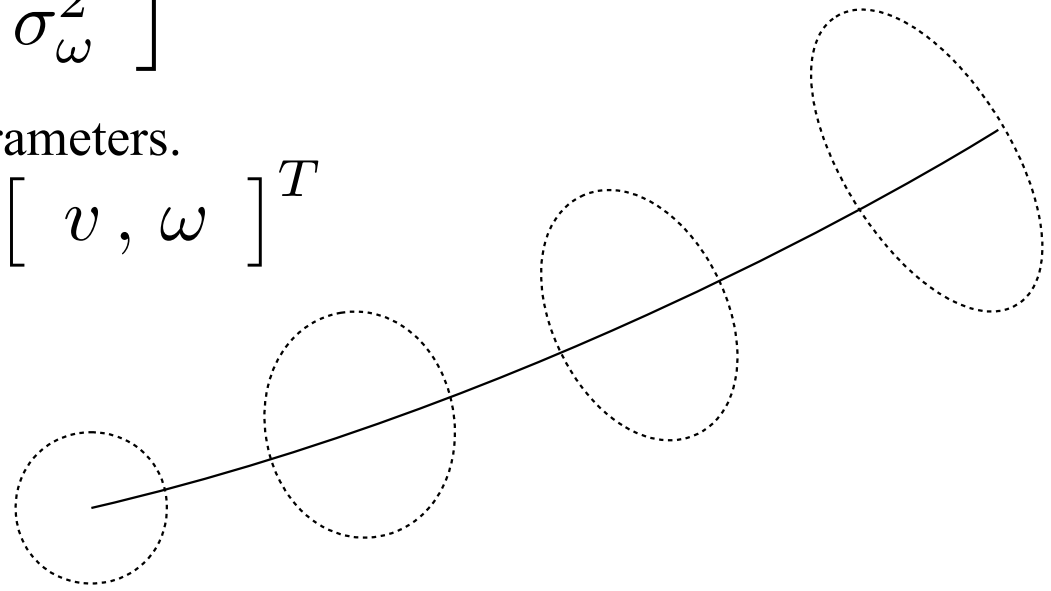
Evolution of uncertainty (form with noisy input)

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

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

Still two tuning parameters.

Remember: $u = \begin{bmatrix} v \, , \, \omega \end{bmatrix}^T$



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

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 $\dot{q}_l$.

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

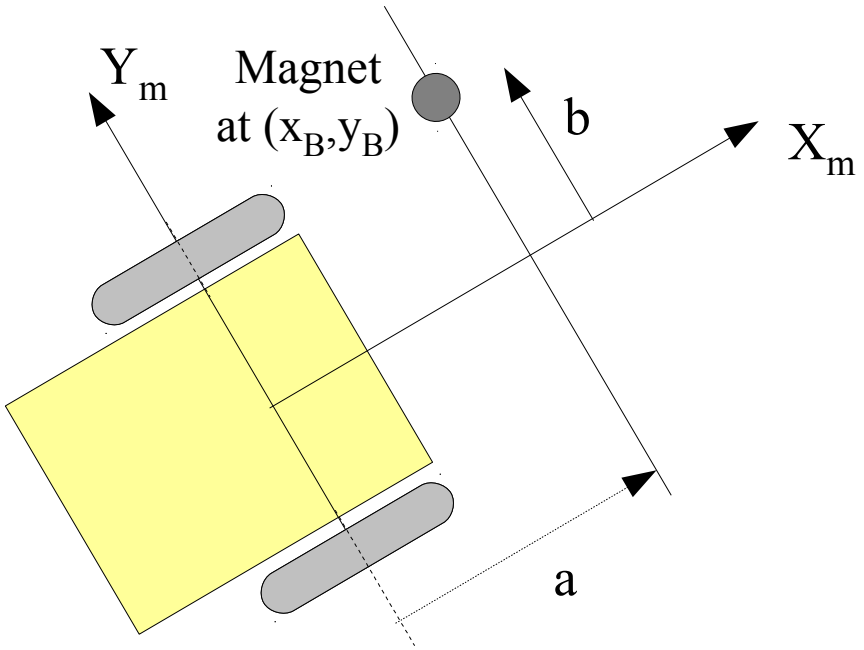
A reasonable form for $Q_{\dot{q}}$:

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

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

Measurement equation

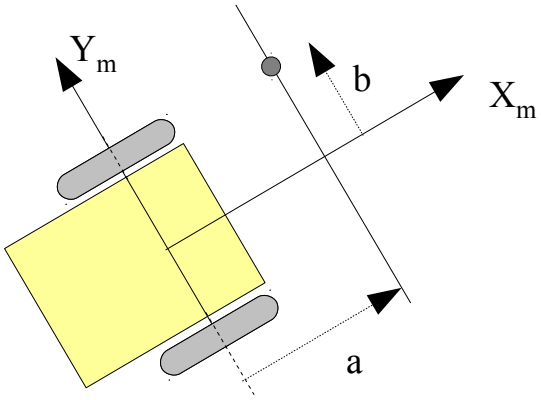
The robot detecting a magnet



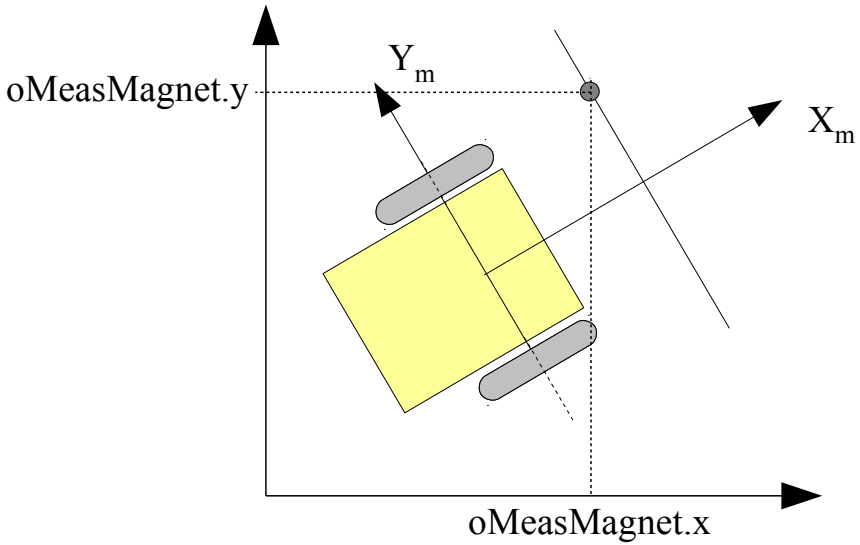
- Express in plain English what the sensor measures.
- Write the measurement equation

Which magnet to use?

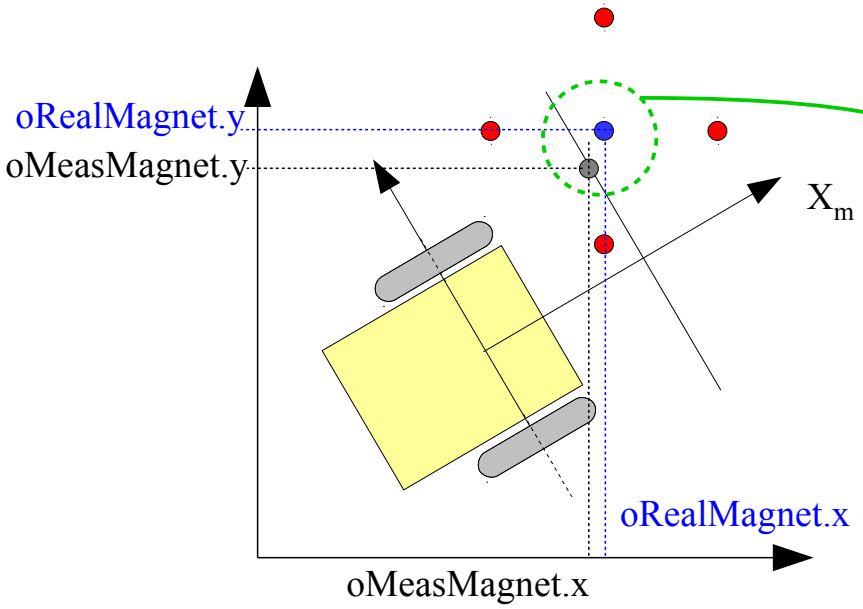
- Practical problem: the magnets are not signed beacons.
- The next slides explain how the problem is solved.
- It uses the same notations as in the program, so you can understand the code.
- Come back to these explanations if you need to.



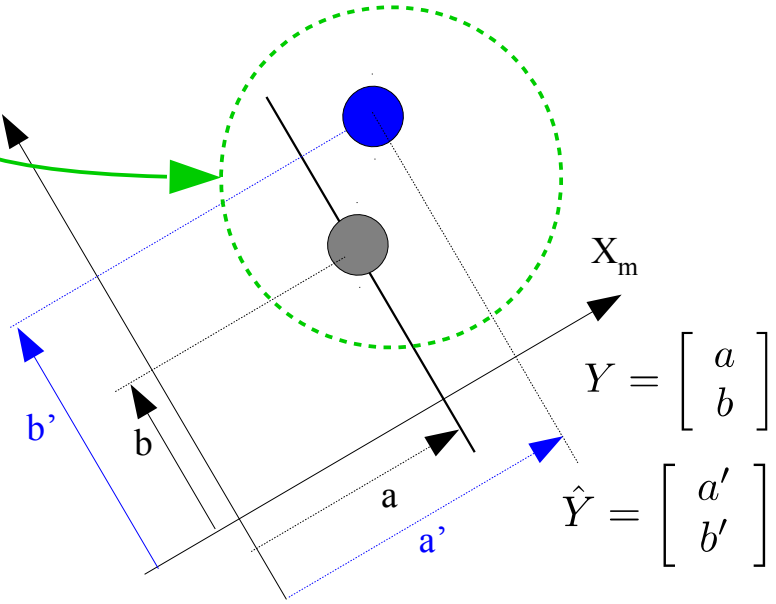
Step 1: measurement vector Y



Step 2: oMeasMagnet: coordinates of the measurement point in absolute frame



Step 3: No magnet coordinates exactly match, but one of them is closest: it's the candidate magnet. The algorithm assumes it's the correct one, and it will fail if not.



Step 4: The expected measurement is the coordinates of the real magnet in the robot frame

Setting the threshold

- Let Z be a random vector (zero mean) and Q its covariance matrix.

  $\sqrt{Z^t Q^{-1} Z}$ is a **dimensionless** distance.

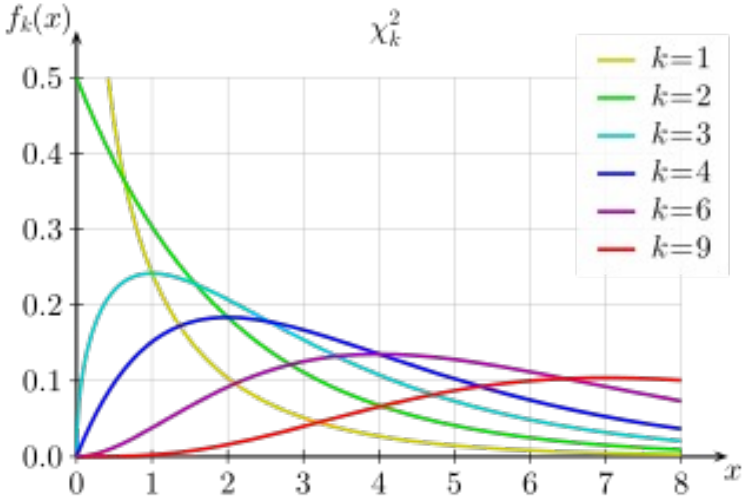
- If Z is a Gaussian zero mean random vector, then

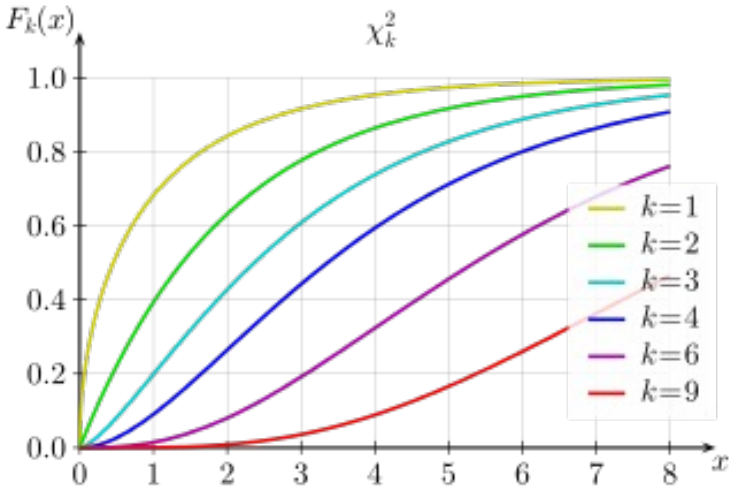
  $Z^t Q^{-1} Z$ follows a χ^2 distribution with dim(Z) dof.

- The shape of the distribution varies with the dimension of Z, that's why this parameter must be taken into account in determining the Mahalanobis distance threshold.

Setting the threshold

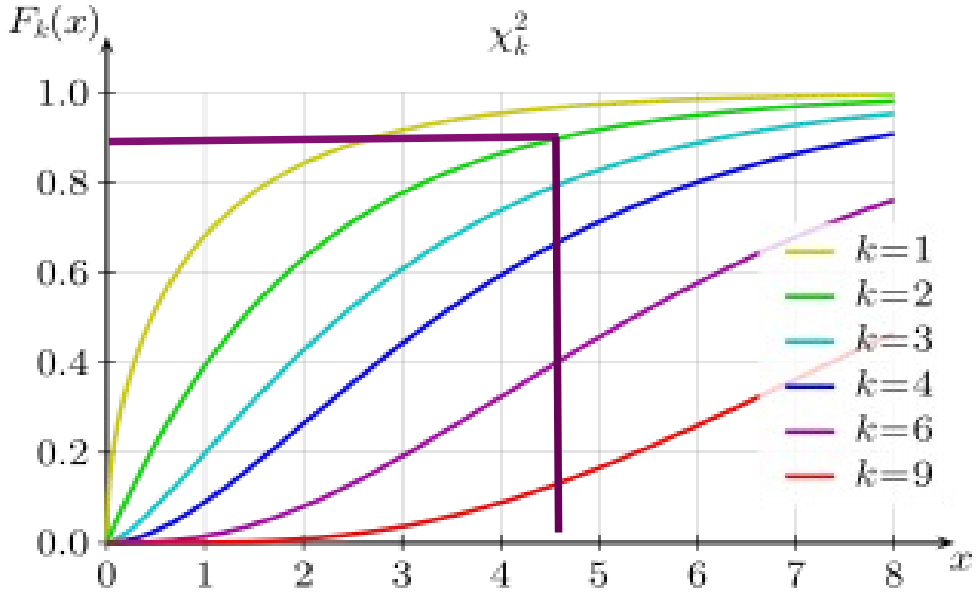
- Here are the shapes of the probability density and the cumulative distribution function (source Wikipedia), with k the number of degrees of freedom.





Setting the threshold

- Use of the cumulative distribution function.



if dim(Z)=2, then $proba(Z^t Q^{-1} Z \leq 4.6) \approx 0.9$

Setting the threshold

- The cumulative distribution function is available in tables (easily found online) and through the chi2inv function in Matlab.
- d^2 = chi2inv(p, number of dof) returns the squared distance such that :

$$proba(Z^t Q^{-1} Z \leq d^2) = p$$

- As you can see from the shapes of the distributions, dim(Z) has to be taken into account.

How do we use this theory?

- We apply it to the vector $Y - \hat{Y}$
- We use it to detect whether a measurement is acceptable or not.
- This will allow us to check whether we have associated the measurement to the correct beacon (here the correct magnet).
- In other words, we use it to perform the matching between measurements and unsigned beacons.
- Value of p. 0.95 is fine. Meaning: if the probablity of the event is less than 5 %, let's call it abnormal. Here: if the difference we have between the measurement and its predicted value has less than 5 % of probability, let's say we did not match the measurement to the correct magnet and let's not use the measurement.

A few remarks

- You have a lot of information to process. A slow start is normal.
- Review all information carefully before proceeding.
- Also, before starting, look at all the odometry-estimated paths using “showOdometry.m”.
- Remember that the odometry-estimated path is anything **but** a reference.
- The lab is not graded via a report, but via specific questions in the exam.
- Concentrate on understanding the how the Kalman filter works. What you need to get a good grade is not to memorize stuff, it is to understand.

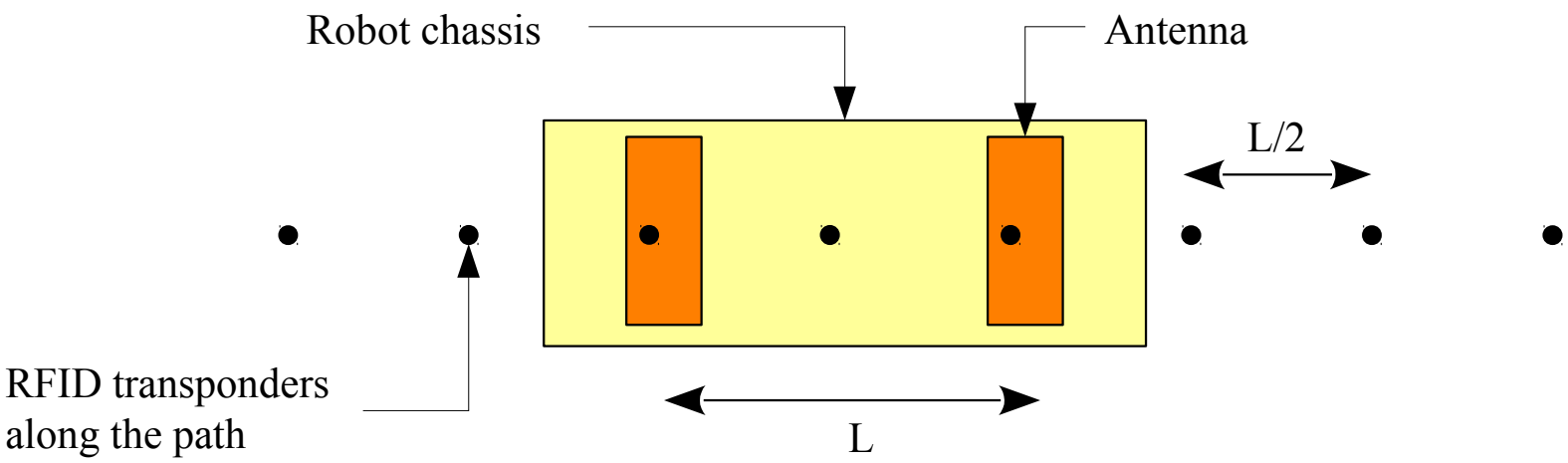
Project history

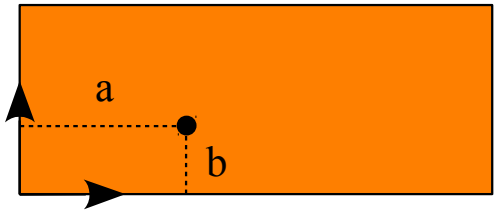
- Project initiator and principle of the system: G. Garcia
- Sensor prototype: J. Rousseau
- Characterization of the sensor and implementation of a sequential algorithm as described in the class:
  H. Chame & M. Gaudenzi de Faria.
- First Kalman filter implementation:
  F. Chichosz & M. Samuel.
- Version in this lab: G. Garcia

Industrial example

Container transport in ports

RFID tags and antenna solution





The antenna measures the position of the RFID transponder in the antenna frame

From the positions of the transponders relative to the two antennas and the known positions of the antennas relative to the vehicle, and the positions of the transponders in the absolute reference frame, the position of the vehicle is calculated.
Between two detections, odometry is used.

Vehicle and antennas

RFID tags and antenna solution

- The transponders are signed beacons. When the antenna passes over the transponder, it emits its position.
- The algorithm is very easy to implement.
- The locations of the transponders define the possible paths. It's not very flexible.
- Two different vehicles can use the same transponders if the distance bewteen their antennas agree.
- The system is costly: 10 to 20 k€ per antenna. 30 to 50 € per transponder, including setup, which includes measurement of the transponder position by surveying and programming of the coordinates in the transponder memory.
- With the methods developed in this lab, you can dramatically enhance the flexibility and considerably reduce the cost of the solution. With the results of the VASCO project, we got rid of the whole system using SLAM.

Project history

- Project initiator and principle of the system: G. Garcia
- Sensor prototype: J. Rousseau
- Characterization of the sensor and implementation of a sequential algorithm as described in the class:
  H. Chame & M. Gaudenzi de Faria.
- First Kalman filter implementation:
  F. Chichosz & M. Samuel.
- Version in this lab: G. Garcia

Using Zoom

- Use “ask for help” to call me to your room.
- If audio between me and you is lost, then most of the time the problem is solved by leaving the breakout room and entering it back.
- When in the breakout room, you are in a meeting with your group of people. You can talk to each other without bothering anyone. You can talk to me when I join your room.
- Use the screen sharing utility to show me what you are doing if necessary.
- I can request control of your mouse and keyboard. Keyboard is not perfect, due to different keyboard layouts. Mouse comes in handy to show you things on your own screen.