Inertial Navigation

As we have seen from the localization algorithms, the robot uses its built-in sensors to obtain information that allows the localization process. While we normally rely on sensors that directly observe some kind of landmark or distinctive feature in the environment the robot is working in, there are other sources of information that are available and can be helpful. In particular, it is possible to use ob-board sensors to provide additional information useful in keeping track of the robot's position. This is useful in supporting the localization process, or to provide localization information in environments where there are no landmarks or features supporting our Markov localization methods.

Inertial navigation is a process that relies on using information from accelerometes and gyroscopes to determine the direction and speed of a robot's motion. Given an initial position, inertial navigation can be used to keep track of where the robot is over time.

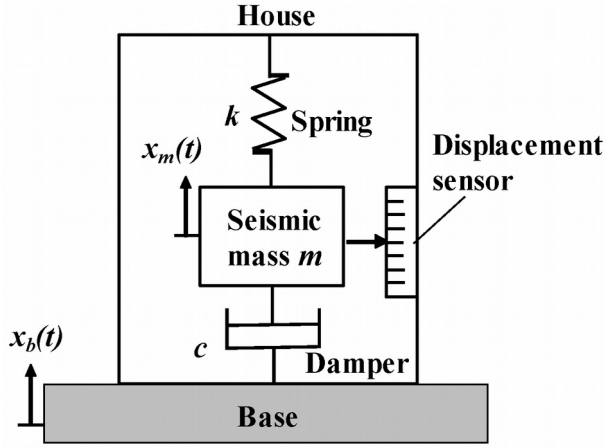
Unlike Markov localization, inertial navigation does not rely on a map, and provides position, velocity, and orientation for the robot with respect to its initial state.

Measuring Acceleration and Rotation

Inertial navigation relies on being able to measure both linear and rotational (angular) acceleration. Whenever the robot moves, it must be subjected to either or both of these, and by measuring them, we can determine the forces involved, as well as the velocity and direction of motion, or the amount of rotation the robot is undergoing as a result of its actions or of environmental forces acting on it.

Measuring linear acceleration is relatively straightforward. The idea is to use Newton's first law of motion – objects have inertia. A linear accelerometer consists of a ***proof mass*** (this is a lump of material that is allowed to move along some pre-defined axis), and a measurement device that determines how much the proof mass has moved.

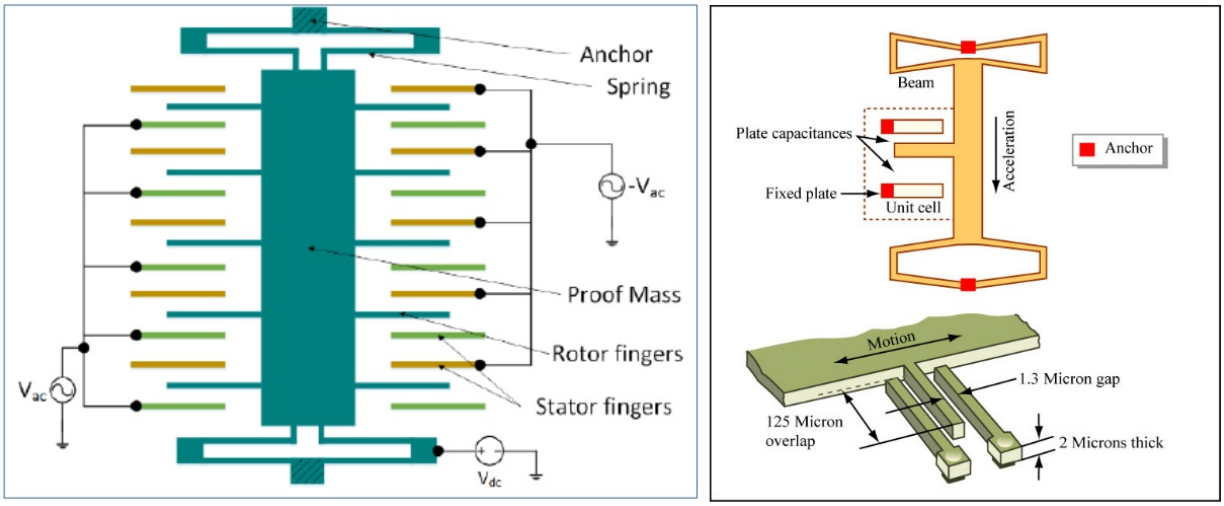
Here's an illustration of a simple 1D linear accelerometer consisting of a mass-and-spring system:



(Image: Scholarly Community Encyclopedia, CC BY-NC-ND)

In the accelerometer shown above, a proof mass is connected to a housing by a spring and damper. Acceleration in the vertical direction will cause the housing to move, while the proof mass' inertia will cause it to stay in place. This causes the relative position of the mass with respect to the casing to change, and this change is measured and used to determine how much acceleration was applied to the device.

Current electronic accelerometers use the same principle, but they are built entirely from solid state components with moving parts. Modern accelerometers belong to a class of devices known as ***MEMS*** **–** micro electromechanical systems. A schematic of how a MEMS accelerometer works is shown below



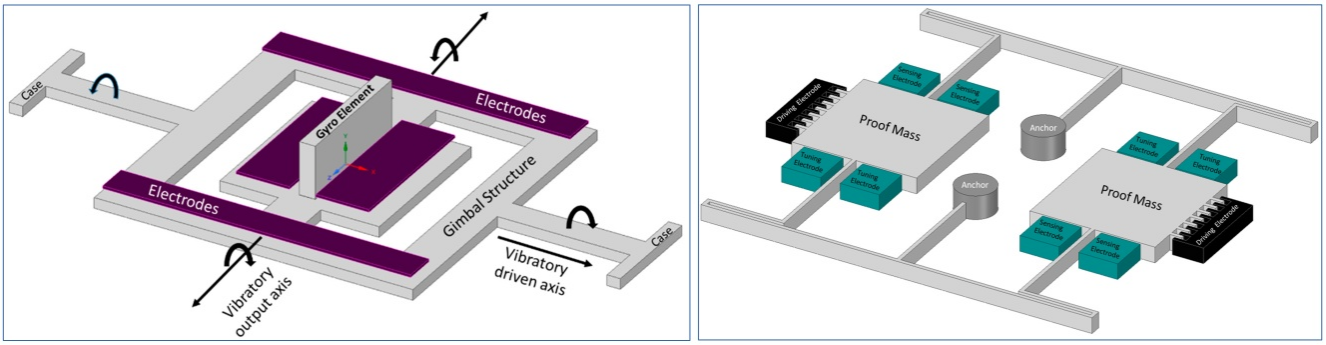
(Images: Left – Wikmedia foundation, by Sinha et al., CC-SA 4.0. Right – MIT Open Courseware, MIT Open Courseware license)

An accelerometer can only provide information along the axis of motion. We need three accelerometers oriented orthogonal to each other to provide information along the three possible directions of motion.

Along with acceleration, we need information regarding whether the robot is turning. This requires another type of sensor.

Gyroscopes

A gyroscope is a device that relies on the following principle: A mass that is rotating on a plane, or a mass that is oscillating on a plane will resist (because of inertia) a change in the orientation away from this plane. Therefore gyroscopes contain either a rotating proof mass, or an oscillating component. Schematics for two different gyroscope designs are shown below



(Images: Scholarly Community Encyclopedia, CC BY-NC-ND)

Whatever the structure of the gyroscope, what we obtain from it is a measure of rotation. The most common type of gyro is called a ***rate gyro*** and provides the amount of change in angle per unit of time. Just like with accelerometers, three gyroscopes are needed to provide information about rotation with regard to the three spatial axes.

A more complete explanation of how MEMS accelerometers and gyroscopes work can be found here: https://www.youtube.com/watch?v=9X4frIQo7x0

It must be noted that just like wny other sensor, accelerometers and gyroscopes are affected by noise, and therefore the information they provide is inaccurate to a smaller or larger degree depending on many factors that are impossible to fully account for. This will become important in a moment.

Calculating position and orientation

From the acceleration information provided by the sensors, the process of determining the robot's orientation, velocity, and position over time follows from the following principles:

- Robot orientation is obtained by integrating the angle's rate of change over time
- Robot velocity is obtained by integrating linear acceleration over time
- Robot position is obtained by integrating velocity over time

The first step is to determine the robot's orientation and for this we first need to define two coordinate frames – ***the global coordinate frame*** describes the position and orientation of the robot as well as any other objects, landmarks and obstacles in the robot's environment with respect to a pre-defined point chosen as the origin. The robot moves and rotates within this global coordinate frame, but its own orientation doesn't have to match that of the global X, Y, and Z directions – therefore, the robot has its own ***body coordinate frame*** which consists of a central location (typically at the center of the robot), and the diection of the coordinate axes that are aligned with the robot's body.

We need both of these because the sensors in the robot can only report rate of change with regard to the ***body coordinate frame***, but we need to know robot position and orientation in the ***global coordinate frame***.

For orientation, we can use a 3x3 rotation matrix to provide a conversion between directions in the ***body coordinate frame*** and the corresponding directions in the ***global coordinate frame***

$$\vec{g} = C \cdot \vec{b}$$

Estimating the orientation of the robot within the global coordinate frame becomes a matter of keeping track of matrix ***C*** as the robot moves and turns while going about its task. This is done, as noted before, by integrating the angle rate of change provided by the gyro sensors, and updating the rotation matrix accordingly. The resulting update is given by (see https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf for a complete and thorough derivation of this equation)

$$C(t+\delta t) = C(t) + (I + \frac{sin(\sigma)}{\sigma}B + \frac{1-cos(\sigma)}{\sigma^2}B)$$

where ***I*** is a 3x3 identity matrix,

$$B = \begin{bmatrix} 0 & -\omega_{bz}\delta t & \omega_{by}\delta t \\ \omega_{bz}\delta t & 0 & -\omega_{bx}\delta t \\ -\omega_{by}\delta t & \omega_{bx}\delta t & 0 \end{bmatrix}$$

and

$$\sigma = |\vec{\omega_b}\delta t|$$

The vector $\vec{\omega_b} = [\omega_{bx} \quad \omega_{by} \quad \omega_{bz}]$ contains the rate of change measurements along the ***body coordinate frame coordinate axes***, and δt is the sampling period (the amount of time that elapsed between consecutive measurements from the gyroscopes).

Error will creep into our estimate of ***C*** because of approximation error in the integration process (the update above comes from a Taylor series expansion, and is significantly affected by the sampling period), and by noise in the gyroscopes. Therefore, you must expect that the estimate ***C(t)*** will grow further and further apart from the correct rotation matrix as time goes on. This can not be avoided, and can only be corrected by using external sources of information beyond what the gyroscopes provide.

Estimating the position of the robot is straightforward given the rotation matrix ***C***. Given an acceleration vector in the ***body coordinate frame*** which consists of the acceleration values reported by the robot's three accelerometers, we first convert that acceleration vector into the corresponding direction in the ***global coordinate frame***

$$\vec{a}_g(t) = C(t) \cdot \vec{a}_b(t),$$
$$\vec{a}_b(t) = [a_{bx}(t) \quad a_{by}(t) \quad a_{bz}(t)]$$

Given the acceleration vector in the global coordinate frame, we update the robot's velocity as follows:

$$\vec{v}_g(t+\delta t) = \vec{v}_g(t) + \delta t \cdot (\vec{a}_g(t+\delta t) - \vec{g}_g),$$

notice that we are compensating for gravity (which produces a constant acceleration on the robot) by substracting the corresponding amounts along each of the directions of the global coordinate frame. Finally, we update the robot's position with

$$\vec{s}_g(t+\delta t) = \vec{s}_g(t) + \delta t \cdot \vec{v}_g(t+\delta t)$$

Similarly to the rotation matrix ***C***, errors due to noise in the accelerometers as well as inaccuracies and approximations in the integration process (once more, significantly affected by the sampling period) will cause the velocity and position estimates to diverge over time from the true values.

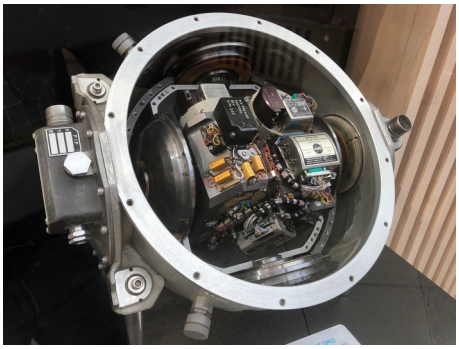
The error introduced into the estimation process for orientation, velocity, and position is called ***drift***, and can be significant over longer intervals of time. The amount of drift a particular implementation of inertial navigation will incur depends on the quality (and hence the price!) of the sensor components used to measure acceleration and rotation, as well as the quality and accuracy of the numerical software used to compute the resulting position and orientation, as an example, the table below shows a comparison of different implementations

How is this useful?

Given the problem of drift, inertial navigation can not be relied on for accurate control of an robot or automated agent over long intervals of time. However, for shorter intervals, inertial navigation can be sufficient for aiding the robot in completing its task, especially in situations where standard localization techniques can't work (e.g. lack of distinctive landmarks – airplanes mid flight, submarines, inside buildings, etc.), and GPS is either not available or unreliable (e.g. urban environment with large structures that bounce GPS signals causing error in location estimates, robotic navigation on Mars).

Another important application is motion compensation – a drone, for example, can use the information provided by inertial navigation to determine if it is being affected by external forces such as wind, and to compensate for such forces so as to remain hovering at a specified position, or to carry out very precise pre-planned motions.

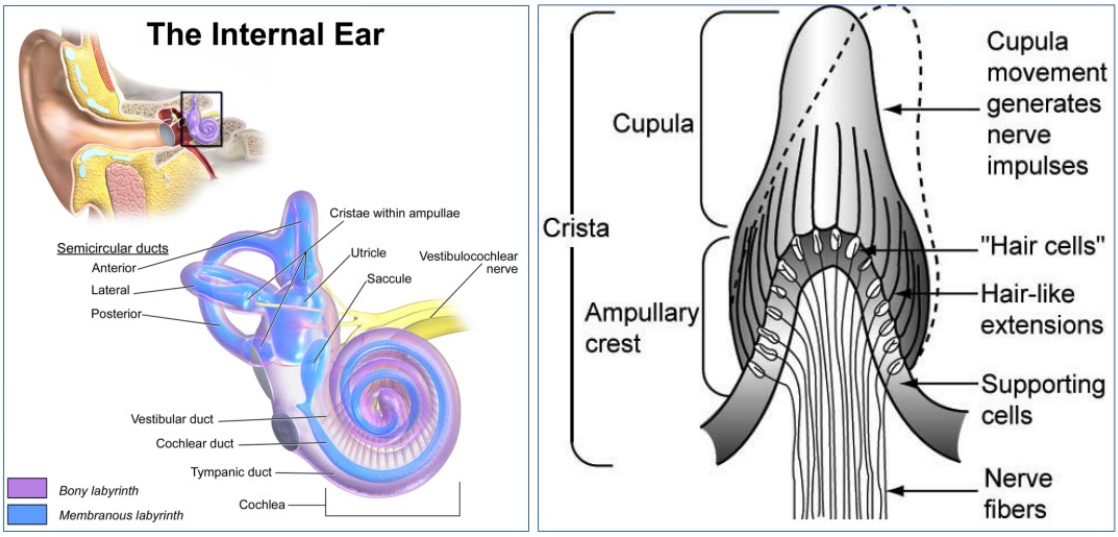
Inertial navigation provides an additional source of information to complement and improve the estimates provided by other techniques or sensors. Inertial navigation is extensively used in the airline industry, the image below shows an older model of inertial reference unit from the Apollo program



(Image: Wikimedia commons, by ArnoldReinhold, CC-BY-SA 4.0)

Human Physiology

As an interesting note, the human body uses inertial sensors extensively for everyday tasks including locomotion, maintaining balance, compensating for head/body movements while keeping gaze direction stable, and determining whether the body is undergoing acceleration or rotation. The human body uses sensors in the inner ear to achieve this.



(Images: Left – WikiJourna of Medicine, CC-BY-SA 4.0, Right – NASA, Public Domain)

The image on the left shows the inner ear sensors that provide inertial information, the semi-circular canals detect rotation along 3 axes, while two components, the *utricle* and the *saccule* detect linear acceleration (also 3 axes). These sensors use the inertia of liquid contained within these structures to detect motion: When the body undergoes acceleration or rotation, the solid components of the inner ear move but the liquid they contain tends to remain in place due to inertia. This is detected by specialized cells (right image) whose hair-like structures are deflected by the liquid, and this movement is translated into a signal used by our brain to accurately estimate acceleration and rotation.