Mobile robotics

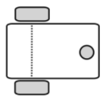
Notes from the a.y. 2021/2022 course held by Prof. Alessandro Farinelli

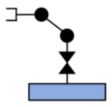
Author: Lorenzo Busellato

UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Contents

1 Wheeled locomotion


Mobile Robot


Robotic arm scheme (source [AMR])

Mobile robotics deals with **mobile robots**, i.e. robots that have the capability of moving their whole body around in the environment they are placed in.

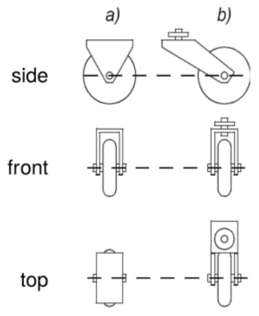
Mobile robots differ from robot manipulators in the fact that their position in the world reference frame is not fixed, and there are rolling and sliding constraints at the wheel-ground contact point to consider.

A mobile robot is a **non-holonomic system**, meaning that the differential equations that describe the robot kinematics are not integrable to obtain the final solution. This introduces the necessity of understanding the constraints imposed on the robot by the wheels, as well as the necessity of knowing the path the robot took during its motion.

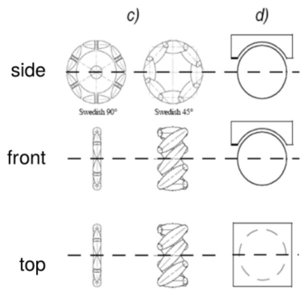
1.1 Wheel types and arrangements

There are four main types of wheels used in robotic applications:

- **Standard** wheels have at most 2 degrees of freedom, i.e. the rotations around the wheel axle and around the contact point.
- **Castor** wheels have at most 3 degrees of freedom, i.e. the rotations around the wheel axle, around the contact point and around the castor axle.
- **Swedish** or **mecanum** wheels have 3 degrees of freedom, i.e. around the wheel axle, around the contact point and around the rollers. The rollers are inclined with respect to the wheel axle by an angle $0 < \varphi \leq 90°$.
- **Spherical** wheels have 3 degrees of freedom, i.e. the rotations of the sphere around the three axes.

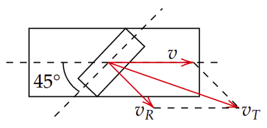


Standard and castor wheels (source [AMR])



Swedish and spherical wheels (source [AMR])

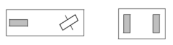
Swedish wheels are a particular type of wheel that can realize 3 degrees of freedom in the plane without needing a steering system. This is achieved by having a series of rollers on a standard wheel that are able to rotate independently of the wheel's rotation. If we consider a single roller:

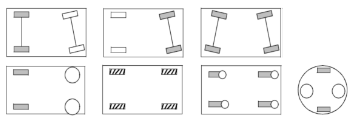


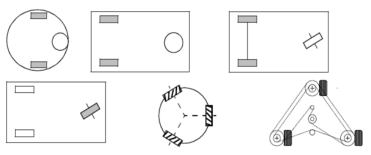
The roller has the effect of changing the direction of the wheel velocity v, and also to change its modulus with the additional velocity of the roller, v_R. By having multiple Swedish wheels on a robot and rotating them in different directions the redirected velocities of the wheels partly cancel out and result in a single velocity along any direction of the plane.

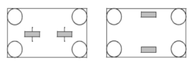
The minimum number of wheels to achieve stability is three. Depending on the application however fewer or more than three wheels may be used. In general stability is achieved if the **center of gravity** of the robot (i.e. the weighted average of the points that make up the robot, weighted with respect to their mass) lies within the poligon whose vertexes are the contact points with the ground of the wheels. In principle more wheels improve stability, but more wheels than three make the system hyperstatic, i.e. introduce more constraints than degrees of freedom. In this case the robot needs a flexible suspension system.

The wheels on a robot can then be arranged in a number of ways, depending on the application.


Two wheels (source [AMR])

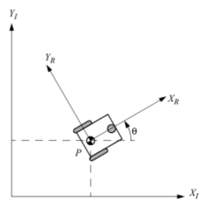

Four wheels (source [AMR])


Three wheels (source [AMR])


Six wheels (source [AMR])

1.2 Forward kinematics

To study forward kinematics we model the robot as a rigid body, in which we identify a reference point P, origin of a local reference frame R. The robot moves inside a global reference frame I, and the point P is represented in the global frame as:



Global and local reference frame (source [AMR])

$$\xi_R = \begin{bmatrix} x_R \\ y_R \\ \theta_R \end{bmatrix} = R(\theta)\xi_I = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ \theta_I \end{bmatrix}$$

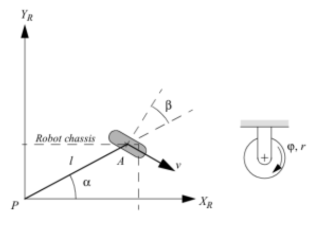
For mobile robots it's enough to consider only orthogonal rotation matrices to describe the mapping between frames. The problem then is to determine the speed of the robot in the inertial (global) frame given the spinning speeds of the wheels $\dot{\varphi}_i$, the heading θ and the geometric properties of the robot, i.e. the wheel radius r and the distance l of P from each wheel.

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2, ..., \dot{\varphi}_n) = R^{-1}(\theta)\dot{\xi}_R$$

To solve the problem we first assume that the plane of the wheel is vertical, that the contact point with the ground is unique and there is no sliding in the direction of the wheel normal. This means that we need to introduce two constraint, a rolling constraint and a sliding constraint.

For standard wheels the **rolling constraint** forces all motion components on the wheel plane to be due to the wheel rotation speed:

$$\begin{bmatrix} \sin(\alpha+\beta) & -\cos(\alpha+\beta) & -l\cos\beta \end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$



Fixed standard wheel and its parameter (source [AMR])

The **sliding constraint** forces all motion perpendicular to the plane to be null:

$$\begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta \end{bmatrix} R(\theta)\dot{\xi}_I = 0$$

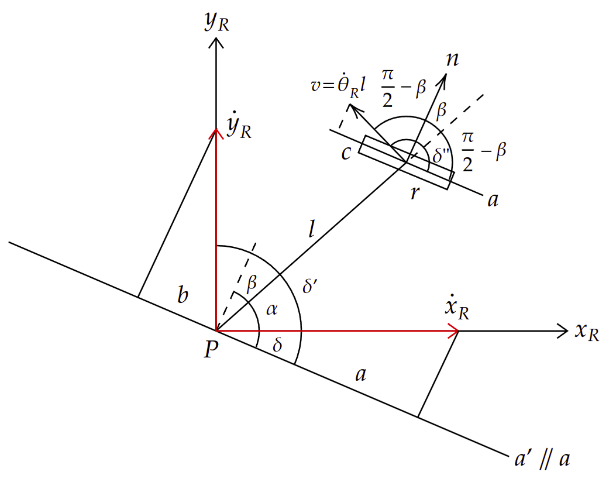
For swedish wheels the results are similar, but we need to account for the parameters of the rollers, namely their inclination γ with respect to the wheel plane, their radius r_{sw} and the speed at which they rotate, $\dot{\varphi}_{sw}$.
The rolling constraint for a swedish wheel is:

$$\begin{bmatrix}\sin(\alpha+\beta+\gamma) & -\cos(\alpha+\beta+\gamma) & -l\cos(\beta+\gamma)\end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi}\cos\gamma = 0$$

While the sliding constraint is:

$$\begin{bmatrix}\cos(\alpha+\beta+\gamma) & \sin(\alpha+\beta+\gamma) & l\sin(\beta+\gamma)\end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi}\sin\gamma - r_{sw}\dot{\varphi}_{sw} = 0$$

Example 1.1 (Derivation of the rolling constraint)
The constraints discussed above come from geometrical considerations. For example we consider a fixed standard wheel:



To define a rolling constraint we need to project the three components of motion, $\dot{x}_R$, $\dot{y}_R$ and $\dot{\theta}_R$, onto the wheel plane, identified by the axis a, and check that the result is due only because of the wheel rotation speed.
For $\dot{x}_R$ we have:

$$a = \dot{x}_R\cos\delta = \dot{x}_R\cos\left(\frac{\pi}{2}-\alpha-\beta\right) = \dot{x}_R\cos\left(\frac{\pi}{2}-(\alpha+\beta)\right) = \dot{x}_R\sin(\alpha+\beta)$$

For $\dot{y}_R$ we have:

$$b = \dot{y}_R\cos\delta' = \dot{y}_R\cos\left(\frac{\pi}{2}+\delta\right) = \dot{y}_R\cos\left(\pi-(\alpha+\beta)\right) = -\dot{y}_R\cos(\alpha+\beta)$$

For $\dot{\theta}_R$ we have:

$$c = \dot{\theta}_R\cos\delta" = \dot{\theta}_R\cos\left(\frac{\pi}{2}-\beta+\beta+\frac{\pi}{2}-\beta\right) = \dot{\theta}_R\cos\left(\pi-\beta\right) = \dot{-}\theta_R\cos(\beta)$$

Since $\begin{bmatrix}\dot{x}_R & \dot{y}_R & \dot{\theta}_R\end{bmatrix} = \dot{\xi}_R = R(\theta)\dot{\xi}_I$ *we can then write the rolling constraint:*

$$\begin{bmatrix}a & b & c\end{bmatrix} - r\dot{\varphi} = 0 \implies \begin{bmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & -l\cos\beta\end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

The sliding constraint is derived in a similar way, only projecting motion along the wheel normal.

1.3 Robot maneuverability

Mobile robots have multiple wheels of different types. We define the **number of wheels** as:

$$M = N_f + N_s$$

where N_f is the number of fixed wheels and N_s is the number of steerable wheels.
We can then aggregate the constraints imposed by the wheels. For the rolling constraint we have:

$$J_1(\beta_s)R(\theta)\dot{\xi}_I - J_2\dot{\varphi} = 0 \quad \text{with } \varphi(t) = \begin{bmatrix}\varphi_f(t)\\ \varphi_s(t)\end{bmatrix}, J_1(\beta_s) = \begin{bmatrix}J_{1f}\\ J_{1s}(\beta_s)\end{bmatrix} \in \mathbb{R}^{(N_f+N_s)\times 3}, J_2 = diag(r_1,\ldots,r_M)$$

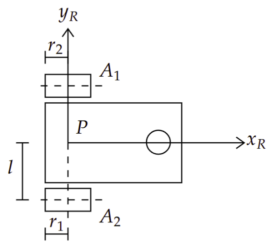
And for the sliding constraint:

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0 \qquad \text{with } C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{12}(\beta_s) \end{bmatrix} \in \mathbb{R}^{(N_f+N_s)\times 3}$$

Finally we can derive the forward differential kinematic model for any robot by putting together the above expressions:

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix} \implies \dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix}^{-1} \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix}$$

Example 1.2 (Differential drive)
Consider a differential drive robot:



The robot has $M = N_f + N_s = 2 + 0 = 2$ wheels. Since there are no steerable wheels ($N_s = 0$), the constraint matrices are reduced to their fixed parts:

$$J_1(\beta_s) = J_{1f} = \begin{bmatrix} \sin(\alpha_1+\beta_1) & -\cos(\alpha_1+\beta_1) & -l\cos\beta_1 \\ \sin(\alpha_2+\beta_2) & -\cos(\alpha_2+\beta_2) & -l\cos\beta_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{bmatrix}$$
$$C_1(\beta_s) = = C_{1f} = \begin{bmatrix} \cos(\alpha_1+\beta_1) & \sin(\alpha_+1\beta_1) & l\sin\beta_1 \\ \cos(\alpha_2+\beta_2) & \sin(\alpha_2+\beta_2) & l\sin\beta_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The other matrices we need are:

$$J_2 = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \qquad \dot{\varphi} = \begin{bmatrix} \dot{\varphi}_1 & \dot{\varphi}_2 \end{bmatrix}$$

The forward kinematics are derived as:

$$\begin{aligned} \dot{\xi}_I &= \left(\begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix} R(\theta=0) \right)^{-1} \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix}^{-1} \begin{bmatrix} J_2\dot{\varphi} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2l} & -\frac{1}{2l} & 0 \end{bmatrix} \begin{bmatrix} r_1\dot{\varphi}_1 \\ r_2\dot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} \frac{r_1\dot{\varphi}_1}{2} + \frac{r_2\dot{\varphi}_2}{2} \\ 0 \\ \frac{r_1\dot{\varphi}_1}{2l} - \frac{r_2\dot{\varphi}_2}{2l} \end{bmatrix} \end{aligned}$$

Maneuverability is a combination of mobility due to sliding constraints and additional freedom due to steering. It's defined as:

$$\delta_M = \delta_m + \delta_s$$

where δ_m is called **degree of mobility** and represents the number of degrees of freedom that can be directly influenced by the change in wheel speed, and δ_s is called **degree of steerability**, which represents the number of controllable steering parameters.

To define the degree of mobility we notice that the sliding constraint implies that $R(\theta)\dot{\xi}_I$ must belong to the null space of $C_1(\beta_s)$:

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0 \implies R(\theta)\dot{\xi}_I \in ker(C_1(\beta_s))$$

In other words the greater the rank of $C_1(\beta_s)$ (i.e. the smaller the dimension of its kernel), the less mobility a robot has:

$$\delta_m = 3 - rank(C_1(\beta_s)) \quad 0 < rank(C_1(\beta_s)) \le 3$$

The degree of steerability is defined as:

$$\delta_s = rank(C_{1s}(\beta_s)) \qquad 0 \leq \delta_s < 2$$

The addition of steerable wheels results in more sliding constraints being applied to the system, and so a decrease in maneuverability. The steering however lets us recover some degrees of maneuverability with the possibility of changing the wheel's orientation.

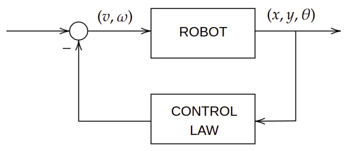
Overall the degree of maneuverability $\delta_M = \delta_m + \delta_s$ does not uniquely identify a robot configuration. For example $\delta_M = 3$ both for a tricycle and a differential drive.

1.4 Motion control

Now we need a kinematic controller that makes the robot follow a trajectory specified by position and velocity described over time. In general we won't consider robot dynamics. This is a difficult task since mobile robots are non-holonomic MIMO (Multi-Input, Multi-Output) systems.

The first approach is **open loop control**, in which we divide the trajectory into precomputed line segments. The drawback of this approach is that the precomputation step is computationally heavy, and the resulting method is not dynamic with respect to path changes. The resulting trajectory is also not smooth. Furthermore the open loop implies error accumulation over time.

A better approach is **feedback control**, in which we look for a matrix K, with $K_{ij} = k(t, e)$, such that:



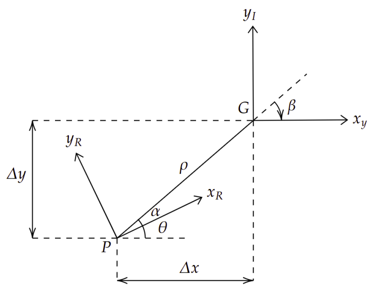
$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = Ke \implies \lim_{t\to\infty} e(t) = 0$$

Typically we deal first with (v, ω) because it is easier. Then we can transform to the inertial frame:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} C\theta & 0 \\ S\theta & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

To simplify the problem even further we transform the coordinates into polar coordinates:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$
$$\alpha = -\theta + atan2(\Delta y, \Delta x)$$
$$\beta = -\theta - \alpha$$



Notice that β is defined positive in the clockwise direction, while α and θ are positive in the counterclockwise direction. A smart way of placing the inertial frame is to place its origin coincident with the goal point of the trajectory, so that the robot position in the inertial frame is equal to the error of the control law.

The system becomes:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -C\alpha & 0 \\ \frac{S\alpha}{\rho} & -1 \\ -\frac{S\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad \text{with } \alpha \in I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

Which is the case in which the robot is facing towards the goal,and v is positive, and:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} C\alpha & 0 \\ -\frac{S\alpha}{\rho} & 1 \\ \frac{S\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad \text{with } \alpha \in I_2 = \left(-\pi, -\frac{\pi}{2}\right] \cup \left(\frac{\pi}{2}, \pi\right]$$

Which is the case in which the robot is facing away from the goal, and v is negative.

We now introduce the following control law:

$$v = k_\rho \rho \quad \omega = k_\alpha \alpha + k_\beta \beta$$

So:

$$\begin{bmatrix} \dot\rho \\ \dot\alpha \\ \dot\beta \end{bmatrix} = \begin{bmatrix} -k_\rho \rho C\alpha \\ k_\rho S\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho S\alpha \end{bmatrix} \quad \text{with } \alpha \in I_1$$

It is known that such a control law is locally exponentially stable if:

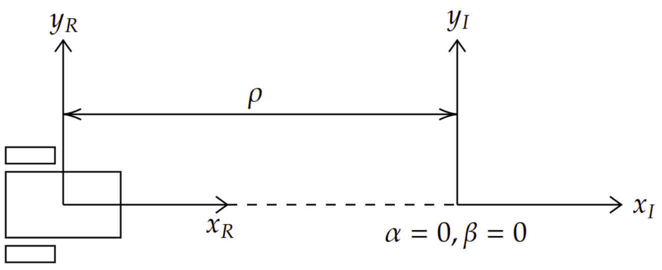
$$k_\rho > 0 \land k_\beta < 0 \land k_\alpha - k_\rho > 0$$

Furthermore we can impose a strong stability condition, i.e. one that imposes that the robot doesn't change direction during its approach to the goal:

$$k_\rho > 0 \land k_\beta < 0 \land k_\alpha + \frac{5}{3}k_\beta - \frac{2}{\pi}k_\rho > 0$$

The above implies that $\alpha \in I_1(I_2) \forall t$ if $\alpha(t=0) \in I_1(I_2)$.

Example 1.3
Consider the following situation:

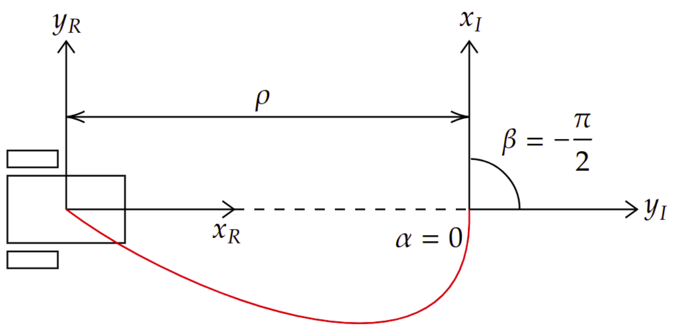


Since $\alpha = 0, \beta = 0$, if we apply the control law we get:

$$\begin{bmatrix} \dot\rho \\ \dot\alpha \\ \dot\beta \end{bmatrix} = \begin{bmatrix} -k_\rho \rho C\alpha \\ k_\rho S\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho S\alpha \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \\ 0 \\ 0 \end{bmatrix}$$

So, as we could have guessed, if $k_\rho > 0$ the system converges to the goal position while only moving along x_R.

Example 1.4
Consider the same situation as before, only with x_I and y_I swapped:



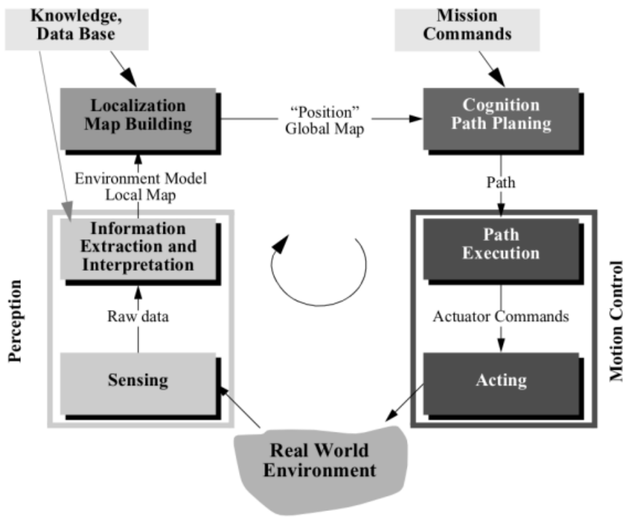
Now $\beta = \frac{\pi}{2}$ (since β is positive in the clockwise direction), if we apply the control law we get:

$$\begin{bmatrix} \dot\rho \\ \dot\alpha \\ \dot\beta \end{bmatrix} = \begin{bmatrix} -k_\rho \rho C\alpha \\ k_\rho S\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho S\alpha \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \\ k_\beta \frac{\pi}{2} \\ 0 \end{bmatrix}$$

In this case the robot will change orientation during its motion, resulting in a trajectory similar to the one in red.

2 Perception

Perception, i.e. the usage of sensor to extract data from the environment, is a fundamental part of the artificial intelligence pipeline.

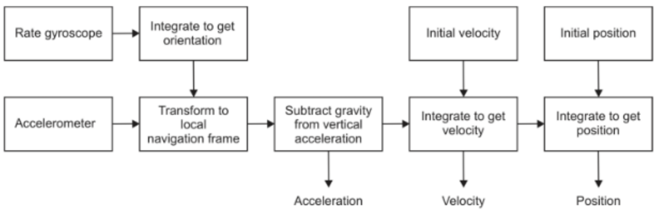


General control scheme for mobile robot systems (Source [AMR])

Perception is particularly challenging for robots because data is inherently noisy, and results from partial observations of the environment. Furthermore the transformation from data to context is not a trivial one.

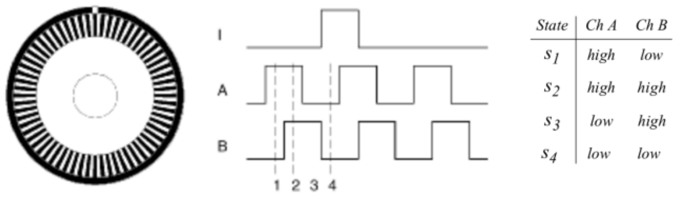
The main categories of sensors commonly employed in mobile robotics are:

- **Tactile sensors**, i.e. contact switches, bumpers, optical barriers etc.
- **Heading sensors**, which allow the robot to determine its orientation (**yaw**) and inclination (**roll** and **pitch**) with respect to a given reference frame. The main sensor in this category are:
  - **Gyroscopes**, which in robotics are devices based on the Sagnac effect, i.e. the phase shift proportional to angular velocity of laser beams traveling through optical fiber in opposite directions.
  - **Accelerometers**, which are spring-mass-damper systems implemented as MEMS (Micro Electro-Mechanical Systems), that measure all external forces applied to them (including gravity).
  - **IMUs (Inertial Measurement Units**, which integrate accelerometers and gyroscopes to estimate the relative position (x,y,z) and the orientation (roll,pitch,yaw) of a robot, as well as its velocity and acceleration with respect to an inertial frame.

    
    Rate gyroscope
Integrate to get orientation
Initial velocity
Initial position
Accelerometer
Transform to local navigation frame
Subtract gravity from vertical acceleration
Integrate to get velocity
Integrate to get position
Acceleration
Velocity
Position

    IMUs require the knowledge of initial velocity as well as gravity compensation. The main issue with IMUs is drift (two integrations in an open loop configuration), so recalibration is often needed.
  - **Beacons** and **GPS**, which utilize artificial landmarks to which signals are sent. By estimating the time-of-flight, i.e. the time spent by the signal to travel to the landmark and back, the robot's position with respect to an inertial frame can be determined. GPS uses satellites as landmarks, but since indoor GPS is often not precise, RFID, WIFI or Bluetooth landmarks are used.

- **Wheel encoders** are devices that convert position into an analog or digital signal by means of counting how many times a beam of light passes through slits in a wheel attached to the rotation axle.



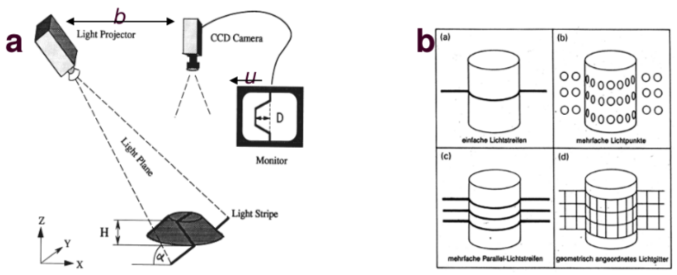
State	Ch A	Ch B
s_1	high	low
s_2	high	high
s_3	low	high
s_4	low	low

Quadrature optical encoder (source [AMR])

  Typically encoders have two concentric series of slits that are out of phase, so that two signals in quadrature are produced. This lets the robot know not only the angular position but the rotation direction as well.

- **Range sensors** utilize the same principle of GPS, but by evaluating the time-of-flight on the reflection of a signal from the surrounding environment. The signal can be sound (**sonars**) or light (**lidars**). The main difference between the two is the speed at which the signal travels, which is obviously much higher for a lidar. This means that to employ a lidar a robot must be able to accurately measure time, and this is the main source of inaccuracies in the sensor's readings. Lidars also cannot measure distance against non-reflective materials such as glass.

  Another type of range sensors are **structured light** sensors, that project beams of light in a certain arrangement onto the environment, so that a camera can detect changes in the arrangement due to the different depths of the objects in the scene.



Principles for light structure sensors and possible patterns (source AMR)