CMPE 185 Robot

Revision for midterm - 1

Chapter 1

Robot locomotion is the collective name for the various methods that

robots use to transport themselves from place to place.

An actuator: a component of a machine that is responsible for moving and controlling a mechanism or system, it requires a control signal and a source of energy.

Most common issue w locomotion:

- Stability
- Characteristic of contact
- Type of environment

The degree of freedom (DOF) of a mechanical system is the

number of independent parameters that define its configuration.

Translation and rotation:

- Moving up and down (elevating/heaving);
- Moving left and right (strafing/swaying);
- Moving forward and backward (walking/surging);
- Swivels left and right (yawing);
- Tilts forward and backward (pitching);
- Pivots side to side (rolling).

The gait is characterized as the distinct sequence of lift and release events of the individual legs.

- It depends on the number of legs
- The number of possible events N for a walking machine with k kegs is

$$N = (2k - 1)!$$

- With this frame, we describe the robot state as position + orientation

$$\xi_I = [x, y, \theta]$$

Coordinate Change – Example

- Suppose a range sensor mounted on a robot detects an obstacle at position $[1, 3]^T$. Suppose the robot is at position $[2, 0]^T$ in the inertial frame with orientation $\pi/3$. Find the position of the obstacle in the inertial frame.
- Step 1: find the rotation matrix

$$R = \begin{bmatrix} \cos\frac{\pi}{3} & -\sin\frac{\pi}{3} \\ \sin\frac{\pi}{3} & \cos\frac{\pi}{3} \end{bmatrix}$$

- Step 2: find the position

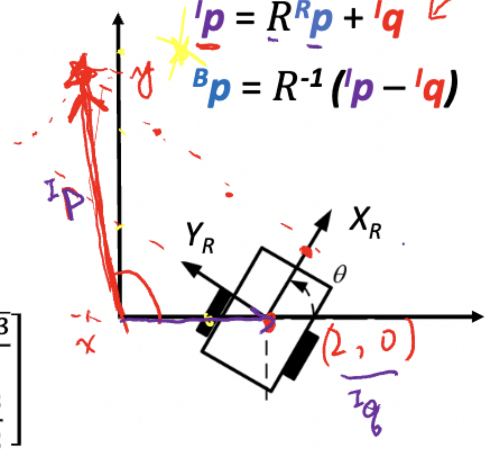
$${}^I\mathbf{q} = \begin{bmatrix} 2 \\ 0 \end{bmatrix},\ {}^R\mathbf{p} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$${}^I\mathbf{p} = \begin{bmatrix} \cos\frac{\pi}{3} & -\sin\frac{\pi}{3} \\ \sin\frac{\pi}{3} & \cos\frac{\pi}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{5}{2} - \frac{3\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} + \frac{3}{2} \end{bmatrix}$$

$$R \cdot {}^Rp + {}^Iq$$

$${}^I\mathbf{p} = R\,{}^R\mathbf{p} + {}^I\mathbf{q}$$

$${}^B\mathbf{p} = R^{-1}\left({}^I\mathbf{p} - {}^I\mathbf{q}\right)$$



Transformations

- Let's put our equations in matrix form:

$$\underbrace{\begin{pmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{pmatrix}}_{\dot{\xi}_I} = \underbrace{\begin{pmatrix} \cos(\theta) & 0 & 0 \\ \sin(\theta) & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R^{-1}(\theta)} \underbrace{\begin{pmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{pmatrix}}_{\dot{\xi}_R} \Leftrightarrow \begin{cases} \dot{x}_I = \dot{x}_R \cos\theta \\ \dot{y}_I = \dot{y}_R \sin\theta \\ \dot{\theta}_I = \dot{\theta}_R \end{cases}$$

- Or we can rewrite:

$$\dot{\xi}_I = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}$$

Differential Inverse Kinematics – Decomposition

- Step 1: turn so that the wheels are parallel to the line between the original and final position of the robot origin.

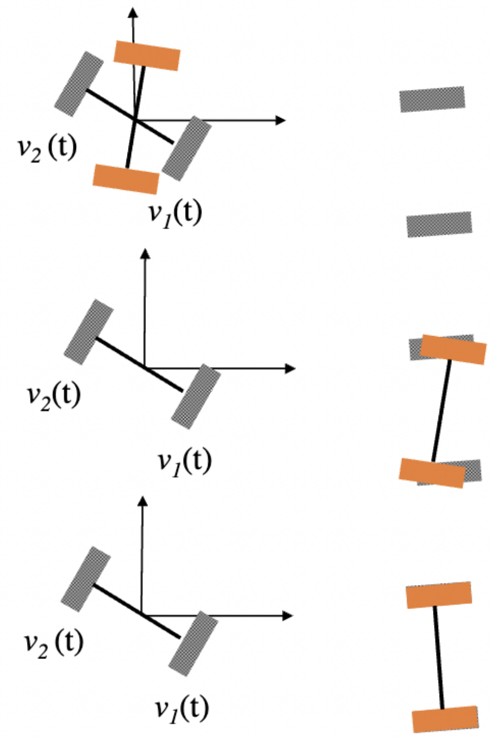
$$v_1(t) = -v_2(t) = v_{\max}$$

- Step 2: drive straight until the robot's origin coincides with the destination

$$v_1(t) = v_2(t) = v_{\max}$$

- Step 3: rotate again in order to achieve the desired final orientation

$$-v_1(t) = v_2(t) = v_{\max}$$



A rotary encoder, also called a shaft encoder, is an electromechanical device that converts the angular position or motion of a shaft or axle to analog or digital output signals.

Absolute encoders: measure the current orientation of a wheel

Incremental encoders: measure the change in orientation of a wheel

Dead reckoning

- The process of estimating one's current position based upon a previously determined position and advancing that position based upon known speed, elapsed time, and course.

'

$$\Delta x = \Delta s \cos(\theta + \frac{\Delta\theta}{2})$$

$$\Delta y = \Delta s \sin(\theta + \frac{\Delta\theta}{2})$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{2L}$$

$$\Delta s = \frac{\Delta s_l + \Delta s_r}{2} \qquad \Delta\theta = \frac{(\Delta s_r - \Delta s_l)}{2L}$$

- If my robot starts at the origin (position = $[0, 0]_{T,}$ and orientation is 0), where is it located after 0.1s, given that 10 ticks were recorded for the right wheel and 6 ticks for the left wheel. The wheel radius is 2m, the total ticks per revolution is 100. The distance between wheels is 4m.

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \frac{\Delta s_r - \Delta s_l}{4L}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \frac{\Delta s_r - \Delta s_l}{4L}) \\ \frac{\Delta s_r - \Delta s_l}{2L} \end{bmatrix}$$

For each wheel:

$$\Delta s_l = 2\pi r \frac{\Delta tick_l}{N}$$

$$\Delta s_r = 2\pi r \frac{\Delta tick_r}{N}$$

Control architectures should consider:

- Code Modularity
  - Allows programmers to interchange environment types sensors, path planners, propulsion, etc.
- Localization
  - Embed specific navigation functions within modules to allow different levels of control (e.g., from task planning to wheel velocity control)

Temporal Decomposition:

- Facilitates varying degrees of real-time processes

Control Decomposition:

- Defines how modules should interact: serial or parallel?

- ***State*** = Representation of what the system is currently doing
- ***Dynamics*** = Description of how the state changes
- ***Reference*** = What we want the system to do
- ***Output*** = Measurement of (some aspects of the) system
- ***Input*** = Control signal
- ***Feedback*** = Mapping from outputs to inputs

Control Theory = How to pick the input signal u?

Goal may be to design

- Regulating control: maintain a fixed output
- Servo control: follow a changing reference
- so that the system:
  - is stable (e.g., bounded-input-bounded-output)
  - rejects disturbances
  - is robust to parameter changes

PI Controller

Pros:

- accelerates the movement of the process towards setpoint
- eliminates the residual steady-state error

Cons:

- may result in overshooting the set point

Why derivative controllers?

- Damping friction is a force opposing motion, proportional to velocity
- Try to prevent overshoot by damping controller response.
- Derivative control is “happy” the error is not changing (Things not getting better, but not getting worse either)
- Estimating a derivative from measurements is fragile, and amplifies noise

PID Control

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

- **P**: contributes to stability, medium-rate responsiveness
- **I**: tracking and disturbance rejection, slow-rate responsiveness. May cause oscillations
- **D**: fast-rate responsiveness. Sensitive to noise
- **PID**: by far the most used low-level controller.
  - However, stability is not guaranteed

Parameter tuning is very important for PID controller

- Manual tuning

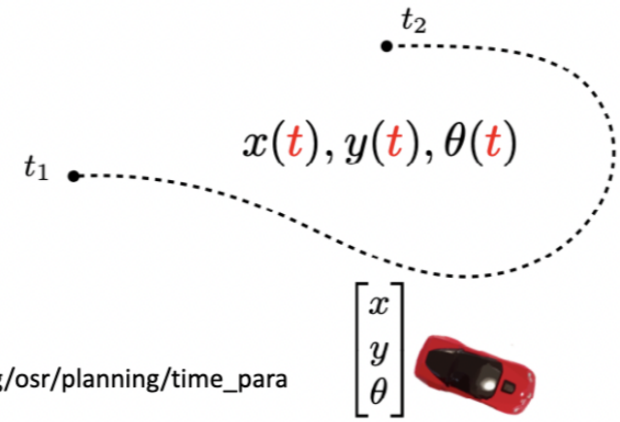
1. Increase P term until performance is adequate or oscillation begins
2. Increase D term to dampen oscillation
3. Go to 1 until no improvements possible.
4. Increase I term to eliminate steady-state error.

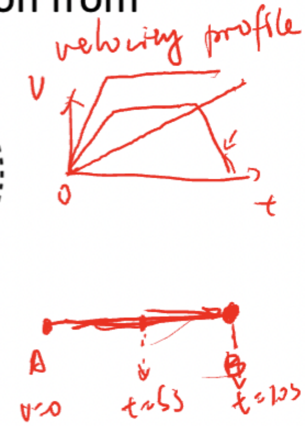
Closed Loop Response	Rise Time	Overshoot	Settling Time	Steady State Error
Increase K_P	Decrease	Increase	Small change	Decrease
Increase K_I	Decrease	Increase	Increase	Eliminate
Increase K_D	Small change	Decrease	Decrease	Small change

By properly tuning the PID gains

How do we define a reference?

- **Option 1:** Time-parameterized *trajectory*
  - Pro: Useful if we want the robot to respect time constraints
  - Con: Sometimes we care only about deviation from reference

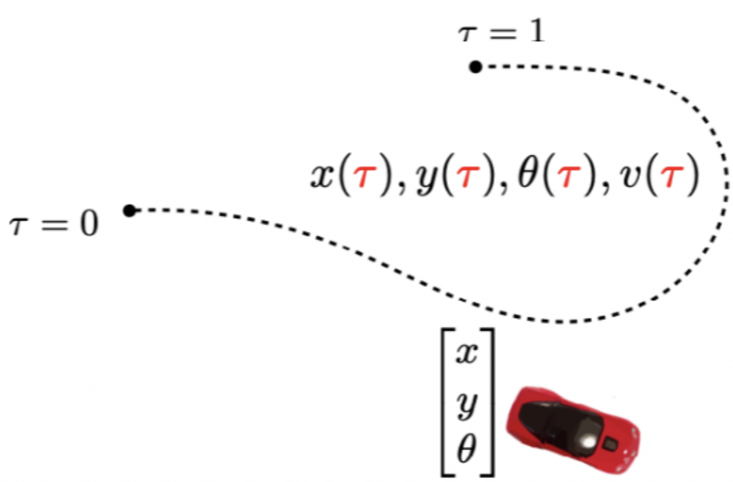




http://www.osrobotics.org/osr/planning/time_parameterization.html

How do we define a reference?

- **Option 2:** Index-parameterized *path*
  - Pro: Useful for conveying the shape you want the robot to follow
  - Con: Can’t control when robot will reach a point



Classification of Sensors

- Robot = sensors + actuators
- Sensors are the key components for perceiving the environment

What:

- **Proprioceptive sensors**
  - measure values internally to the system (robot)
  - e.g. motor speed, wheel load, heading of the robot, battery status
- **Exteroceptive sensors**
  - information from the robots environment
  - distances to objects, intensity of the ambient light, unique features.

How:

- **Passive sensors**
  - Measure energy coming from the environment; very much influenced by the environment
- **Active sensors**
  - emit their proper energy and measure the reaction
  - better performance, but some influence on environment

Sensors basic characteristics:

Range:

- Lower and upper limits
- Ex: IR range sensor measures distance between 10 & 80 cm

Resolution:

- Minimum difference between two measurements
- For digital sensors, it is usually the A/D resolution

Dynamic range

- Used to measure spread between lower and upper limits of sensor inputs
- Formally, it is the ratio between the maximum and minimum measurable input, usually in decibels (dB)
- Ex: A sonar range sensor measures up to a max distance of 3m, with smallest measurement of 1cm

Linearity

- A measure of how linear the relationship between the sensor's output signal and input signal
- Linearity is less important when signal is treated after with a computer

Bandwidth or Frequency

- The speed with when a sensor can provide a stream ofreadings
- Usually there is an upper limit depending on the sensor and the sampling rate
  - ex: sonar takes a long time to get a return signal
- Higher frequencies are desired for autonomous control
  - ex: if a GPS measurement occurs at 1 Hz and the autonomous vehicle uses this to avoid other vehicles that are 1 meter away

Sensitivity

- Ratio of output change to input change
  - ex: range sensor will increase voltage output 0.1 V for every cm distance measured

Accuracy

- The difference between the sensor's output and the true value.

Precision

- The reproducibility of sensor results

System Errors:

- Random Error

    - Non-deterministic
    - Not predictable
    - Usually described probabilistically
- Systematic Error
    - Deterministic
    - Caused by factors that can be modeled
        - ex: optical distortion in camera

Sensor uncertainty can be calculated using the given formula and using Gaussian distribution.

An Inertial Measurement Unit (IMU) is a combination of sensors that uses gyroscopes, accelerometers and sometimes magnetometers to estimate the linear and angular motion of a moving vehicle with respect to an inertial frame, and the earth's magnetic field.

Image filtering:

- Linear: every pixel is replaced by a linear combination of its neighbors
- Shift-invariant: the same operation is performed on every point on the image.