ROBOT PERCERTION

Perception is the ability of the robot to perceive its environment.

A robot have → Tracking ability
↳ Scene perception
↳ self location perception

Sensing systems

Robotic perception relies on the quantity and quality of information that it can obtain from sensors.

- PROPIOCEPTIVE SENSORS: collect info that are internal of the robot.
- EXTEROCEPTIVE SENSORS: collect info on the robot surrounding.

NOTE: PROP. give us an idea of the state of the robot (where it should be in theory), EXT give a feedback from the environment to have a more accurate knowledge.

PREDICTION: $\hat{x}(t) = Vt + \varepsilon_t + x_0 = Vut + x_0$ PROP. sensors

REAL: $\dot{x}(t) = V \Rightarrow x(t) = Vt + x_0$ EXT. sensors

ESTIMATION ERROR: $\tilde{x}(t) = \hat{x}(t) - x(t) = \varepsilon_t$

Using only PROP. sensors $\varepsilon_t \to \infty$ because we can never know exactly our position.

Also EXT. sensors have error: it includes random and constant errors.

STATE ESTIMATION

For a rigid body a state that describe its posture is defined as $S(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \in \mathbb{R}^3$

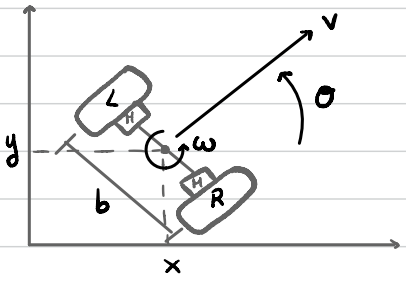
In general, a state is a vector $\in \mathbb{R}^n$ of variables that describe completely a robot.

2 FONDAMENTAL EQUATIONS: each robotic system can be described by 2 principal models:

- *Motion model* $\dot{s} = f(s, u)$, or in a discrete way $S_{k+1} = AS_k + Bu_k$.*
  ↳ Describe how the state evolves through time. It's a prediction model based on control input u (e.g. wheels velocity).
- *Measure model* $z = h(s)$, or in its linear form $z_k = HS_k$
  ↳ Describe what sensor measure (z) in a state s. Its our observation model on the environment. IT give us information (FEEDBACK) about our real state.

* Matrix A its the derivative of f wrt s ⇒ how s_k influences s_{k+1}
Matrix B its the derivative of f wrt u ⇒ how input u_k influences s_{k+1}

Example with a Differential Drive Robot (unicycle)



$c = \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}$ Angular velocity of right and left wheel

$\dot{s} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(s, u)$

$V_R = \omega_R r \quad V_L = \omega_L r \quad r$: wheel radius

$\omega = \frac{V_R - V_L}{b}$

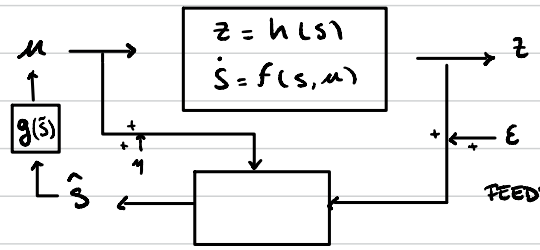
$$\dot{s} = f(s,u) = \begin{bmatrix} \cos(\theta) V \\ \sin(\theta) V \\ \omega \end{bmatrix} = \begin{bmatrix} C_\theta & 0 \\ S_\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \qquad S_{k+1} = S_k + \begin{bmatrix} C_{\theta k} & 0 \\ S_{\theta k} & 0 \\ 0 & 1 \end{bmatrix} \Delta t \, u_k$$

How compute the best state estimation $\hat{s}$

The idea is to use measures z to correct hypotesis given by motion model.

OBJECTIVE: found initial state S_0 of the system given → measures z, → input u

SYSTEM



we know z thanks to EXT. sensors

we know u thanks to PROP. sensors

(1) At initial time ($k=0$) the system is at state S_0.
Sensors made measure z_0.
↳ Using measurement equation we got: $z_0 = H S_0$

(2) At first step ($k=1$) we give a command u_0 and system state is S_1. Using motion equation (because at this point robot is moving): $S_1 = AS_0 + Bu_0$ [$S_{k+1} = AS_k + Bu_k$]
At time S_1 sensors made a new measure: $z_1 = HS_1$ [$z_k = HS_k$]

(3) We have to substitute because we want that everything depends on S_0:
↳ $z_1 = H(AS_0 + Bu_0) = (HA)S_0 + (HB)u_0$

(4) Matrix form

$$\begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \overbrace{\begin{bmatrix} H \\ HA \end{bmatrix}}^{O} S_0 + \overbrace{\begin{bmatrix} 0 \\ HB \end{bmatrix}}^{T} u_0$$

→ matrix that maps the effect of command on measures

OBSERVABILITY MATRIX → if it is full rank, the system is observable
↳ Observability means that the initial state of a system can be determined

More compact $Z = O S_0 + T U$

To solve the equation we have to compute the inverse of matrix O. Since is often not square and invertible, we have to use the pseudoinverse:
$\hat{S}_0 = (O^T O)^{-1} O^T (z - TU)$ ⇒ Best estimation of S_0 based on ALL measurments.

Actual Prediction

$\underbrace{z_0 - \hat{z}_0}_{\text{INNOVATION}} = H(s_0 - \hat{s}_0)$ Base of ESTIMATION PROBLEM $\Rightarrow \underbrace{\hat{s}_0 = \hat{s}_0^- + H^+(z_0 - \hat{z}_0^-)}_{\text{We take our Hypotesis } \hat{s}_0^- \text{ and we correct it with the new information } H^+(z_0 - \hat{z}_0^-)}$ STATIC ESTIMATOR

CASE DETERMINE s_0 USING GPS DATA

we know $z_0 = Hs_0$, $z_1 = Hs_1$, Input and model

In this case state vector represents position and velocity $\longrightarrow s = [x, v]$

$x_1 = x_0 + v_0 \Delta t$ } How robot moves through time

$v_1 = v_0$

OBJECTIVE: found A that satisfies $\underbrace{[x_1, v_1]}_{s_1} = A\underbrace{[x_0, v_0]}_{s_0}$

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

$z_1 = Hs_1 = H(As_0 + B u_0)$ in this way z_1 is in function of s_0

$$\underbrace{\begin{bmatrix} z_1 \\ z_1 - HBu_0 \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} H \\ HA \end{bmatrix}}_{O} \underbrace{s_0}_{x} \Rightarrow \text{Linear Relation}$$

$H = [1\ 0]$ since GPS measure only position and state is [position; velocity]

Observability Matrix $O = \begin{bmatrix} H \\ HA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & \Delta t \end{bmatrix} \Rightarrow$ Invertible $\forall \Delta t \neq 0$

At this time we have an initial guess $\hat{s}_0^-$.

If I compute $\hat{s}_0 = \hat{s}_0^- + W(z_0 - \hat{z}_0)$ I update my knowledge of s_0^- with informations $W(z_0 - \hat{z}_0^-)$. [W is a generic matrix]

The knowledge of the model is $\hat{s}_1^- = A\hat{s}_0 + Bu_0$

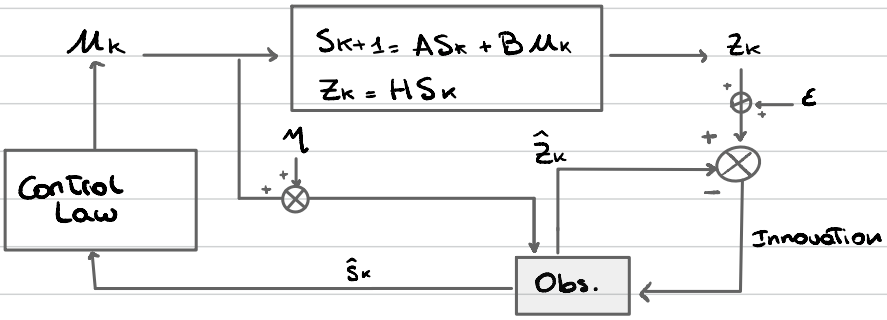
↳ If I put the two equations together:

$$\hat{s}_1^- = A\hat{s}_0^- + AW(z_0 - \hat{z}_0^-) + Bu_0 = A\hat{s}_0^- + Bu_0 + \overbrace{L}^{AW}(z_0 - \hat{z}_0^-) \Rightarrow \text{consider also DYNAMIC}$$

($H\hat{s}_0$ under z_0; $H\hat{s}_0$ over $\hat{z}_0^-$)

$\tilde{s}_1 = \hat{s}_1 - s_1 = A\tilde{s}_0 + LH\tilde{s}_0 = (A - LH)\tilde{s}_0$ purpose: bring error To zero $\Rightarrow$ system will be asyntotically stable

LUENBERG OBSERVER



Up to now we have tried to found the best estimation for S_0 using a block of past measurements. Now we have to consider that the robot is actually moving, so we need a system that updates its estimation at each step k. This is what Luenberg Observer do. It is described by:

$$\hat{S}_{k+1} = AS_k + Bu_k + L(z_k - H\hat{S}_k)$$

- $AS_k + Bu_k$ = Prediction, what should be next state based on previous estimation S_k
- $(z_k - H\hat{S}_k)$ = Innovation, z_k is real measure and HS_k is predicted measure. The difference is PREDICTION ERROR.
- L = Gain Matrix, decides how much we "trust" of prediction Error. It use error to correct our prediction.

Luenberg observe is an ONLINE DYNAMIC approach to update estimation with new informations. It uses the last estimation TO correct the current one.

PSEUDOINVERSE METHOD

Uses all data to compute backward the best estimation. In the pseudoinverse $(H^TH)^{-1}H^T$ all measurement errors are used to correct the estimation of initial state.

[we use D, H, a but are equivalent TO z', O, so]

- $D = [d_1, \dots d_i]$ measures
- $(d_i - h_i a)$ ERROR for each measure
- Total Error = $\Sigma (d_i - h_i a)^2$ guarantee non-negative errors
- Matrix Form $(D-Ha)^T(D-Ha)$

OBJECTIVE: Find value of a that minimize difference between D(real) and Ha(prediction).

(1) FOUND MINIMUM WITH THE DERIVATIVE : compute derivative wrt a and set equal to zero

$$(D-Ha)^T(D-Ha) = D^TD - D^THa - a^TH^TD + a^TH^THa$$

$d/da \rightarrow$

$$-2H^TD + 2H^THa = 0$$

$$H^THa = H^TD$$

dividing H^TH = left multiply by $(H^TH)^{-1}$, $(H^TH)^{-1}(H^TH) = I$

$$(H^TH)^{-1}H^THa = (H^TH)^{-1}H^TD$$

$$a = (H^TH)^{-1}H^TD$$

↳ this solution guarantee the minimum quadratic error

CALIBRATION

Physical Quantity Q → [Instrument] → z → $\hat{z} = f(Q)$ MEASUREMENT PROCESS

($\hat{z}$: measurement result; Q: measurand)

Measurment Process allow To associate a quantity To a physical phenomenon.
To make a measurment you need a REFERENCE and a SCALE.
You always have uncertainties → Quantization
→ Definitional Uncertainty
→ Interaction uncertainty

ACCURACY: how close we are to the true value (for a single measure)
PRECISION: represents the dispersion around the value. Its a property of the instrument.
TRUENESS: compute average of measures and compute the difference between this average and true value.

CALIBRATION PROCESS:

(1) $\frac{1}{n}\sum_i z_i = \bar{z}$ mean of measures

(2) $b = \bar{z} - z_a$ Bias, difference between real (z_a) and measured

(3) $z_i^* = z_i - b$ Define a new measure

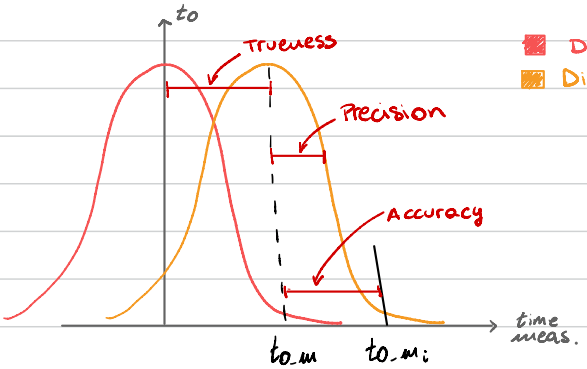
(4) Check $\frac{1}{n}\sum_i z_i^* = \frac{1}{n}\sum_i (z_i - b) = \left(\frac{1}{n}\sum_i z_i\right) - b = \bar{z} - b = z_a$ => you converge To The actual value

↳ You need A LOT OF DATA to get closer and closer to actual value.

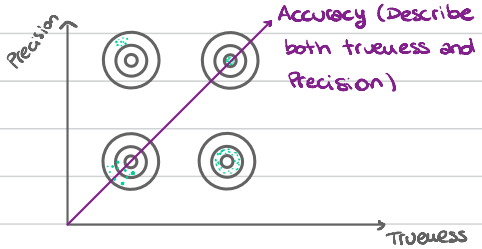
---

compute the distance $d = \frac{(t_1 - t_0)}{2} c$ t_1 = time of return t_0 = time of start of signal

$t_{0\_m} = t_0 + \varepsilon$ where $\varepsilon = \frac{n_{\delta t_i}}{n}$ $\frac{1}{n}\sum_i \varepsilon_i \simeq \emptyset$ => CALIBRATION



Distribution with calibration
Distribution without calibration



$d_m = d + \frac{\varepsilon_1 - \varepsilon_0}{2} c$ $\frac{\varepsilon_1 - \varepsilon_0}{2} c = \eta$ uncertainty in our measurment system

Find uncertainty in the distance → TYPE A ANALYSIS: experiment and collect data.
→ TYPE B ANALYSIS: theoretical tools.

FIND UNCERTAINTY

After calibration, bias is removed. Now our averaged measure converges to the true value. However, each single measure is affected by a random error, that create a dispersion around a true value (due to the instrument precision).

PROBLEM: how we can quantify the DOUBT that we have on the final result? How we can an interval in which is more likely to found true value d?

SOLUTION: define an UNCERTAINTY INTERVAL. Thanks to calibration, this interval is centered in our best estimation (d_m) $\Rightarrow d \in [d_m - \Delta d \; ; \; d_m + \Delta d]$

$d_m \in [d - \Delta d \; ; \; d + \Delta d]$

To FOUND Δd we use Type A/B Analysis.

Type A Analysis $\Rightarrow$ Empirical Method, make experiments

1. DEFINE UNCERTAINTY WITH MSE

The idea is to define uncertainty Δd like a measure of a mean dispersion of our measures (d_i) around the true value (d). The standard metric for this is MSE. We compute Δd as STANDARD UNCERTAINTY: $\Delta d = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_i (d_i - d)^2}$

↳ This value define an interval around our estimate in which the true value is very likely to be found.
If we want a greater confidence we can use EXPANDED UNCERTAINTY: $K \cdot \Delta d$ with $K > 1$

2. CONNECTION WITH LEAST SQUARES PROBLEM

If we consider d as the unknown parameter to estimate, we can write the problem of minimization of squared error, that lead to the solution of PSEUDOINVERSE. If we use vector of ones ($\mathbb{1}$) as matrix H, the solution $\hat{d} = (H^T H)^{-1} H^T D$ can be semplified in the computation of the mean of measures: $\hat{d} = \frac{1}{n}\sum d_i$, that is our correct and intuitive measure with no bias.

3. SAMPLE VARIANCE

MSE has a problem: require the knowledge of d, but we can NEVER know the true value of a measure. The solution is to compute the SAMPLE VARIANCE. This formula estimate the dispersion using SAMPLE MEAN ($\hat{d}$) instead of the true value d. To obtain an impartial estimation, we divide by $(n-1)$ instead n.

$s^2 = \frac{1}{(n-1)} \sum_i (d_i - \hat{d})^2$ → compute uncertainty only with measures collected.

Type B Analysis ⟹ Theoretical Approach

This approach is useful when we can't do a lot of experiments. We use our prior knowledge and then we formalize it with probability.

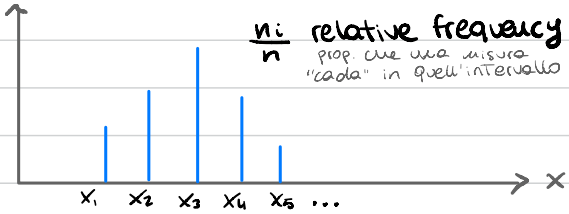
↳ Estimation of uncertainty modeling it using probability theory.

Measure ⟹ Random variable ⟸ Our knowledge of this variable is represented by a PROBABILITY DENSITY FUNCTION (PDF).

PDF $p(x)$: graph that indicates which values our variable is more likely to assume.

EXPECTED VALUE $E(x)$: is the "best hypotesis" that we can do on the value of the variable.

↳ How to apply? After calibration, $E(x)$ is the BEST approximation to real value:

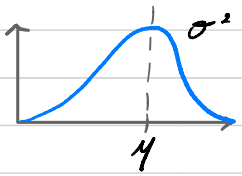


$$\begin{cases} E(x) = \sum_i \left(\frac{n_i}{n}\right) x_i & \text{DISCRETE (from histogram)} \\ E(x) = \int_{-\infty}^{+\infty} x\, p(x)\, dx & \text{CONTINUOUS CASE} \end{cases}$$

VARIANCE AND PRECISION: variance is a measure of the dispersion of the PDF around true mean value.

↓ σ^2 ↑ precision ; ↑ σ^2 ↓ precision ⟹ Prec. = $1/\sigma^2$

GAUSSIAN DISTRIBUTION: is usually used because (for the Central Limit Theorem) when we sum a lot of independent errors, result tends always to be a Gaussian.



BAYES THEOREM: allow to update our knowledge

$$Pr(A_i|B) = \frac{Pr(B|A_i)\, Pr(A_i)}{\sum Pr(B|A_i)\, R(A_i)}$$

How to analyze and describe PDF

We want to characterize uncertainty using MOMENTS.

Summarizing, we can *model* our uncertainty on a *measure* x (random variable) using a PDF $p(x)$. Expected value $E(x)$ is the best estimation, and variance (σ^2) says the precision of this estimation. To completely describe a PDF we need to compute its STATISTICAL MOMENTS.

MOMENTS OF A DISTRIBUTION

A statistical *moment* is the expected value of a power of the random variable.

Moment of order n: $E(x^n)$

- FIRST MOMENT $E(x)$: *mean* μ
- SECOND MOMENT $E(x^2)$: dispersion. Useful to compute the variance
- CENTRAL MOMENTS $E((x-\mu)^n)$: *moments* computed with respect to the *mean*. Second central moment is the variance $\sigma^2 = E((x-\mu)^2)$

If we know all the *moments* of a distribution, we have a complete knowledge about that distribution.

CHARACTERISTIC FUNCTION $\Phi(f)$

It is the Fourier transform of the PDF. $\Phi(f) = E(e^{-j2\pi fx}) = \int_{-\infty}^{+\infty} p(x) e^{-j2\pi fx} dx$

It's derivatives computed in $f=0$ give us moments of the distribution.

$\left.\frac{d^n \Phi(f)}{d^n f}\right|_{f=0} = (-j2\pi)^n E(x^n)$ => If we have $\Phi(f)$ of a PDF, we can compute any *moment* $E(x^n)$ deriving it n times.

APPLICATION TO GAUSSIAN DISTRIBUTION

Moments of a gaussian: $E((x-\mu)^n) = \begin{cases} 0 & \text{if } n \text{ is odd} \\ \sigma^n (n-1)!! & \text{if } n \text{ is even} \end{cases}$ => Gaussian is simmetric

Standard Gaussian $x \sim N(0,1)$: it's useful standardize each gaussian variable. If we have a variable y with μ and σ^2, we can standardize it into a variable z with $\mu = 0$ and $\sigma^2 = 1$

↳ $z = \frac{y-\mu}{\sigma^2}$

↳ If we want to see the probability that $a < z < b$ we can simply look in table.

$Pr(a \le y \le b) = Pr\left(\frac{a-\mu}{\sigma} \le z \le \frac{b-\mu}{\sigma}\right)$

OTHER USEFUL FORMULA:

Median: center of a distribution => $Pr(x \le \bar{y}) = Pr(x > \bar{y}) = 0.5$

Variance: $\sigma^2 = E((x-\mu)^2) = E(x^2) - [E(x)]^2$

UNCERTAINTY OF A VARIABLE ⟹ UNCERTAINTY OF A SYSTEM

If we have a state with more variables $s = [x, y, \theta]$ (for example the state of the robot) have to describe uncertainty on x, y, θ simultaneously.

JOINT PMF $P(x, y)$ for discrete variables: gives the probabily that first variable is exacty x and second is y.

JOINT PDF $P(x, y)$ for continuous variables: surface in a 3D space, where the height in a point (x, y) represents density probability of that pair of values.

MARGINALIZATION: used if we have joint PDF but we want to come back to probabilities of x and y.
We integrate on all the value of the variable we want to eliminate.
↳ $P(x) = \int_{-\infty}^{+\infty} P(x, y)\, dy$

QUANTIFY CORRELATION

How we can measure numerically if and how 2 uncertainty are linked?

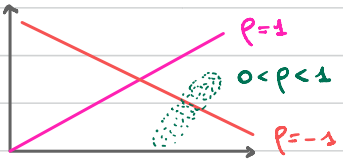
$Pr(x|y) = Pr(x)$ if x, y are independent

$Pr(x|y) \geq Pr(x)$ if x, y are correlated. So in this case we are accumulating knowledge

CENTRAL JOINT MOMENT: $E\left((x-\mu_x)^r (y-\mu_y)^q\right)$
- if $r=1, q=0 \Rightarrow 0$
- if $r=0, q=1 \Rightarrow 0$
- if $r=1, q=1 \Rightarrow$ COVARIANCE

Covariance measures says how 2 variables "moves together" with respect to their mean.

$$cov(x, y) = \begin{cases} > & \text{positive correlation} \\ 0 & \text{no correlation} \\ < & \text{negative correlation} \end{cases}$$



The normalization of $cov(x,y)$ is the correlation coefficient $\rho = \frac{cov(x,y)}{\sqrt{\sigma_x^2 \cdot \sigma_y^2}} \in [-1, 1]$

! correlation ≠ independence

Two variables are independent iff $P(x, y) = P(x, y)$

If x, y are independent $\Rightarrow cov(x, y) = 0$

If $cov(x, y) = 0 \not\Rightarrow x, y$ are independent

↳ Not always, because the level of dependency can be not linear

GAIN INFORMATION WITH BAYES

Bayes theorem allow us to gain information and make estimation better thanks to the measurement that we collect.

$$\underbrace{p(x \mid x_m)}_{\text{Posterior}} = \frac{\overbrace{P(x_m \mid x)}^{\text{Likelihood}}\overbrace{P(x)}^{\text{Prior}}}{P(x_m)} = \frac{P(x_m \mid x)P(x)}{\int P(x_m \mid x)P(x)\,dx}$$

Keep collecting measurement allow us to reduce uncertainty gaining knowledge. The posterior of a measurement become the prior of the measurement taken later.

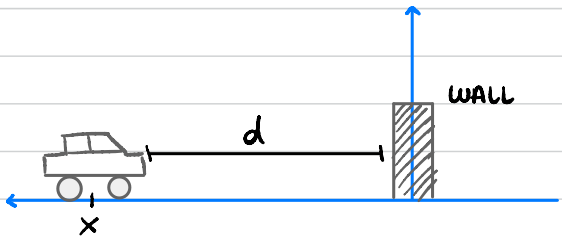
If we don't have ANY prior knowledge, we consider x as a unknown deterministic variable. We use MAXIMUM LIKELIHOOD ESTIMATION.

PARAMETRIC FUNCTION (depends on θ_m) to apply Bayes Theorem:

$$P(\theta \mid \theta_m) = \frac{\overbrace{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(\theta-\theta_m)^2}{\sigma^2}}}^{\text{Likelihood}} \cdot \overbrace{\frac{1}{2\pi}}^{\text{Prior}}}{\int_0^{2\pi} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(\theta-\theta_m)^2}{\sigma^2}} \frac{1}{2\pi}\,d\theta} = f(\theta_m)\, e^{-\frac{1}{2}\frac{(\theta-\theta_m)^2}{\sigma^2}}$$

↳ we can put it outside the integral, it doesn't depend on θ

EXAMPLE



Δt: sampling time of the sensor

$\dot{x} = v$ velocity sensed from the tacometer

Velocity measured at time k is $v_{k,m} = v_k + \varepsilon_k$ with $\varepsilon_k \sim N(0, \sigma^2)$ measure error

Robot use these estimations of velocity measures to estimate its position along time.

At every Δt it update its position estimation $\hat{x}$. $\hat{x}_{k+1} = \hat{x}_k + \Delta t \cdot v_{k,m}$

Substituting in sensor model: $\hat{x}_{k+1} = \hat{x}_k + \Delta t(v_k + \varepsilon_k) = \hat{x}_k + \Delta t \cdot v_k + \Delta t \cdot \varepsilon_k$

We can define $\eta_k = \Delta t \cdot \varepsilon_k$ is the process noise. It's the error that "enter" in our system at each step due to imperfect measure. And $\eta_k \sim N(0, \Delta t^2 \sigma^2)$

Starting from a know position x_0: $\hat{x}_{k+1} = x_0 + \Delta t \sum_0^k v_i + \sum_0^k \eta_i$

↳ After k+1 steps we have accumulated $\Sigma \eta_i$ random errors.

Estimation error: $\tilde{x}_{k+1} = \underbrace{\hat{x}_{k+1}}_{\text{Where robot thinks to be}} - \underbrace{x_{k+1}}_{\text{Where robot actually is}} = \sum_0^k \eta_i \rightarrow N(0, (k+1)\Delta t^2 \sigma^2)$

↳ using proprioceptive sensors

Measurements Model

We have 2 "words" that iteratively evolves in parallel:

1) ACTUAL SYSTEM: what's really happen. Robot moves and its real position x_k evolves. A sensor measures this position, but measure z_k is affected by random noise β_k.

2) ESTIMATION SYSTEM: what robot beliefs it's happening. Its estimation of position $\hat{x}_k$ evolves using noisy measurements collected by sensors. Doing this it accumulates error γ_k.

HOW TO USE z_k TO CORRECT ROBOT ESTIMATION γ_k:

(1) Prediction (Prior)

- Starting from known initial position x_0, robot uses its *model* to predict where he will be in the next step. $\rightarrow \hat{x}_1 = x_0 + \Delta t\, v_0 + \gamma_0$
- $\hat{x}_1$ is a random gaussian variable $\hat{x}_1 \sim \mathcal{N}(x_0 + \Delta t\, v_0, \Delta T^2 \sigma^2)$
- $\hat{x}_1$ is a PRIOR knowledge, the best hypotesis on the actual position of the robot.

(2) correction (Likelihood and Prior)

- A sensor give a *measure* z_1. $\rightarrow z_1 = x_1 + \beta_1$
- Innovation: robot make a comparison between $\hat{z}_1$ and $\hat{x}_1$. The difference is called innovation. $\rightarrow z_1 - \hat{z}_1$

  If $z_1 - \hat{z}_1 \neq 0$ we have to correct our prediction $\hat{x}_1$.
- To formalize correction we use Joint Probability.

  $y = \begin{bmatrix} \hat{x}_1 \\ z_1 \end{bmatrix}$ is a vector of random variables.

Since $\hat{x}_1$ and z_1 are gaussian, joint PDF $P(y)$ is gaussian. $P(y)$ contains informations about uncertainty on prediction, uncertainty on *measure* and correlation between them.
COVARIANCE MATRIX entirely describes uncertainty of the system

$$C(y) = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_z \\ \rho\sigma_x\sigma_z & \sigma_z^2 \end{bmatrix}$$

Final objective is to compute <u>POSTERIOR $P(\hat{x}_1 | z_1)$</u>, our updated knowledge after seeing the *measurment*.

If vector $y = \begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}(\mu_y, C(y))$, so also $P(x|z)$ is a gaussian distribution.

$$\begin{cases} \mu_{x|z} = \mu_x + \underbrace{C(x,z)\,C(z)^{-1}}_{K}(z - \mu_z) & \text{It's our new best estimation} \\ C(x|z) = C(x) - \underbrace{C(x,z)\,C(z)^{-1}\,C(z,x)}_{K} & \text{It's our new uncertainty} \end{cases}$$

(Prior; KALMAN GAIN; Innovation — Prior; uncertainty Reduction)

KALMAN GAIN: it's a matrix that acts like optimal gain. Weights innovation and say how much we have to correct our prior estimation.

$\uparrow K \downarrow C(z)$ less uncertainty $\rightarrow$ good sensor

$\downarrow K \uparrow C(z)$ more uncertainty $\rightarrow$ noisy sensor

FISHER INFORMATION

This matrix is equal to $C(y)^{-1}$. It shows how different sources of information are weighted in estimation computation.

MAHALANOBIS DISTANCE

It's equal to the esponential of gaussian PDF: $(y-\mu_y)^T C(y)^{-1} (y-\mu_y)$. It measures how much a point y is far from mean μ_y, taking care of uncertainty $C(y)$. This mean that components with more uncertainty, weight less in the final error computation.

Measurement Loop: (summary)

1. PREDICTION: using dynamic model to compute the prior estimation μ_x, $C(x)$
2. CORRECTION: receive a measure z. Compute gain K and update estimation to obtain $\mu_{x|z}$, $C(x|z)$
3. The updated state is the new prior for the next prediction.

Propagation of Uncertainty

Up to now we have seen how to update uncertainty when we receive a new measure. Now we have to understand what happens to uncertainty when state evolves along time following model dynamics.

When covariance matrix "pass" through a function (of dynamic model), it come out deformed.

CASE 1: Linear Functions

Model $z = Hx + \varepsilon$ where state $x \sim \mathcal{N}(\mu_x, P)$; noise $\varepsilon \sim \mathcal{N}(0, R)$

what is the uncertainty of z?

$z \sim \mathcal{N}(\mu_z, C(z))$ where $\begin{cases} \mu_z = H\mu_x \\ C(z) = HPH^T + R \end{cases}$ $\Rightarrow$ Total uncertainty of measure z is the sum of

- HPH^T: uncertainty of the state (P), propagated through sensor (H).
- R: uncertainty introduced by sensor

Formula of $C(z)$ is used in the Prediction phase of Kalman Filter. When you predict the measure, this formula is used to compute covariance of innovation $(z - \mu_z)$, that is used to compute optimal gain K.

CASE 2: Non Linear Functions

Model $z = h(x, \varepsilon)$ where $h(\cdot)$ is a non-linear function

The solution is to linearize the function: approximate with Taylor expansio at I° order.

Instead of H (of linear case) we have Jacobian J.

$\mu_z \simeq h(\mu_x, \mu_\varepsilon)$

$C(z) \simeq J P J^T + J_\varepsilon R J_\varepsilon^T$ (If noise is additive we simplify to $JPJ^T + R$).

EXAMPLE unicycle

Equation of motion (non-linear) : $\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \Delta t \cdot v_k \cos\theta_k \\ \Delta t \, v_k \sin\theta_k \\ \Delta t \cdot \omega_k \end{bmatrix}$

An estimation filter need, at each prediction step, compute Jacobian of this function to propagate covariance matrix P from k to k+1.

From static Uncertainty to Stochastic Process

Error of a sensor is not a single random number, but a sequence of random numbers, one for each time istant.

The set of variables, that represents all the measur taken by a robot, is called STOCHASTIC PROCESS.

A stochastic process $X(t)$ is a family of time-indexed rand. variables.

MEAN and VARIANCE IN TIME:

- mean (μ_{t_1}): expected value of the variable at time t_1.
- variance ($\sigma^2_{t_1}$): variance at Time T_1.

RELATION IN TIME

- Covariance $C_x(t_i, t_j)$: measure covariance between rand.var. x at time t_1 and same var. at T_2.
- AUTOcorrelation $R(t_i, t_j)$: measure not centered on mean
  ↳ $C_x(t_i, t_j) = R(t_i, t_j) - \mu(t_i)(t_j)$

WIDE SENSE STATIONARY (WSS) PROCESSES

A process is wws if its statistical properties doesn't change in time.

1. mean = cost
2. AUTOcorrelation independent by delay => $R(t_i, t_j)$ doesn't depend by absolute time t_i and t_j, but depends on $\tau = t_i - t_j$

So, at the end, we describe the process in function of τ.

WHITE PROCESS

It's a wss process that is completly uncorrelated in Time.

AUTOcorrelation of a white process is Dirac delta → $R(\tau) = \sigma^2 \delta(\tau)$

↳ correlation $\neq 0 \iff \tau = 0$ The value of the process at a certain istant doesn't have any relation with its value in any other instant.

POWER SPECTRAL DENSITY: white process has same power at each frequency.

MARKOV CHAIN

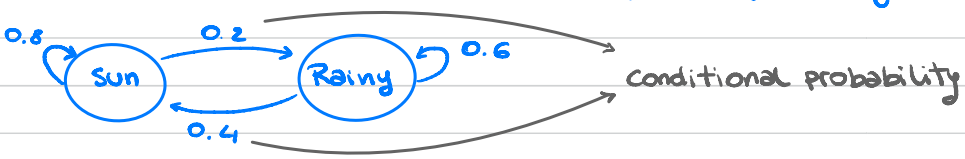
Markov processes are a special type of stocasthic processes.

DEF: in a Markov process the probability of future state x_{k+1} depends only on present state x_k, and not on the sequence of past states. $Pr(x_{k+1} | x_k, x_{k-1}, \dots x_0) = Pr(x_{k+1} | x_k)$

Markov chain says how evolves probability to be in a single state.

EXAMPLE

Suppose you want to know if tomorrow will be rainy or sunny knowing that today there is Sun.



$x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ Sunny $\Rightarrow$ Probability of today
Rainy

$Pr(x_1 = Sun)$ = Prob. that tomorrow will be sunny = 0.8

$Pr(x_1 = Rainy) = 0.2$

$Pr(x_1 = Sun) = Pr(x_2 = Sun \wedge x_0 = Sun) + Pr(x_1 = Sun \wedge x_0 = Rain)$ Total probability law

$\overset{!}{=} Pr(x_1 = Sun \mid x_0 = Sun) Pr(x_0 = Sun) + Pr(x_1 = Sun \mid x_0 = Rain) Pr(x_0 = Rain)$

$\overset{!}{=} Pr(x_1 = Sun \mid x_0 = Sun)$

$\overset{!}{=} [0.8; 0.4] \pi x_0$ where $\pi x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ probability of x_0

$Pr(x_1 = Rain) = [0.2, 0.6] \pi x_0$

$P = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix} \Rightarrow \pi x_1 = P \pi x_0 = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}$

With Markov chain we can make predictions, for example 10 days ahead.

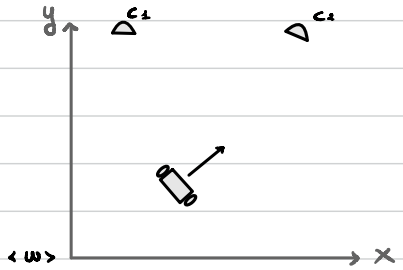
$\pi_{10} = \pi_9 P^T = \pi_8 P^T P^T = \underline{\pi_0 (P^T)^{10}}$

$\pi x_1 = P \pi x_0$ where P is a stocasthic matrix

Since usually πx_n are expressed as row vectors $\Rightarrow \pi x_1 = \overset{[1,0]}{\pi^T x_0} P^T$

ESTIMATION ALGORITHMS

Suppose we have an unicycle, and 3 cameras that measure the pos. and orient. of the unicycle with respect to a reference frame.



SYSTEM: unicycle, which state s_k at time k is its pose on the plane $s_k = [x_k, y_k, \theta_k]^T$

SENSOR: 3 cameras (c_1, c_2, c_3), each of them give a measure of complete state of the robot.

MEASURE MODEL (for each camera): $z_i = H s_k + \varepsilon_i$ where $H = I_3$ because we assume that each camera measure directly pose $[x, y, \theta]$.

$\varepsilon_i \sim N(0, R)$ → each camera has a measure error, that is a white Gaussian process.

COMBINATION OF MEASURES: treat measures from sensors like a one big measure coming from a virtual sensor.

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \qquad H = \begin{bmatrix} H \\ H \\ H \end{bmatrix} = \begin{bmatrix} I_3 \\ I_3 \\ I_3 \end{bmatrix} \qquad \eta = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

Noise η has its covariance matrix R. Errors between sensors are independent, so covariance between ε_i and ε_j is zero.

$$R_{tot} = \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{bmatrix}$$

PROBLEM: $Z = H_{tot} s_k + \eta$ → If sensors would have same uncertainty ($R_1 = R_2 = R_3$), we can use pseudoinverse $\hat{s}_k = (H^T H)^{-1} H^T Z$. But, since sensors have different uncertainty, we have to weight more measure coming from sensors with less uncertainty (and viceversa).

So we use Weighted Least Squares (WLS) → $\hat{s}_k = ({}^k H^T {}^k R^{-1} {}^k H)^{-1} {}^k H^T {}^k R^{-1} Z = L_k$ Estimator

↳ Minimum variance unbiased estimator ($E(\hat{s}_k) = s_k$)

↳ R^{-1}: weight matrix (↓unc. ↑weight)

$\hat{s}_k$ is the estimation that minimize quadratic error.

We have 3 weighted averages, one for each variable (x, y, θ)

To know the precision of L_k, we have to compute covariance matrix of estimation error $\tilde{s}_k = \hat{s}_k - s_k$ ↳ Equal To The variance

$E((\hat{s}_k - s_k)(\hat{s}_k - s_k)^T) =$ (since L_k is unbiased)

$= E((\hat{s}_k - E(\hat{s}_k))(\hat{s}_k - E(\hat{s}_k))) =$

$\overset{!}{=} E(L_k \eta_k \eta_k^T L_k^T)$ ⇒ Estimation error depends only on η_k

$\overset{!}{=} L_k {}^k R L_k^T$

$\overset{!}{=} ({}^k H^T {}^k R^{-1} {}^k H)^{-1} = P_k$ = covariance matrix of estimation error $= \mathcal{I}_k^{-1}$ Fisher Information Matrix

We can rewrite $\hat{s}_k = P_k {}^k H^T {}^k R^{-1} {}^k Z$ → M.V.U.E. if unc. gaussian; B.L.U.E. if unc. not gaussian } Function linear

EXAMPLE 1

if $\eta_k \sim \mathcal{N}(0, R)$; $\eta \in \mathbb{R}^n$

if R is positive semi definite $\Rightarrow \dim(\ker(R)) \geq 1$

↳ $\eta_k = G\varepsilon_k$ where $G \in \mathbb{R}^{n \times m}$ $m < n$

$\varepsilon_k \sim \mathcal{N}(0, Q)$ $Q \in \mathbb{R}^{m \times m}$ positive definite

EXAMPLE 2

$z_k = \mathbb{1} x + \varepsilon$; $\mathbb{1} \in \mathbb{R}^{n \times 1}$; $\varepsilon \sim \mathcal{N}(0, R)$ $R = \sigma^2 I_n$ Same uncertainty for variables

Estimate $\hat{x}$

$P_x = ({}^{w}H^T \, {}^{k}R^{-1} \, {}^{k}H)^{-1} = (\mathbb{1}^T 1/\sigma^2 \, I_n \, \mathbb{1})^{-1} = \frac{\sigma^2}{n}$ variance when we compute the average

↑ CONSISTENCY : $\lim_{n \to \infty} P_r(|\hat{x} - x| \geq a) = 0$, $\forall a > 0$

↳ WLS are consistent

$\hat{x} = P_x \, {}^{k}H^T \, {}^{k}R^{-1} \, {}^{k}z = \frac{\sigma^2}{n} \mathbb{1}^T \frac{1}{\sigma^2} I_n z_x = \frac{1}{n} \sum_i^n z_i$ IF you have the same unc. for all variables, optimal estimation is the mean.

EXAMPLE 3

$z_x = \mathbb{1} x + \varepsilon$; $\mathbb{1} \in \mathbb{R}^{n \times 1}$; $\varepsilon \sim \mathcal{N}(0, R)$ $R = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{bmatrix}$ Different uncertainty for variables

Estimate $\hat{x}$

$P_x = (\mathbb{1}^T \begin{bmatrix} 1/\sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_n^2 \end{bmatrix} \mathbb{1})^{-1} = \frac{1}{\sum_i^n \frac{1}{\sigma_i^2}}$

$\hat{x} = P_x \mathbb{1}^T R^{-1} z_x = \frac{\sum_i^n z_i/\sigma_i^2}{\sum_i^n 1/\sigma^2}$ Weighted Average

→ Weight $\in [0,1]$ → How much information you bring to your estimate

↳ Estimation is always between min and max of measurements taken

↓

You sum all positive values → every measure increase information (even with ↑↑ unc.)

EXAMPLE 4

$\left.\begin{aligned} z_1 &= x + \varepsilon_1 \\ z_2 &= x + \varepsilon_2 \end{aligned}\right\}$ $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$; $c(\varepsilon_1, \varepsilon_2) = \rho \sigma_1 \sigma_2$ (there is a relation between ε_1 and ε_2)

$\Downarrow$

$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathbb{1} x + \varepsilon$; $\varepsilon \sim \mathcal{N}(0, R)$ $\longrightarrow$ $R = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$

$\hat{x} = (\mathbb{1}^T R^{-1} \mathbb{1})^{-1} \mathbb{1}^T R^{-1} z$; $R^{-1} = \frac{1}{\sigma_1^2 \sigma_2^2 (1-\rho^2)} \begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix} = \frac{1}{(1-\rho^2)} \begin{bmatrix} 1/\sigma_1^2 & -\rho/\sigma_1\sigma_2 \\ -\rho/\sigma_1\sigma_2 & 1/\sigma_2^2 \end{bmatrix}$

$P_x = (\mathbb{1}^T R^{-1} \mathbb{1})^{-1} = \frac{(1-\rho^2)}{1/\sigma_1^2 + 1/\sigma_2^2 - 2\rho/\sigma_1\sigma_2}$

If $\rho = \pm 1 \Rightarrow \varepsilon_2 = a\varepsilon_1$ with $a \in \mathbb{R}$ ⇒ It means that if they are correlated to each other, second measure doesn't give new information respect to the first measure.

Assume that $\rho = 0 \rightarrow P_x = \frac{1}{1/\sigma_1^2 + 1/\sigma_2^2} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$ conditional variance

$\hat{x} = P_x \mathbb{1}^T R^{-1} z = \frac{\sigma_1^2 z_1 + \sigma_2^2 z_2}{\sigma_1^2 + \sigma_2}$

! You HAVE TO take care of correlation between 2 measurements, otherwise you will over positive estimate the measure

A UNIFIED FRAMEWORK FOR ESTIMATION ALGORITHMS

1. LINEAR ESTIMATORS → apply to systems that can be described by linear equations

- Least Square (LS) Solution

  Tool to solve static estimation problems where we have more measurements than unknown variables.

  Model: $z = Hx + \varepsilon$ $\quad z \in \mathbb{R}^m$ measurements, $x \in \mathbb{R}^n$ state we want to estimate

  Goal: $\hat{x} = \arg\min_x \|z - Hx\|^2$

  Ordinary Least Square (OLS) solution

  ↳ More real scenario, in which uncertainties are different but correlated. We capture this with the MEASUREMENT NOISE COVARIANCE MATRIX $R = E(\varepsilon\varepsilon^T)$

  To account for this, we use WLS method: $\hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} z$ → R^{-1} acts like weight

  ↳ This is the OPTIMAL way to fuse data from multiple sensors with different levels of quality.

- KALMAN FILTER → tool for dynamic linear systems

  Recursive algorithm that provides the optimal state estimate by elegantly combining a system's dynamic model with a series of noisy measurements. It is real-time, dynamic implementation of the Bayesian estimation principles.

  KF requires 2 linear models:

  - PROCESS MODEL: $x_k = Ax_{k-1} + Bu_{k-1} + \underline{v_{k-1}}$ ← process noise
  - MEASUREMENT MODEL: $z_k = Hx_k + \underline{w_k}$ ← measurement noise

  KF operates in a RECURSIVE CYCLE:

  STEP 1: Prediction →
  $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$ state estimate
  $P_k^- = AP_{k-1}A^T + Q$ error covariance (uncertainty propagation)

  STEP 2: Correction →
  $K_k = P_k^- H^T (\underbrace{HP_k^- H^T + R}_{\text{tot. uncertainty of innovation}})^{-1}$ Kalman Gain, weight to balance pred. and meas.

  $\hat{x}_k = \hat{x}_k^- + K_k(\overbrace{z_k - H\hat{x}_k^-}^{\text{Innovation}})$ Innovation weighted by the Kalman Gain

  (↓ Prediction)

  $P_k = (I - K_k H)P_k^-$ update error covariance, reduce unc. after meas.

2. NONLINEAR ESTIMATORS → most real-robots systems

- Non Linear Least Squares (NLS) solution

  MODEL: $z = h(x) + \varepsilon$

  SOLUTION: iterative approach, called LINEARIZATION

  1. Initial guess x_0
  2. $h(x) \approx h(\hat{x}_i) + J_i(x - \hat{x}_i)$ approximation with Taylor around $\hat{x}_i$.
  3. $z - h(\hat{x}_i) \approx J_i \Delta x_i$ where $\Delta x_i = x_i - \hat{x}_i$ correction step
  4. $\Delta x_i = (J_i^T J_i)^{-1} J_i^T (z - h(\hat{x}_i))$ LS formula to solve for x_i
  5. $\hat{x}_{i+1} = \hat{x}_i + \Delta x_i$ update estimate
  6. Repeat until converges

• Extended Kalman Filter (EKF)

IDEA: linearize the non linear models at each time step around the current best estimate, then apply standard KF equations.

MODEL: $x_k = f(x_{k-1}, u_{k-1}) + v_{k-1}$

$z_k = h(x_k) + \omega_k$

EKF CYCLE:

(1) Prediction $\begin{cases} \hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}) & \text{state estimate} \\ P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q & \text{error covariance, where } F = \text{Jacobian of } f \end{cases}$

(2) correction → kalman Gain $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$ where H = Jacobian of h

→ update state estimate $\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(x_k^-))$

→ update error covariance $P_k = (I - K_k H_k) P_k^-$

Not R.V. → Non-Bayesian estimator	R.V. → Bayesian estimator
WLS	MAP
ML	MMSE

M.A.P. $\max_{\hat{x}} P(\hat{x}|z) = \max_{\hat{x}} P(z|x)P(x)$

$$\begin{cases} \mu_{x|z} = \dfrac{\theta_\varepsilon^2}{\theta^2+\theta_\varepsilon^2}\mu_x + \dfrac{\theta^2}{\theta^2+\theta_\varepsilon^2} z \\ \theta^2_{x|z} = \dfrac{\theta^2\theta_\varepsilon^2}{\theta^2+\theta_\varepsilon^2} \end{cases}$$

- good prior $\Rightarrow \theta^2 << \theta_\varepsilon^2 \Rightarrow$ MAP $\simeq \mu_x$
- Bad prior $\Rightarrow \theta^2 >> \theta_\varepsilon^2 \Rightarrow$ MAP $\simeq z$

MAP $\dfrac{\sigma_\varepsilon^2}{\bar{\theta}^2+\theta_\varepsilon^2}\bar{\mu}_x + \dfrac{\bar{\theta}^2}{\bar{\theta}^2+\theta_\varepsilon^2} z$ is UNBIASED?

- Yes if $\bar{\mu}_x = \mu_x$ because $E(\hat{x}) = \mu_x = E(x)$
- No if $\bar{\mu}_x \neq \mu_x \Rightarrow$ wrong choice of the prior
  - the solution is to take more measurements and combine the average of measurements with prior

MMSE $E\left((x-\hat{x}(z))^2\right) \xrightarrow{\text{R.V.}} \iint (x-\hat{x}(z))^2 P(z,x)\, dz\, dx = \ldots = \min_{\hat{x}} E\left((x-\hat{x}|z)^2\right)$

KALMAN FILTER

Discrete time model (Actual system)
$$\begin{cases} x_{k+1} = A_D x_k + B_{Dk} u_k \\ z_k = H x_k + v_k \end{cases} \quad \text{where uncertainty } v_k \sim N(0, R_k)$$

System Model (Estimated system)
$$\begin{cases} \bar{x}_{k+1} = A_{Dk}\, \bar{x}_k + B_{Dk}\, \bar{u}_k \\ \bar{z}_{k+1} = H_k\, \bar{x}_k \end{cases} \quad \text{where actual input } \bar{u}_k = u_k + \varepsilon_k \quad \varepsilon_k \sim N(0, Q_k)$$

If model is unbiased $\Rightarrow E(\bar{x}_{k+1}) = x_{k+1} \Rightarrow$ we can propagate mean and variance of system

$\hat{x}_{k+1} = E(\bar{x}_{k+1}) = E(A_{Dk}\bar{x}_k + B_{Dk}\bar{u}_k) = A_{Dk}\hat{x}_k + B_{Dk}u_k$ since they're deterministic I can put the u outside expected operator

Uncertainty of $\hat{x}_{k+1}$ is, by definition, the covariance matrix: $E\left((\hat{x}_{k+1} - E(\hat{x}_{k+1}))(\hat{x}_{k+1} - E(\hat{x}_{k+1}))^T\right)$

$= E(\tilde{x}_{k+1}\tilde{x}_{k+1}^T)$ with $\tilde{x}_{k+1}$ estimation error

$= E((A_{Dk}\tilde{x}_k + B_{Dk}\varepsilon_k)(A_{Dk}\tilde{x}_k + B_{Dk}\varepsilon_k)^T)$

$\tilde{x}_k$ and ε_k uncorrelated → ε_k is white (Gaussian with zero mean)

$= A_{Dk}\underbrace{E(\tilde{x}_k\tilde{x}_k^T)}_{P_k}A_{Dk}^T + B_{Dk}\underbrace{E(\varepsilon_k\varepsilon_k^T)}_{Q_k}B_{Dk}^T$

$\Rightarrow P_{k+1} = A_{Dk}P_kA_{Dk}^T + B_{Dk}Q_kB_{Dk}^T$ If A_{Dk} is not asintotically stable (like a vehicle), covariance of estimation error (P_{k+1}) NEVER decrease. It's always zero or positive. At every time step uncertainty grows.

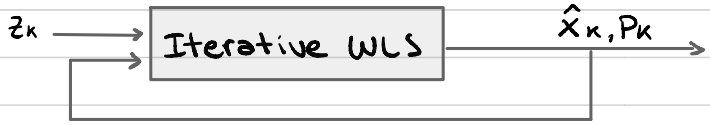
This is how we propagate mean ($\hat{x}_{k+1}$) and covariance (P_{k+1}). This is just model-base, we don't use measures z. This is how works propioceptive sensors. control

P.D.F. $p(x_{k+1} \mid x_k, u_k)$ → What we have computed up to now. An inference considering what was happened before.

WHAT HAPPENS WHEN WE USE MEASUREMENTS?

Iterative solution of WLS assumes the knowledge of $P_k, \hat{x}_k$.

- When z_{k+1} arrives, we compute innovation of the measurements $S_{k+1} = H_{k+1}P_kH_{k+1}^T + R_{k+1}$ (measurement uncertainty)
- Gain is computed as $W_{k+1} = P_kH_{k+1}^TS_{k+1}^{-1}$ → Fisher Information Matrix
- Estimate is computed as $\tilde{x}_{k+1} = \tilde{x}_k + W_{k+1}(z_{k+1} - H_{k+1}\hat{x}_k)$
- $P_{k+1} = (I - W_{k+1}H_{k+1})P_{k+1}$



Deterministic

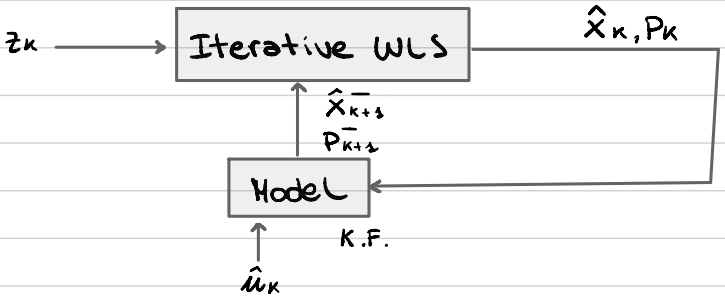
Rely just on measurements

WHAT CHANGES WITH KALMAN FILTER?

Make prediction about next step using the knowledge of the model.

Kalman filter is OPTIMAL.

Optimality holds if model is linear, uncertainties are gaussian and white.



Kalman Filter iterative approach

K.F. (Discrete)

$$\begin{cases} \hat{x}_{k+1}^{-} = A_{Dk}\hat{x}_k + B_{Dk}\bar{u}_k & \text{(Mean of the Gaussian} \rightarrow \text{M.A.P.)} \\ \hat{z}_k = H_k\hat{x}_k & \varepsilon_k \sim \mathcal{N}(0, Q_k) \end{cases}$$

$$P_{k+1} = A_{Dk}P_kA_{Dk}^T + B_{Dk}Q_kB_{Dk}^T$$

$\hat{x}_{k+1}^{-}$, P_{k+1} } Prediction ‖ Bootstrap — $\hat{x}_0$ initial knowledge, P_0 initial uncertainty } PRIOR

$$S_{k+1} = H_{k+1}P_{k+1}^{-}H_{k+1} + R_k$$
$$W_{k+1} = P_{k+1}^{-}H_{k+1}^{-}S_{k+1}^{-1}$$
$$\hat{x}_{k+1} = \hat{x}_{k+1}^{-} + W_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1}^{-})$$
$$P_{k+1} = (I - W_{k+1}H_{k+1})P_{k+1}^{-}$$

} UPDATE = Likelihood + Prior = POSTERIOR
↳ given by measurement

Iterate

Assumptions:

1) ε_k white, zero mean
2) ν_k " "
3) $E(\varepsilon_k \nu_k) = 0$

] B.L.U.E.

$\Rightarrow$ If ε_k, ν_k are Gaussian $\Rightarrow$ MMSE

If 1) and 2) don't hold, there are other solutions that are also optimal, adding more components.

SIMPLE DISTRIBUTED ESTIMATION

EXAMPLE

x_1 and x_2 move on a line. Dynamics of each point i is described by $x_i(k+1) = x_i(k) + a_i d_i(k) + b_i \varepsilon_i(k)$.
$d_i(k)$ is a know movement, $\varepsilon_i(k) \sim \mathcal{N}(0, \sigma_i^2)$ is a Gaussian white noise. To estimate point position, we suppose that it receives, at regular intervals n_i, an absolute measure of its position (e.g. GPS), described by $z_i(n_i k) = x_i(n_i k) + \eta_i(n_i k)$ where $\eta_i(n_i k) \sim \mathcal{N}(0, \xi_i^2)$ it's a white Gaussian noise.

To manage the estimation, we use an INTERMITTENT KALMAN FILTER.

- Prediction update $\hat{x}_i$ and P_i

$$\left.\begin{aligned} \hat{x}_i(k+1)^- &= \hat{x}_i(k) + a_i d_i(k) \\ P_i(k+1)^- &= P_i(k) + b_i^2 \sigma_i^2 \end{aligned}\right\} \text{Priori}$$

- Update step occurs every n_i

$$\left.\begin{aligned} P_i(k+1) &= \xi_i^2 W \\ \hat{x}_i(k+1) &= \hat{x}_{(k+1)}^- + W\left(z_i(k+1) - \hat{x}_i(k+1)^-\right) \end{aligned}\right\} W = \frac{P_i(k+1)^-}{P_i(k+1)^- + \xi_i^2} = \text{Kalman Gain}$$

COLLABORATIVE LOCALIZATION

At this point we add relative measures. It is assumed that agent i, every m_i steps, measures its position relative to agent j using sensors like LiDAR and camera.

Model $\Delta_{ij}(m_i k) = (x_j(m_j k) - x_i(m_i k)) + \eta_{ij}(m_i k)$ with $\eta_{ij} \sim \mathcal{N}(0, \xi_{ij}^2)$ Gaussian noise

Agent j must communicate its estimate $\hat{x}_j$ and covariance P_j to agent i. Total uncertainty become $\xi_{ij}^2 + P_j(m_i k)^-$, which includes agent j's estimate.

When agent i updates using only the relative measurement, the innovation covariance becomes

$$S_i(k+1) = P_i(k+1)^- + \xi_{ij}^2 + P_j(k+1)^-$$

If agent i receives both the absolute measurement z_i and the relative measurement z_{ij} at the same instant, the filter handles a multivariate update.

$$\bar{z}(k+1) = \left[z_i(k+1),\ z_{ij}(k+1)\right]^T \quad \text{measure vector}$$

$$R_i(k+1) = \operatorname{diag}\left(\left[\xi_i^2,\ \xi_{ij}^2 + P_j(m_i k)^-\right]\right) \quad \text{covariance matrix}$$

CONSISTENCY AND DATA INCEST IN DISTRIBUTED ESTIMATION

When moving to Distributed Estimation with relative measurements, a critical failure mode arises called inconsistency.

PROBLEM → If robot i update its estimate using robot j's estimate, and robot j later uses robot i's estimate they begin to recycle the same information. In a centralized analysis, this manifests as non-zero off-diagonal blocks in the covariance matrix P, meaning the estimation errors of the agents become correlated.

CONSEQUENCE → If you treat these correlated estimates as independent (naive approach), the filter becomes inconsistent: the calculated covariance P becomes artificially small, while the real error diverges. Specifically, the "sum" of the positions $(x_i + x_j)$ becomes an unobservable subspace subject to dead reckoning (unbounded error growth).

LINEAR CONSENSUS

The network topology in a distributed system is defined using Graph Theory.

Network → Graph $G(N, \mathcal{E})$ N: nodes $\mathcal{E}$: communication edges

ADJACENCY MATRIX (A): encodes the connectivity ($a_{ij} = 1$ if connected, 0 otherwise).

DEGREE MATRIX (D): a diagonal matrix where each element is the degree (number of neighbors) of the node.

LAPLACIAN MATRIX (L): Defined as $L = D - A$. This is the crucial operator for continuous-time consensus.

Key properties: the Laplacian matrix is always positive semidefinite. Importantly, for the system to reach a consensus, the graph must be "rooted". If the graph is undirected, the Laplacian is symmetric, and the algebric connectivity determines the speed of convergence.

LINEAR CONSENSUS and MATRIX DESIGN

For discrete-time systems, the consensus protocol is governed by the update law $x(k+1) = Qx(k)$, where Q is the transition matrix. The convergence depends entirely on the properties of Q:

- Stochastic Matrix: if Q is row-stochastic (rows sum to 1), the system reaches a consensus on some value, but not necessarily the average.
- Doubly Stochastic Matrix: to guarantee Average Consensus (converging to the exact mean of initial states), Q must be doubly stochastic.

↓

Q must be designed locally. Common methods include Max-Degree weights or Metropolis-Hasting weights, which ensure Q is doubly stochastic for undirected graphs, thereby guaranteeing convergence to the average.