Coupling GPS + vehicle sensors

- For continuity and accuracy purpose
- Strength / Weakness
- GPS: constant accuracy / unavailable indoor and in deep unban environment
- Odometre (wheel speed sensor) and Gyrometre (vehicle turn rate): always work / drift with time

=> **Combination of the strengths of both** GPS and vehicle sensors: continuous positioning system and better accuracy

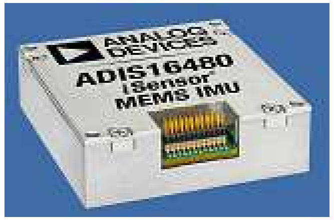
Proprioceptive sensors

- Odometre (wheel encoder, wheel speed sensor…): n pulses returned per revolution. Calibration needed.

- Both positive/negative counting (quadrature)
- CAN bus available data, for each wheel

Proprioceptive sensors

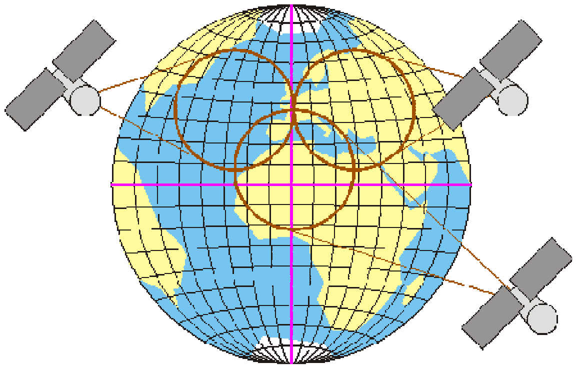
- Gyrometre: turn rate (deg or rad/s)
- Fiber-optic and Micro-Electro-Mechanics systems are the most popular



- IMU: Inertial Meas Units include 3 axis gyro and accelerometers

Exteroceptive sensors

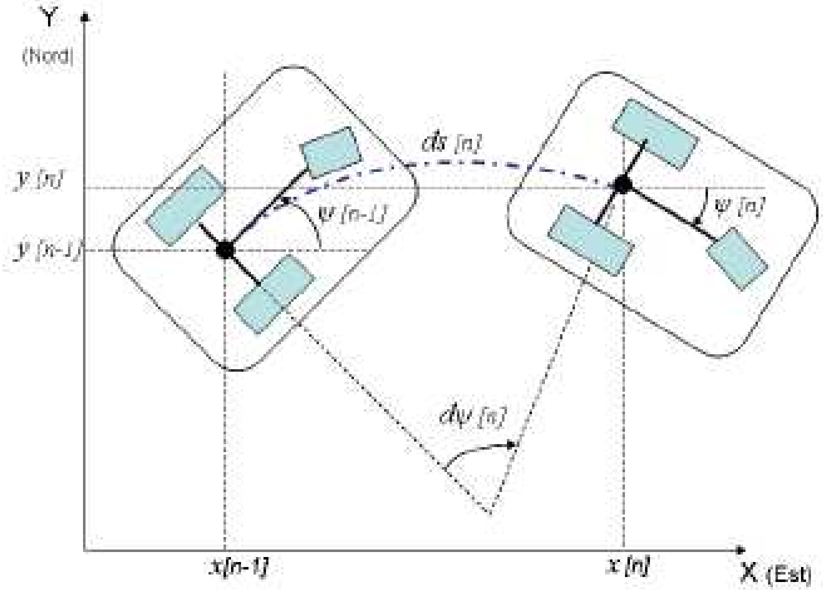
- GNSS
- Camera vision, Lidar: make use of landmarks (that must be detected, recognized and georeferenced)



- SLAM: Simultaneous Location And Mapping

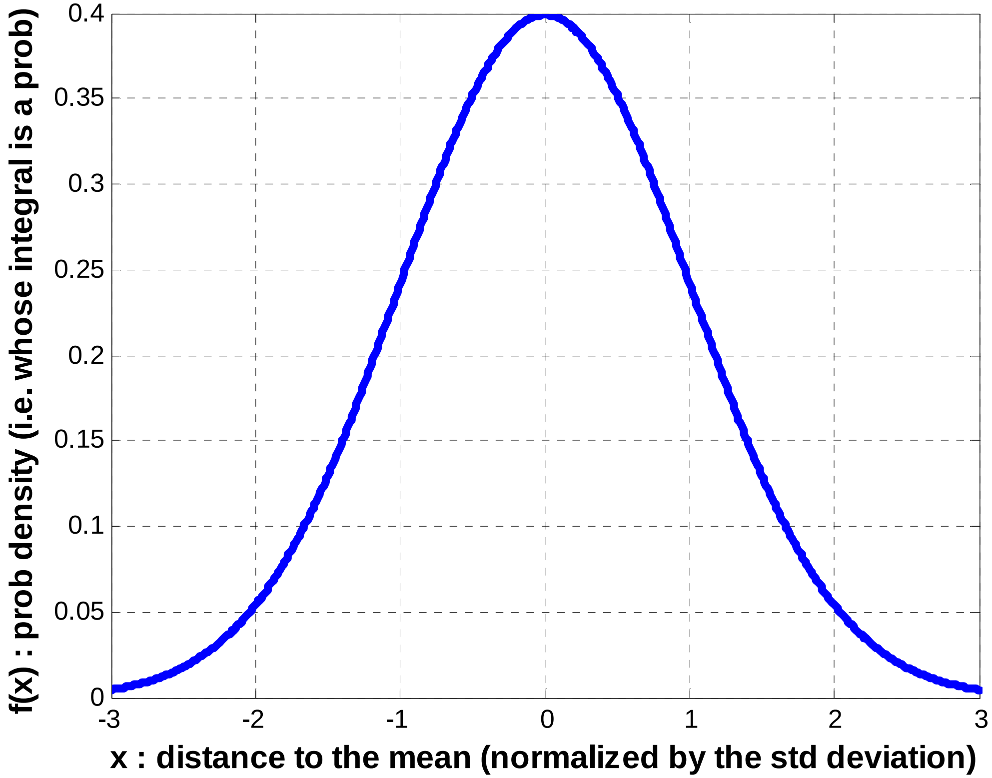
Motion model

- The rover motion is not any!
- Predictive equations, e.g.:
  - double integration of acceleration (planes, submarines…)
  - non-holonomic vehicle constraints (if rolling without slipping)
  - pedestrian stride model



Bayesian fusion

- Normal distributed random variable
- Its pdf: $f(x)=1/\sigma\sqrt{2\pi}\, \exp(-(x-\mu)^2/2\sigma^2)$
- μ: mean and σ: standard deviation (σ^2: variance)



Bayesian fusion

- Fusion of normally distributed random variables
- These are caracterised by their mean and their variance: μ_1, $\sigma^2{}_1$ and μ_2, $\sigma^2{}_2$ (scalar here)
- Which « estimate » maximizes the prob of both « measurements » 1 and 2 to be observed?
- i.e. x? $\max[\exp(-(x-\mu_1)^2/2\sigma^2{}_1 * \exp(-(x-\mu_2)^2/2\sigma^2{}_2)]$
- ⇔ x? $\max\log[\exp(-(x-\mu_1)^2/2\sigma^2{}_1 * \exp(-(x-\mu_2)^2/2\sigma^2{}_2)]$
- ⇔ x? $\max[(-(x-\mu_1)^2/2\sigma^2{}_1 + (-(x-\mu_2)^2/2\sigma^2{}_2)]$
- Derivate: $[(-(x-\mu_1)/\sigma^2{}_1 + (-(x-\mu_2)/\sigma^2{}_2)]$
  Null for: $x = (\mu_1/\sigma^2{}_1 + \mu_2/\sigma^2{}_2) / (1/\sigma^2{}_1 + 1/\sigma^2{}_2)$
- **in Bayesian fusion this estimate is called MAP max a posteriori: the variance-weighted mean**
- its variance is given by: $1 / (1/\sigma^2{}_1 + 1/\sigma^2{}_2)$

Bayesian fusion and Kalman filter

- Fusion of normally distributed random variables
- These are caracterised by their mean and their variance: X1, P1 and X2, P2 (in dimension N)
- Distance of Mahalanobis (or NIS: normalized innovation squared):
- NIS = $\sqrt{(X1-X2)' * inv(P1+P2) * (X1-X2)}$
- if NIS > chi² threshold N dof, incoherent variables
- otherwise, variables can be fused computing:
- Gain: K = P1 * inv(P1+P2)
  Mean: X = X1 + K * (X2-X1)
  Variance: P = P1 – K * P1
- **in a Kalman filter, X1: predicted meas. h(X) and X2: actual meas. Y**

One sets the risk of false alarm (in %)

Hidden Markov chain

- The whole past information is contained in the courant state of the system
  - State vectors (X) are chained (by an evolution model or **prediction model**) with or w/o input vectors (U)
- Hidden state
  - State vectors (X) are not directly observed by meas (Y) but thru an observation model or **estimation model**
- If the models have normal errors and are linear, the Kalman filter gives explicitely the optimal solution of the problem: this filter is also called **predictor/estimator**

Extended Kalman filter

- Generalisation when the models are not linear
- => linearization around the current state (ordre 1 Taylor development)
- Jacobian matrix: partial derivate of the equations of evolution et the equations of observation wrt the state vector (and eventually wrt to the input vector)
- No more optimal
- No guaranteed unicity: one may (in case of severe non linearity) converge to a local minima/maxima

EKF equations

- State system

$$X_k = f_k(X_{k-1}, U_k, V_k)$$
$$Y_k = h_k(X_k, W_k)$$

with $X = (x_1, x_2, ..., x_n)^T$

- Bayesian rules

- Gaussian assumptions

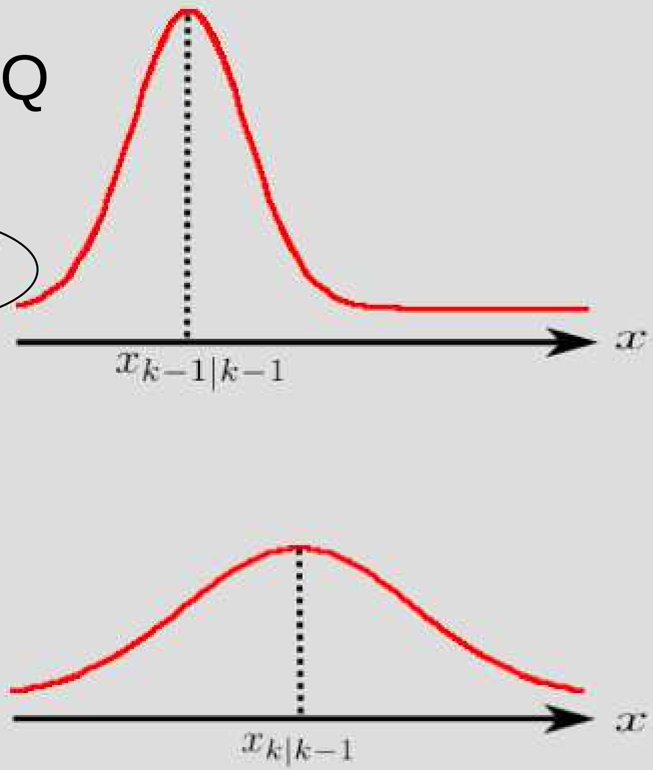
$$p(X_{k-1}|Y_{1:k-1}) \approx \mathcal{N}(\bar{X}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$$
$$p(X_k|Y_{1:k-1}) \approx \mathcal{N}(\bar{X}_{k|k-1}, \mathbf{P}_{k|k-1})$$
$$p(X_k|Y_{1:k}) \approx \mathcal{N}(\bar{X}_{k|k}, \mathbf{P}_{k|k})$$

  – with $P = cov(x_1, x_2, ..., x_n)$

EKF equations

- Prediction step

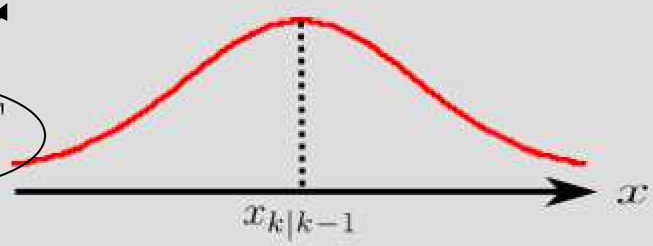
$$p(X_k|Y_{1:k-1}) \approx \mathcal{N}(\bar{X}_{k|k-1}, P_{k|k-1})$$
$$\bar{X}_{k|k-1} = f_k(\bar{X}_{k-1|k-1}, U_k, 0)$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k P_{k-1|k-1}\mathbf{F}_k^T + \mathbf{\Theta}_k \Sigma_U \mathbf{\Theta}_k^T + \underbrace{\mathbf{\Psi}_k \Sigma_V \mathbf{\Psi}_k^T}_{Q}$$



EKF equations

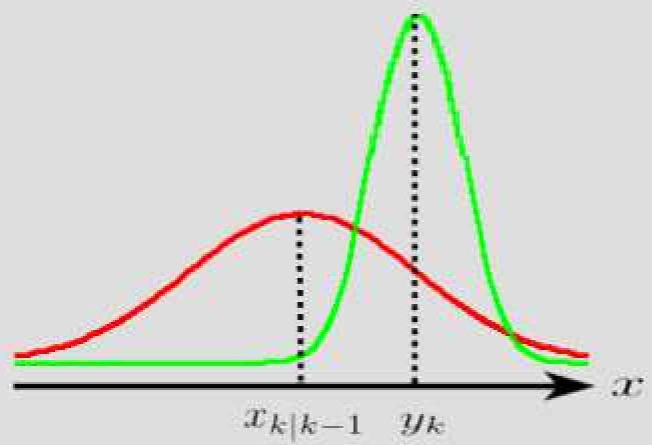
- Prediction step

$$p(X_k|Y_{1:k-1}) \approx \mathcal{N}(\bar{X}_{k|k-1}, P_{k|k-1})$$
$$\bar{X}_{k|k-1} = f_k(\bar{X}_{k-1|k-1}, U_k, 0)$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k P_{k-1|k-1} \mathbf{F}_k^T + \mathbf{\Theta}_k \Sigma_U \mathbf{\Theta}_k^T + \underbrace{\mathbf{\Psi}_k \Sigma_V \mathbf{\Psi}_k^T}_{Q}$$



- Correction step

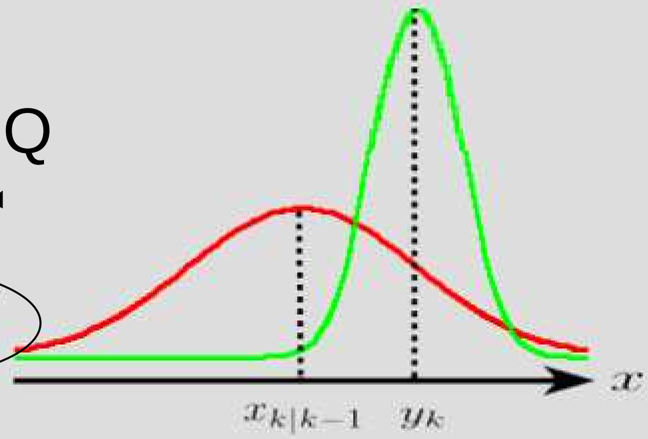
$$p(X_k|Y_{1:k}) \approx \mathcal{N}(\bar{X}_{k|k}, \mathbf{P}_{k|k})$$
$$\bar{X}_{k|k} = \bar{X}_{k|k-1} + \mathbf{K}_k(Y_k - h_k(\bar{X}_{k|k-1}, 0))$$
$$P_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) P_{k|k-1}$$
$$\mathbf{K}_k = P_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k P_{k|k-1} \mathbf{H}_k^T + \underbrace{\mathbf{\Upsilon}_k \Sigma_W \mathbf{\Upsilon}_k^T}_{R})^{-1}$$



EKF equations

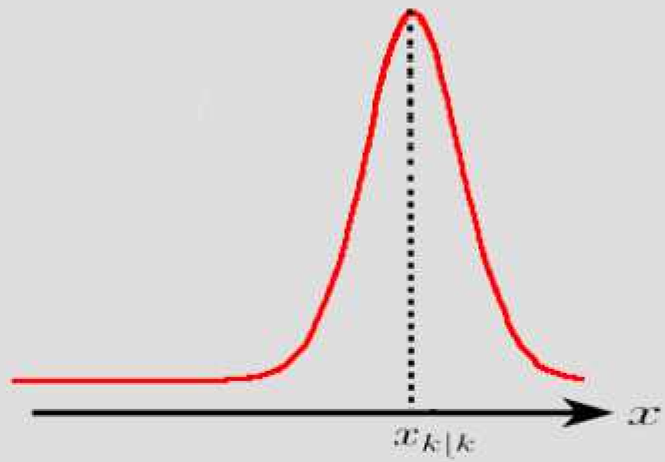
- Prediction step

$$p(X_k|Y_{1:k-1}) \approx \mathcal{N}(\bar{X}_{k|k-1}, P_{k|k-1})$$
$$\bar{X}_{k|k-1} = f_k(\bar{X}_{k-1|k-1}, U_k, 0)$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k P_{k-1|k-1} \mathbf{F}_k^T + \mathbf{\Theta}_k \Sigma_U \mathbf{\Theta}_k^T + \underbrace{\mathbf{\Psi}_k \Sigma_V \mathbf{\Psi}_k^T}_{Q}$$

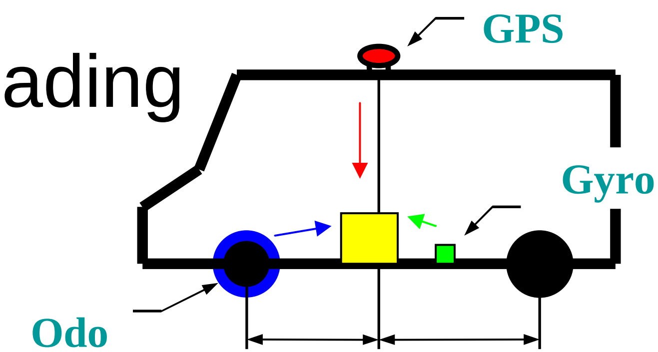


- Correction step

$$p(X_k|Y_{1:k}) \approx \mathcal{N}(\bar{X}_{k|k}, \mathbf{P}_{k|k})$$
$$\bar{X}_{k|k} = \bar{X}_{k|k-1} + \mathbf{K}_k(Y_k - h_k(\bar{X}_{k|k-1}, 0))$$
$$P_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) P_{k|k-1}$$
$$\mathbf{K}_k = P_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k P_{k|k-1} \mathbf{H}_k^T + \underbrace{\mathbf{\Upsilon}_k \Sigma_W \mathbf{\Upsilon}_k^T}_{R})^{-1}$$



Loose coupling for vehicle location



- State: East North (E N) and heading
- Input: Δdistance odo and gyro turn rate
- Evolution of the state:

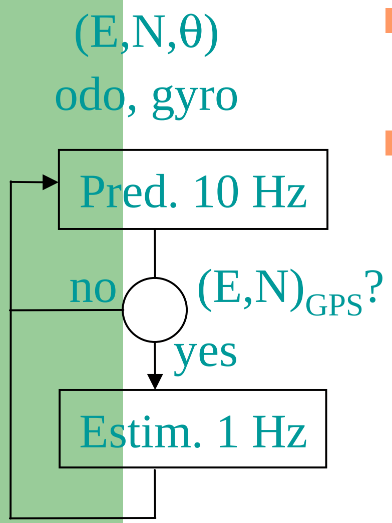
$$E(k+1) = E(k) + \Delta\text{distance odo} * \cos(\text{heading})$$

$$N(k+1) = N(k) + \Delta\text{distance odo} * \sin(\text{heading})$$

$$\text{heading}(k+1) = \text{heading}(k) + \text{gyro turn rate} * \Delta t$$

- GPS observation: lat, lon, alt (E and N in projection)
- Gaussian noises (GPS, odo, gyro)

Loose coupling EKF implementation



- State: east(E), north(N), heading(θ)
- Prediction (at e.g. 10 Hz):
  evolution model of the vehicle using performed distance (odo) and turn rate (gyro)
- Estimation (at e.g. 1 Hz):
  observation model of E and N directly with GPS co-ordinates

Loose coupling for vehicle location

- Jacobian matrixes: F (k):

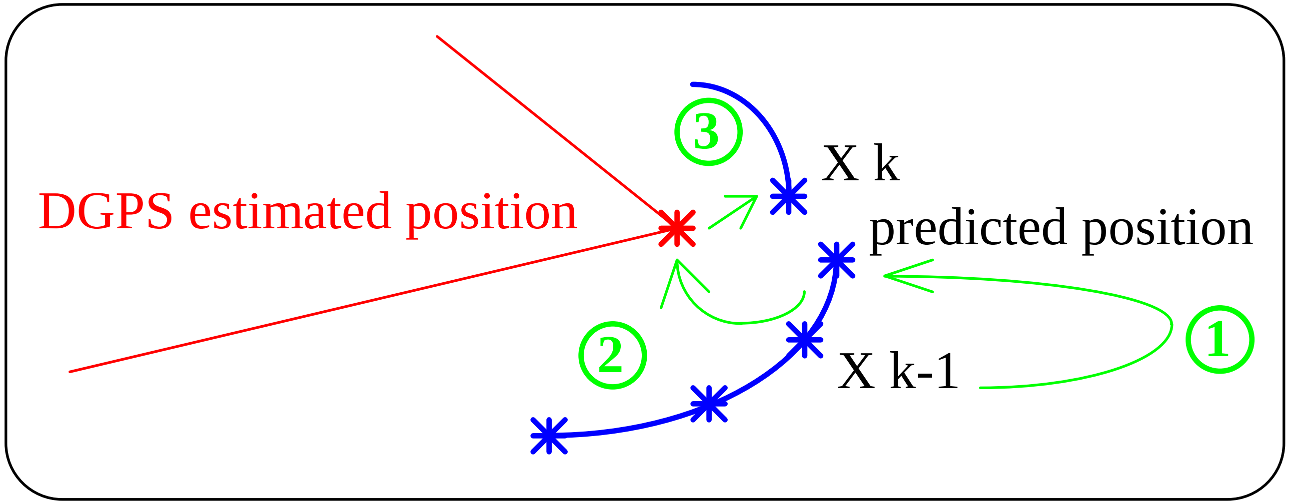
$$\begin{bmatrix} 1 & 0 & -\Delta d * \sin(\text{heading}) \\ 0 & 1 & \Delta d * \cos(\text{heading}) \\ 0 & 0 & 1 \end{bmatrix}$$

and Θ (k):

$$\begin{bmatrix} \cos(\text{heading}) & 0 \\ \sin(\text{heading}) & 0 \\ 0 & \Delta\text{temps} \end{bmatrix}$$

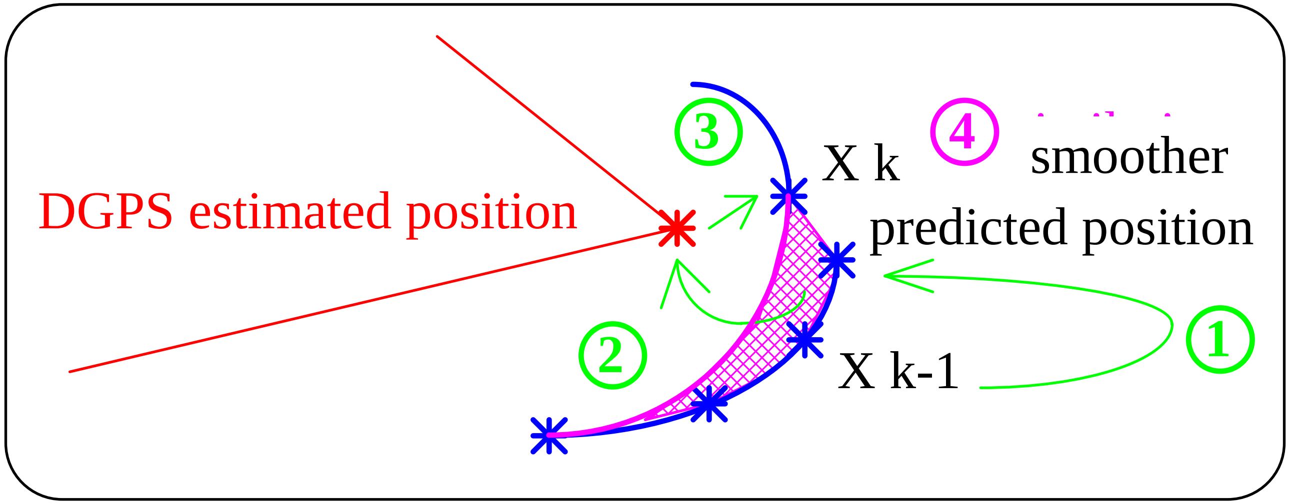
- H (k) linear: [1 0 0 ; 0 1 0] (and state not hidden)

Real-time filtering of vehicle position



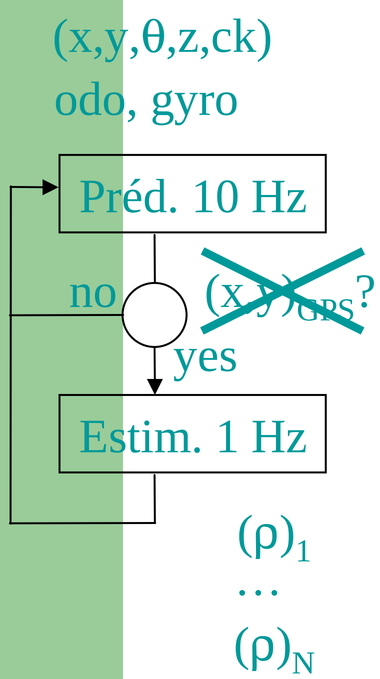
- Prediction: at the odo gyro frequency (10 to 100 Hz)
- Estimation: at the GPS output frequency (1 to 10 Hz)

Offline smoothing of vehicle position



- Smooting: Raugh’s equations (fwd rwd filter)
  - Non causal (future is known)
- Demo application: Gyrolis (freeware)

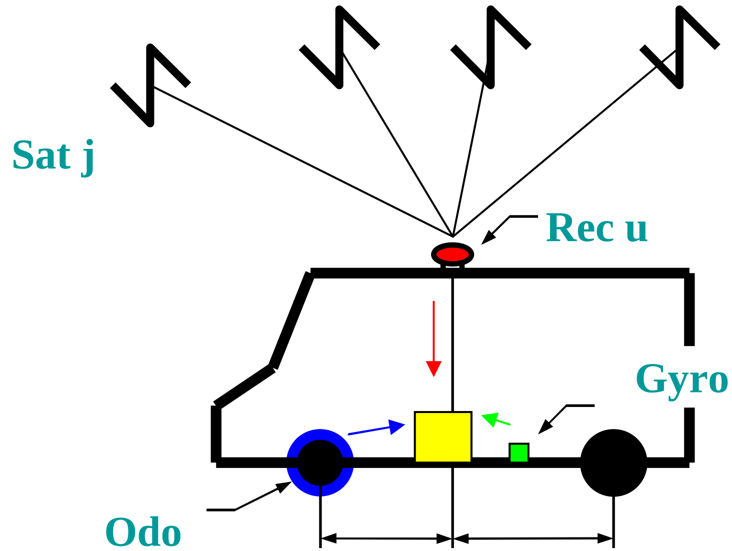
Tight coupling EKF implementation



- State: x, y, θ (like loose coupling) + height z, receiver clock
- Prediction (at 10 Hz): like loose coupling + z and ck assumed constant
- Estimation (at 1 Hz): pseudo-ranges (ρ) for N svs

Tight coupling for vehicle location

- Evolution of the state (with input) like loose coupling



- Pb in 3D: coordinates ENU: x=East, y=North, z=Up
- GPS observations: pseudo-ranges (not: lat, lon, alt, or projected positions)
- $\rho_j = \sqrt{(x_j\text{-}xu)^2 + (y_j\text{-}yu)^2 + (z_j\text{-}zu)^2} + c.ck$

Tight coupling for vehicle location

- Jacobian matrix: H (k) (non linear observation)

$$\begin{bmatrix} ax_1 & ay_1 & az_1 & 1 \\ ax_2 & ay_2 & az_2 & 1 \\ & \ldots & & \\ ax_N & ay_N & az_N & 1 \end{bmatrix}$$

- $ax_j = d\rho_j/dx$, $ay_j = d\rho_j/dy$, $az_j = d\rho_j/dz$
- N satellites

Tight coupling for vehicle location

- Use every satellite initially (N satellites visibles)
  - chi² test on the pseudo-ranges NIS
  - if the test with N svs is negative

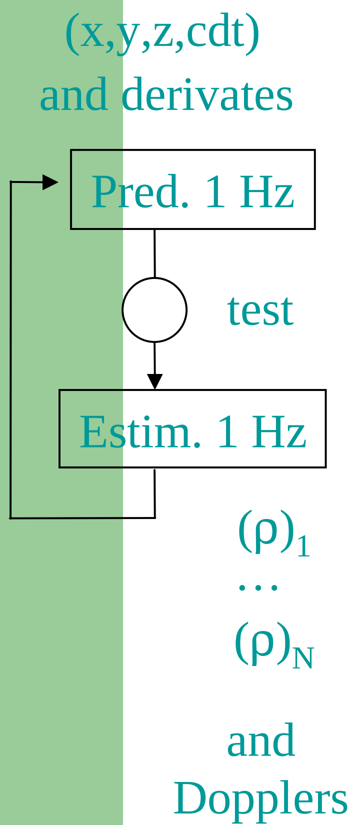
    - one makes weighted residuals (Cholesky)

    - one eliminates the sv with the highest weighted residual

    - one iterates using N-1 svs

Even if N<4 (i.e. no GPS solution) satellite data are used

GPS solution by Kalman filtering



- State: X, Y, Z, receiver clock (cdt) and their derivates
- Prediction (at 1 Hz): dynamic model
- Estimation (at 1 Hz): observation model using pseudo-ranges (ρ) and Dopplers ($\dot{\rho}$) for N svs

Constant velocity model

- State : X,vX, Y,vY, Z,vZ, cdt,$cdt^{\cdot}$
- Input : none
- State prediction :

$$X(k+1) = X(k) + vX * \Delta temps$$

$$vX(k+1) = vX(k)$$

$$cdt(k+1) = cdt(k) + cdt^{\cdot} * \Delta temps$$

$$cdt^{\cdot}(k+1) = cdt^{\cdot}(k)$$

- GPS observation : pseudo-distances et Doppler
- Gaussian noises (model and GPS measurements)

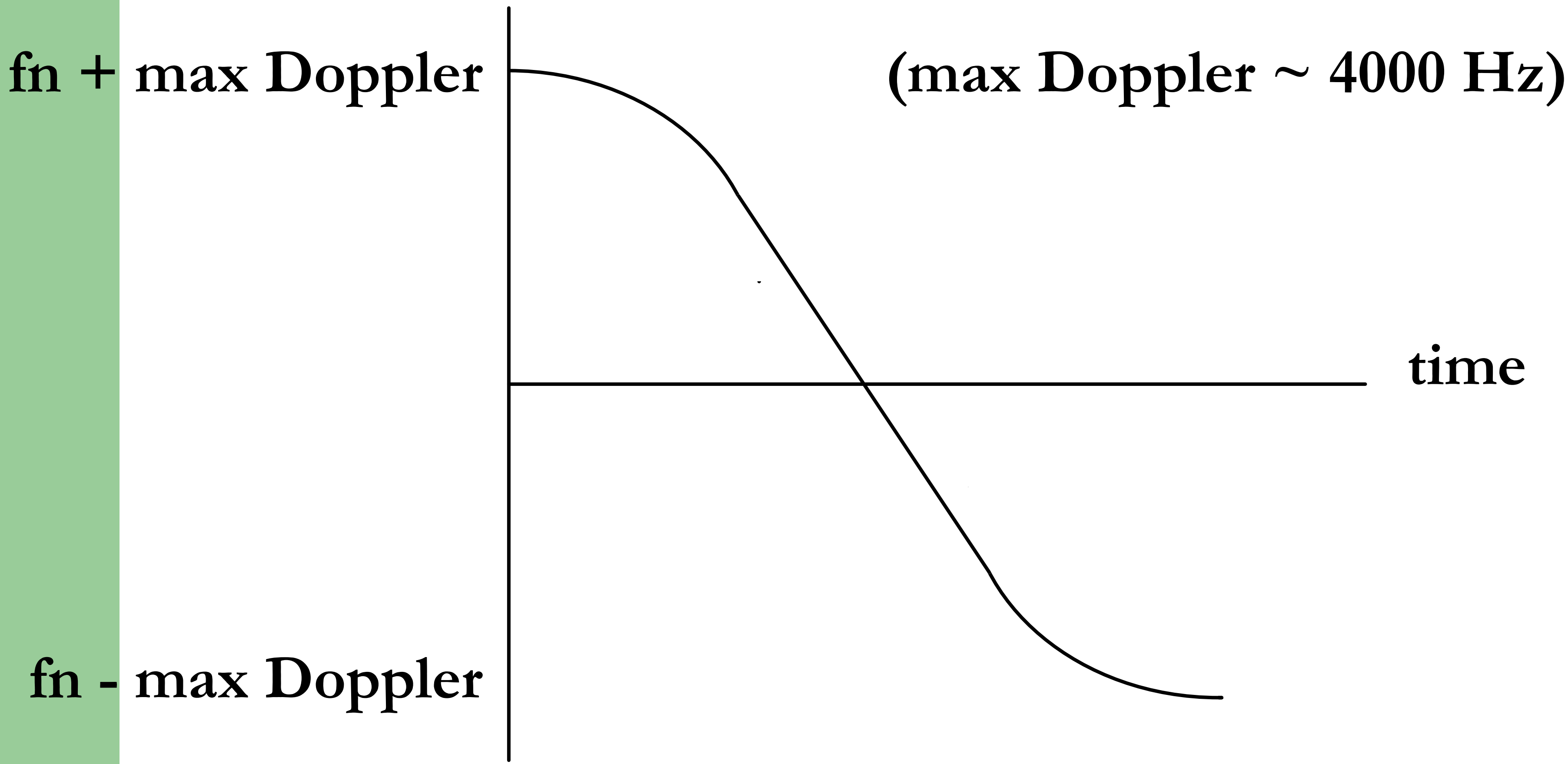
Doppler effect

- measured in frequency: fd (Hz)
- fd = fn (1 – $\dot{\rho}$/c) where
- fn is the nominal frequency (L band : 1.5 GHz)
- c is the speed of light
- $\dot{\rho}$ is the satellite / receiver range rate

Doppler effect

- $\dot{\rho}$ is the satellite / receiver range rate
- vX, vY, vZ : receiver
- vXs, vYs, vZs : satellite
- $\dot{\rho}$ = (v_rec-v_sat).u
- u : unit vector rec-sat

Doppler effect from sat rise to set



GPS solution by Kalman filtering

- Jacobian matrix: H (k) (non linear observation)

$$\begin{bmatrix} ax_1 & ay_1 & az_1 & 1 & 0 & 0 & 0 & 0 \\ & & & \ldots & & & & \\ ax_N & ay_N & az_N & 1 & 0 & 0 & 0 & 0 \\ bx_1 & by_1 & bz_1 & 0 & cx_1 & cy_1 & cz_1 & 1 \\ & & & \ldots & & & & \\ ax_N & ay_N & az_N & 0 & cx_N & cy_N & cz_N & 1 \end{bmatrix}$$

- $ax_j = d\rho_j/dx$, $ay_j = d\rho_j/dy$, $az_j = d\rho_j/dz$ for ranges (ρ_j)
- $bx_j = d\dot{\rho}_j/dx$, $by_j = d\dot{\rho}_j/dy$, $bz_j = d\dot{\rho}_j/dz$ and $cx_j = d\dot{\rho}_j/dvx$, $cy_j = d\dot{\rho}_j/dvy$, $cz_j = d\dot{\rho}_j/dvz$ Dopplers ($\dot{\rho}_j$)

GPS solution by Kalman filtering

- The pb is to fix variance matrixes characterizing noises in both dynamic model (W also noted Q) and observation model (V also noted R)

  R : characterizes errors in pseudo-ranges and Doppler

  Q : integrated white noise whose level (Sp) by trial-and-error method

Q matrix:

$$\begin{bmatrix} Sp.Ts^3/3 & Sp.Ts^2/2 \\ Sp.Ts^2/2 & Sp.Ts \end{bmatrix}$$

Ts = sampletime

GPS solution by Kalman filtering

- Use every satellite initially (N visible satellites)
  - chi² test on both range & Doppler NIS
  - if the test with N svs is negative

    - one makes weighted residuals (Cholesky)

    - one eliminates the sv with the highest weighted residual

    - one iterates using N-1 svs

innov = Y-HX
Pondération :
chol($(HPH'+R)^{-1}$)*innov

Table of χ^2

LOI DE χ^2 OU DE PEARSON

FONCTION DE RÉPARTITION

– Une variable aléatoire Z suit une loi de χ^2 si elle peut prendre toutes les valeurs positives, la densité de probabilité pour la valeur z étant

$$f(z) = \frac{1}{2^{\nu/2}\,\Gamma\left(\frac{\nu}{2}\right)}\, z^{\frac{\nu-2}{2}}\, e^{-\frac{z}{2}} \quad \text{où} \quad \Gamma(n) = \int_0^{\infty} e^{-x}\, x^{n-1}\, dx$$

ν est un nombre entier dit nombre de degrés de liberté. La moyenne de cette loi est égale à ν et sa variance à $2\,\nu$.

– La table qui suit donne la valeur χ_P^2 de la variable Z ayant la probabilité P de ne pas être dépassée, en fonction du nombre ν de degrés de liberté. Cette table est résumée par l'abaque de la page 27.

P est égale à l'aire située à gauche de l'abscisse χ_P^2 au-dessous de la courbe représentant la densité de probabilité.

$$P = \int_{-\infty}^{\chi_P^2} f(z)\, dz$$

Exemple : $\nu = 12 \quad P = 0{,}900 \quad \chi^2_{0,90} = 18{,}5$

– Dès que ν est supérieur à 30, la quantité $[\sqrt{2\,\chi^2} - \sqrt{2\nu - 1}]$ suit une loi très voisine de la loi normale réduite ce qui permet de calculer les limites χ_P^2 pour $\nu > 30$.

Exemple : Pour $\nu = 44$, quelle est la valeur de χ^2 qui a une probabilité de 0,90 de ne pas être dépassée, c'est-à-dire quelle est la valeur de $\chi^2_{0,90}$?

$$\sqrt{2\,\chi^2_{0,90}} - \sqrt{2\nu - 1} = \sqrt{2\chi^2_{0,90}} - \sqrt{87} \approx 1{,}28$$

$$\sqrt{2\,\chi^2_{0,90}} \approx 1{,}28 + \sqrt{87} = 10{,}60$$

$$\chi^2_{0,90} \quad \text{est voisin de} \quad 56{,}2$$

– La variable aléatoire $Z = U_1^2 + U_2^2 + \ldots U_n^2$, pour laquelle tous les U_i sont des variables aléatoire normales réduites et indépendantes, suit la loi de χ^2 à $\nu = n$ degrés de liberté.

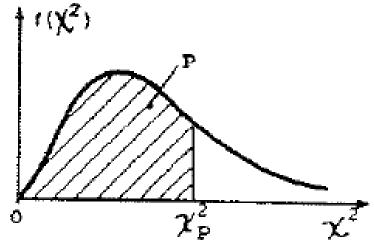
La variable de χ^2 sert, en particulier, à l'étude d'une distribution observée (chapitre II-1), au calcul de l'intervalle de confiance d'une variance (II-2), à la comparaison de variances (IV-3) et de proportions (IV-4).

Table of χ^2

Cette table donne les valeurs de χ^2_P en fonction de P et du nombre ν de degrés de liberté.

ν \ P	0,005	0,010	0,025	0,050	0,100	0,250	0,500	0,750	0,900	0,950	0,975	0,990	0,995	P / ν
1	0,0000	0,0002	0,0010	0,0039	0,0158	0,102	0,455	1,32	2,71	3,84	5,02	6,63	7,88	1
2	0,0100	0,0201	0,0506	0,103	0,211	0,575	1,39	2,77	4,61	5,99	7,38	9,21	10,6	2
3	0,0717	0,115	0,216	0,352	0,584	1,21	2,37	4,11	6,25	7,81	9,35	11,3	12,8	3
4	0,207	0,297	0,484	0,711	1,06	1,92	3,36	5,39	7,78	9,49	11,1	13,3	14,9	4
5	0,412	0,554	0,831	1,15	1,61	2,67	4,35	6,63	9,24	11,1	12,8	15,1	16,7	5
6	0,676	0,872	1,24	1,64	2,20	3,45	5,35	7,84	10,6	12,6	14,4	16,8	18,5	6
7	0,989	1,24	1,69	2,17	2,83	4,25	6,35	9,04	12,0	14,1	16,0	18,5	20,3	7
8	1,34	1,65	2,18	2,73	3,49	5,07	7,34	10,2	13,4	15,5	17,5	20,1	22,0	8
9	1,73	2,09	2,70	3,33	4,17	5,90	8,34	11,4	14,7	16,9	19,0	21,7	23,6	9
10	2,16	2,56	3,25	3,94	4,87	6,74	9,34	12,5	16,0	18,3	20,5	23,2	25,2	10
11	2,60	3,05	3,82	4,57	5,58	7,58	10,3	13,7	17,3	19,7	21,9	24,7	26,8	11
12	3,07	3,57	4,40	5,23	6,30	8,44	11,3	14,8	18,5	21,0	23,3	26,2	28,3	12
13	3,57	4,11	5,01	5,89	7,04	9,30	12,3	16,0	19,8	22,4	24,7	27,7	29,8	13
14	4,07	4,66	5,63	6,57	7,79	10,2	13,3	17,1	21,1	23,7	26,1	29,1	31,3	14
15	4,60	5,23	6,26	7,26	8,55	11,0	14,3	18,2	22,3	25,0	27,5	30,6	32,8	15
16	5,14	5,81	6,91	7,96	9,31	11,9	15,3	19,4	23,5	26,3	28,8	32,0	34,3	16
17	5,70	6,41	7,56	8,67	10,1	12,8	16,3	20,5	24,8	27,6	30,2	33,4	35,7	17
18	6,26	7,01	8,23	9,39	10,9	13,7	17,3	21,6	26,0	28,9	31,5	34,8	37,2	18
19	6,84	7,63	8,91	10,1	11,7	14,6	18,3	22,7	27,2	30,1	32,9	36,2	38,6	19
20	7,43	8,26	9,59	10,9	12,4	15,5	19,3	23,8	28,4	31,4	34,2	37,6	40,0	20
21	8,03	8,90	10,3	11,6	13,2	16,3	20,3	24,9	29,6	32,7	35,5	38,9	41,4	21
22	8,64	9,54	11,0	12,3	14,0	17,2	21,3	26,0	30,8	33,9	36,8	40,3	42,8	22
23	9,26	10,2	11,7	13,1	14,8	18,1	22,3	27,1	32,0	35,2	38,1	41,6	44,2	23
24	9,89	10,9	12,4	13,8	15,7	19,0	23,3	28,2	33,2	36,4	39,4	43,0	45,6	24
25	10,5	11,5	13,1	14,6	16,5	19,9	24,3	29,3	34,4	37,7	40,6	44,3	46,9	25
26	11,2	12,2	13,8	15,4	17,3	20,8	25,3	30,4	35,6	38,9	41,9	45,6	48,3	26
27	11,8	12,9	14,6	16,2	18,1	21,7	26,3	31,5	36,7	40,1	43,2	47,0	49,6	27
28	12,5	13,6	15,3	16,9	18,9	22,7	27,3	32,6	37,9	41,3	44,5	48,3	51,0	28
29	13,1	14,3	16,0	17,7	19,8	23,6	28,3	33,7	39,1	42,6	45,7	49,6	52,3	29
30	13,8	15,0	16,8	18,5	20,6	24,5	29,3	34,8	40,3	43,8	47,0	50,9	53,7	30
40	20,7	22,2	24,4	28,5	29,1	33,7	39,3	45,6	51,8	55,8	59,3	63,7	66,8	40
50	28,0	29,7	32,4	34,8	37,7	42,9	49,3	56,3	63,2	67,5	71,4	76,2	79,5	50
60	35,5	37,5	40,5	43,2	46,5	52,3	59,3	67,0	74,4	79,1	83,3	88,4	92,0	60
70	43,3	45,4	48,8	51,7	55,3	61,7	69,3	77,6	85,5	90,5	95,0	100,4	104,2	70
80	51,2	53,5	57,2	60,4	64,3	71,1	79,3	88,1	96,6	101,9	106,6	112,4	116,3	80
90	59,2	61,8	65,6	69,1	73,3	80,6	89,3	98,6	107,6	113,1	118,1	124,1	128,3	90
100	67,3	70,1	74,2	77,9	82,4	90,1	99,3	109,1	118,5	124,3	129,6	135,8	140,2	100

Table of χ^2



Cet abaque fournit la valeur de χ_P^2 en fonction de P et du nombre ν de degrés de liberté.