GLO-4001/7021
INTRODUCTION À LA ROBOTIQUE MOBILE

Fusion de capteurs

Automne 2019

Philippe Giguère

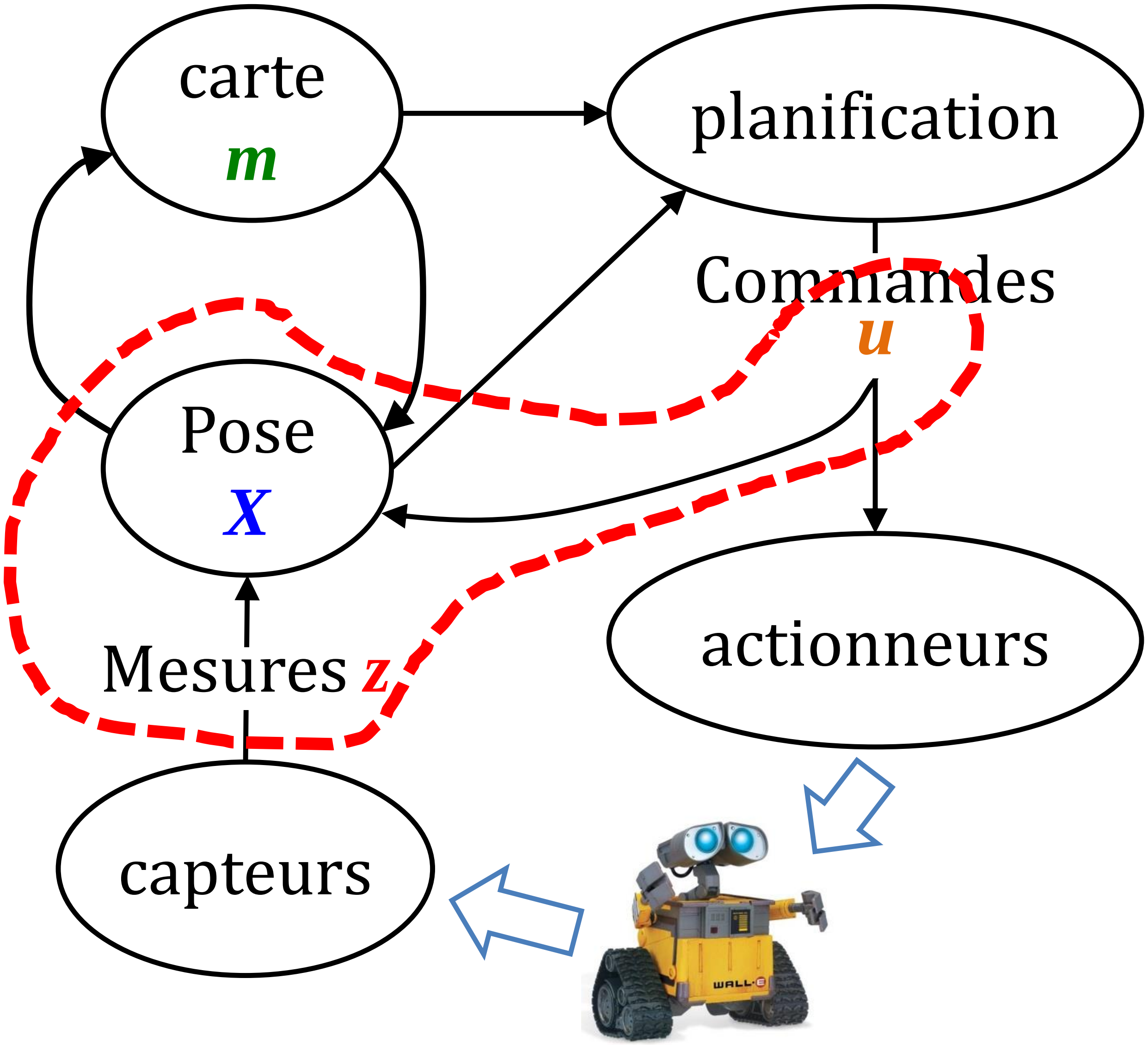
Bourses CRSNG à la maîtrise

- Date limite d'application : **1er décembre**
- Bourse de 17,500$ (sur 12 mois)
- Moyenne de **3,67** dans les 2 dernières années
- http://www.nserc-crsng.gc.ca/Students-Etudiants/PG-CS/CGSM-BESCM_fra.asp

Varia

- TP1 : correction tardive
- Présentations orales GLO-7021 : 29 novembre
- Envoyer votre suggestion d'article pour la présentation orale

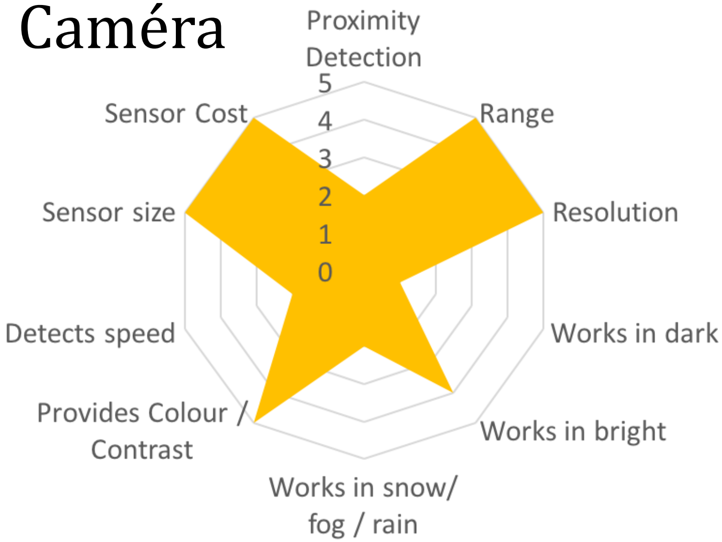
Feuille de route



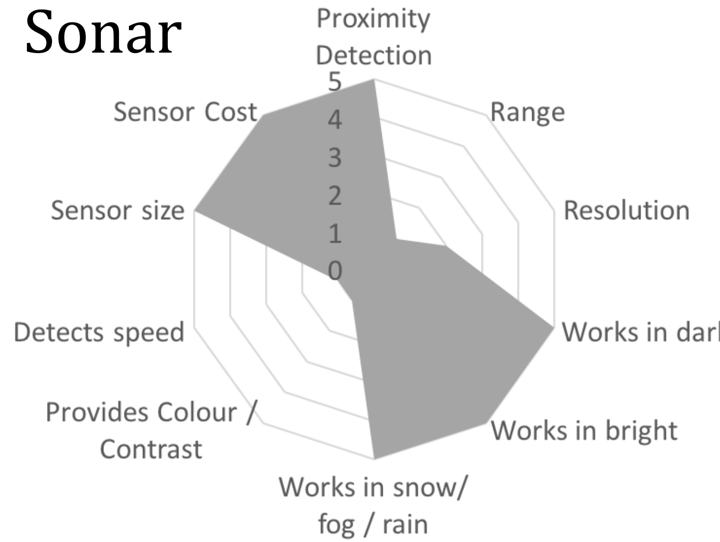
Concept #1
Fusion de capteurs

Capteurs : forces et faiblesses

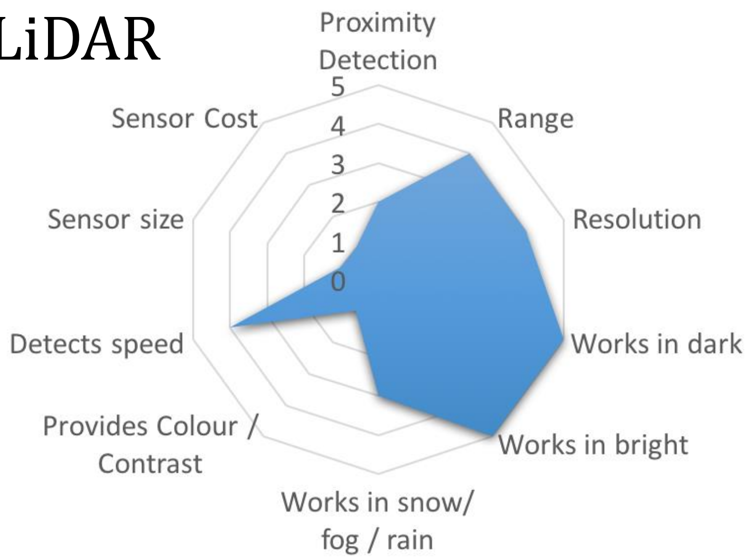
Caméra



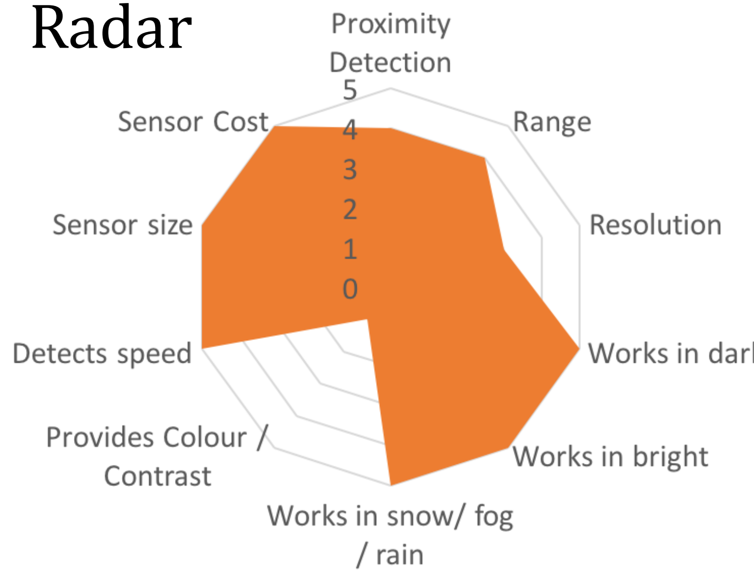
Sonar



LiDAR



Radar



https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/

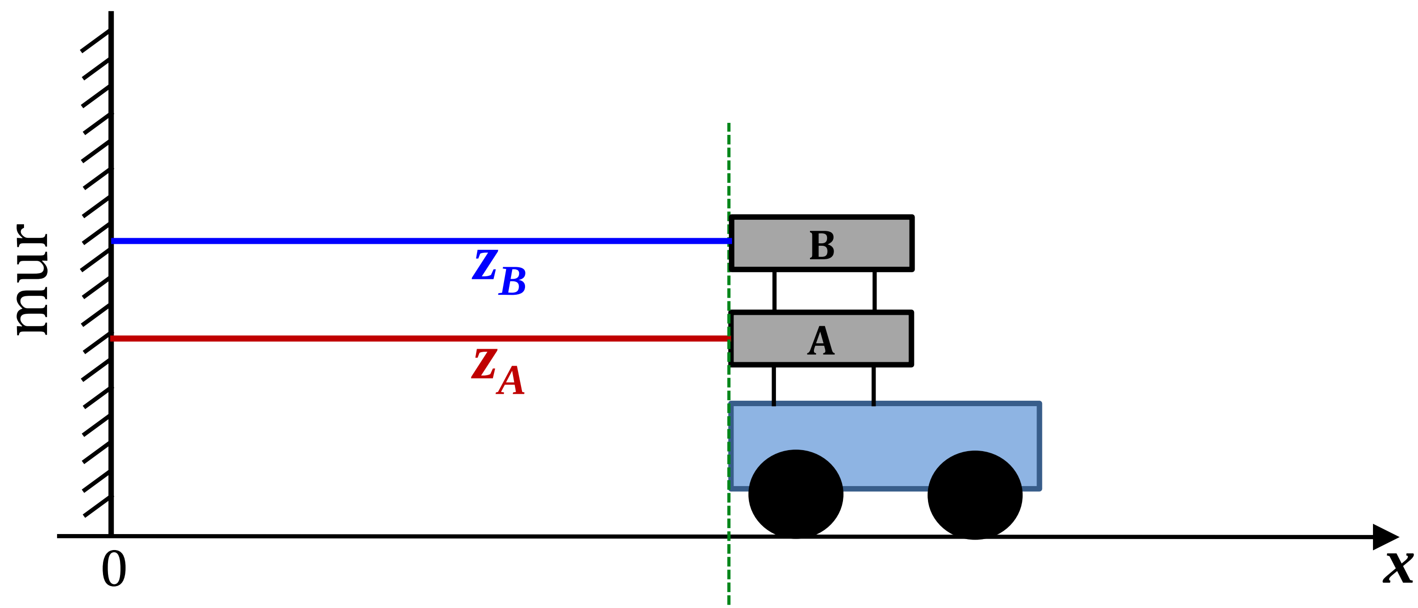
Fusion de capteurs/information

- Capteurs travaillent dans différentes modalités / grandeurs physiques :
  - LiDAR (temps de vol lumière, mètre)
  - odomètre (pulsations, mètre)
  - caméra (intensité lumineuse, pixel)
  - centrale inertielle (accél., gyro, magnéto)
- On a souvent aussi des estimés préalables de pose X
- Comment combiner l'information de ces différentes sources de <u>façon optimale</u>*, en tenant compte des incertitudes (bruits)

**quand c'est possible/calculable en un temps raisonnable*

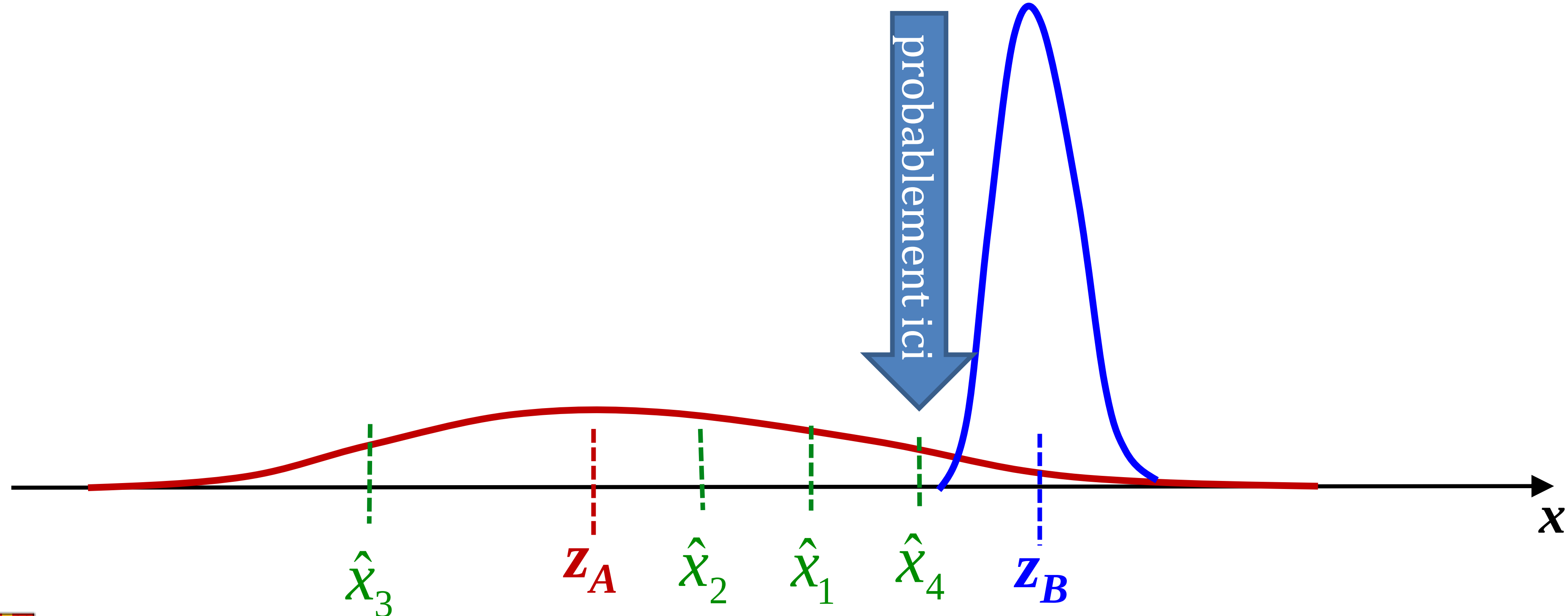
Fusion de capteurs : cas simple

- Soit un robot équipé de deux capteurs de distance (A et B) ayant des bruits différents (σ_A^2 et σ_B^2)



Fusion de capteurs

- Quel est le meilleur estimé $\hat{x}$ de la position du robot à partir des deux mesures z_A et z_B ?
  en tenant compte des bruits/incertitudes σ_A^2 et σ_B^2?



Fusion de capteur : moyenne pondérée

- Combiner les *i* mesures (évidences) obtenues, basée sur l'inverse des variances σ^2_i

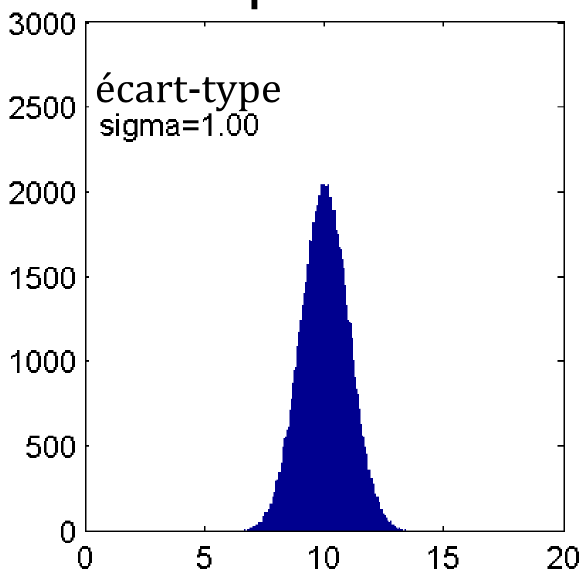
- Donner plus d'importance à la mesure (évidence) ayant une variance σ^2_i plus petite.

$$\boxed{\text{information} \propto \frac{1}{\sigma_i^2}}$$

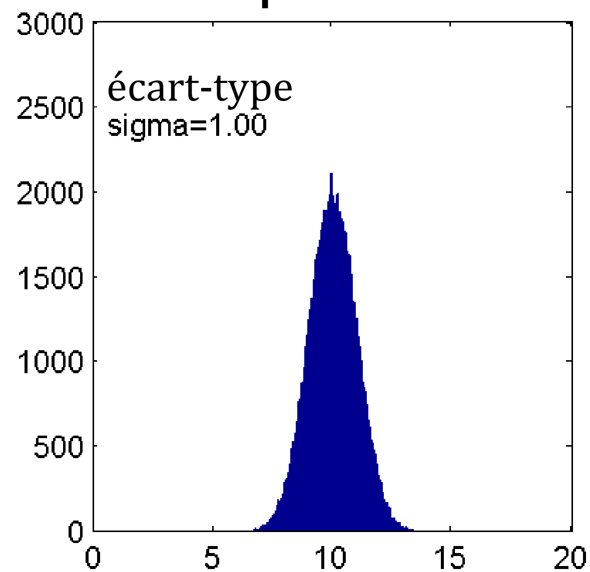
Fusion de capteurs : moyenne

Performances identiques, $\sigma_1^2 = \sigma_2^2$ x=10

capteur 1



capteur 2



modèle du capteur

$$z_i = x + N(0, \sigma_i^2)$$

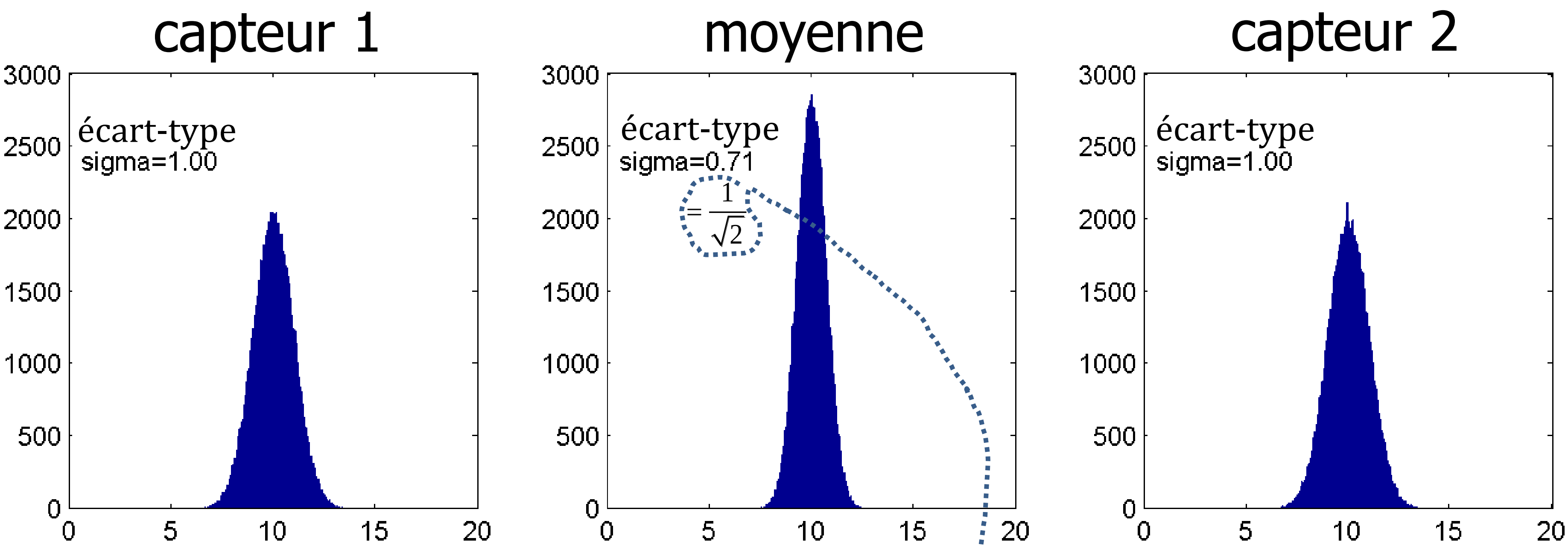
-parfaitement linéaire
-bruit sans biais
-portée illimitée

$$z_1 = 10 + N(0, \sigma_1^2 = 1^2)$$

$$z_2 = 10 + N(0, \sigma_2^2 = 1^2)$$

Fusion de capteurs : moyenne

Performances identiques, $\sigma_1^{\,2} = \sigma_2^{\,2}$ x=10



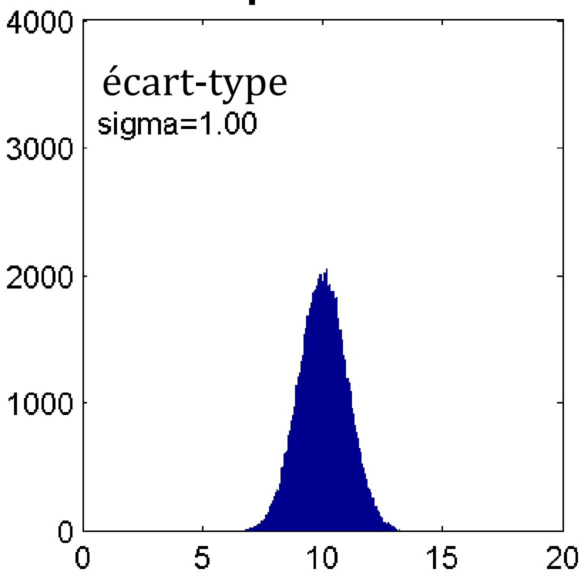
$$\color{red}{z_1} = 10 + N(0, \sigma_1^2 = 1^2) \quad \Rightarrow \quad z_3 = \frac{\color{red}{z_1} + \color{green}{z_2}}{2} \quad \Leftarrow \quad \color{green}{z_2} = 10 + N(0, \sigma_2^2 = 1^2)$$

$$\sigma^2_{moyenne} = \mathrm{var}\left\{\frac{1}{2}z_1 + \frac{1}{2}z_2\right\} = \mathrm{var}\left\{\frac{1}{2}z_1\right\} + \mathrm{var}\left\{\frac{1}{2}z_2\right\} = \frac{1}{4}\mathrm{var}\{z_1\} + \frac{1}{4}\mathrm{var}\{z_2\} = \frac{1}{4}\sigma_1^2 + \frac{1}{4}\sigma_2^2 = \frac{1}{2}(1) = \frac{1}{2}$$

Fusion de capteurs : moyenne

Performances **non**-identiques, $\sigma_1^2 > \sigma_2^2$ $\quad x=10$

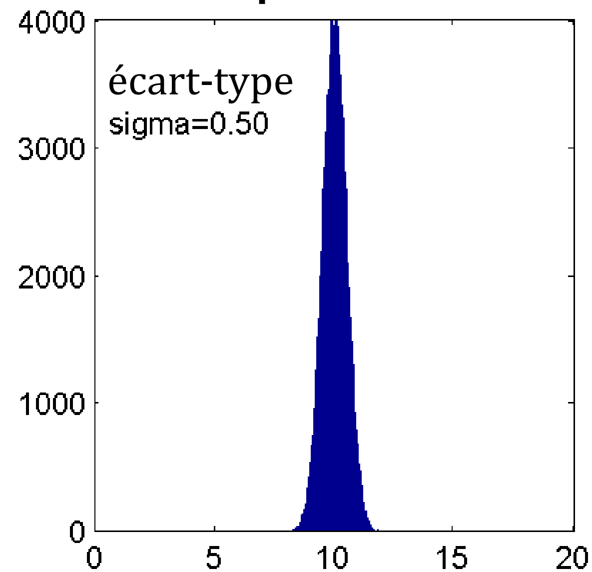
capteur 1



moyenne

?

capteur 2



$$z_1 = 10 + N(0, \sigma_1^2 = 1^2)$$

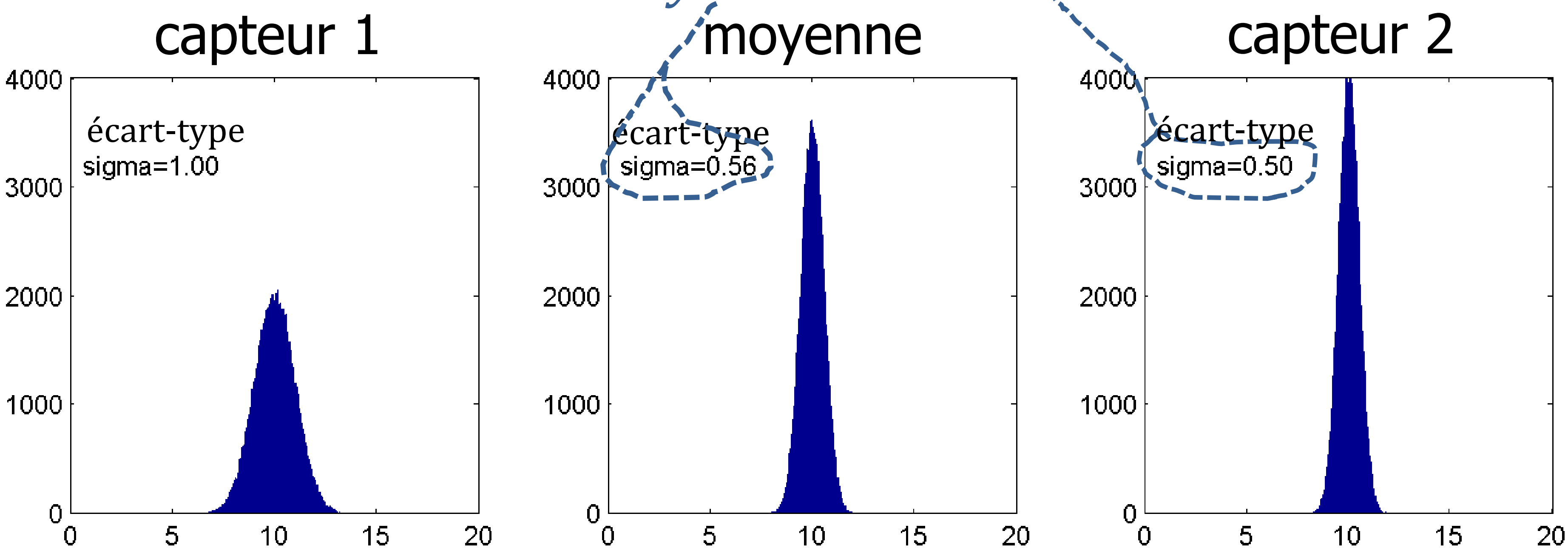
$$z_3 = \frac{z_1 + z_2}{2}$$

$$z_2 = 10 + N(0, \sigma_2^2 = 0.5^2)$$

Fusion de capteurs : moyenne

Performances **non**-identiques, $\sigma_1^2 > \sigma_2^2$ x=10

$\sigma_{moyenne}$ **est pire!**



$$z_1 = 10 + N(0, \sigma_1^2 = 1^2) \Rightarrow z_3 = \frac{z_1 + z_2}{2} \Leftarrow z_2 = 10 + N(0, \sigma_2^2 = 0.5^2)$$

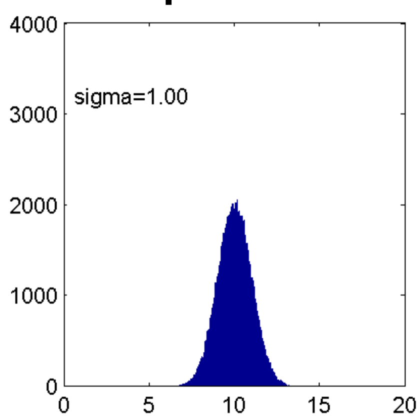
Puis-je quand même soutirer de l'info du capteur 1?

UNIVERSITÉ LAVAL

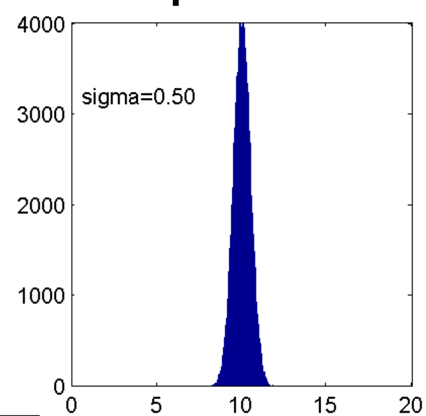
Fusion de capteurs : moyenne pondérée

Capteurs **non**-identiques, $\boldsymbol{\sigma_1}^2 > \boldsymbol{\sigma_2}^2$

capteur 1



capteur 2



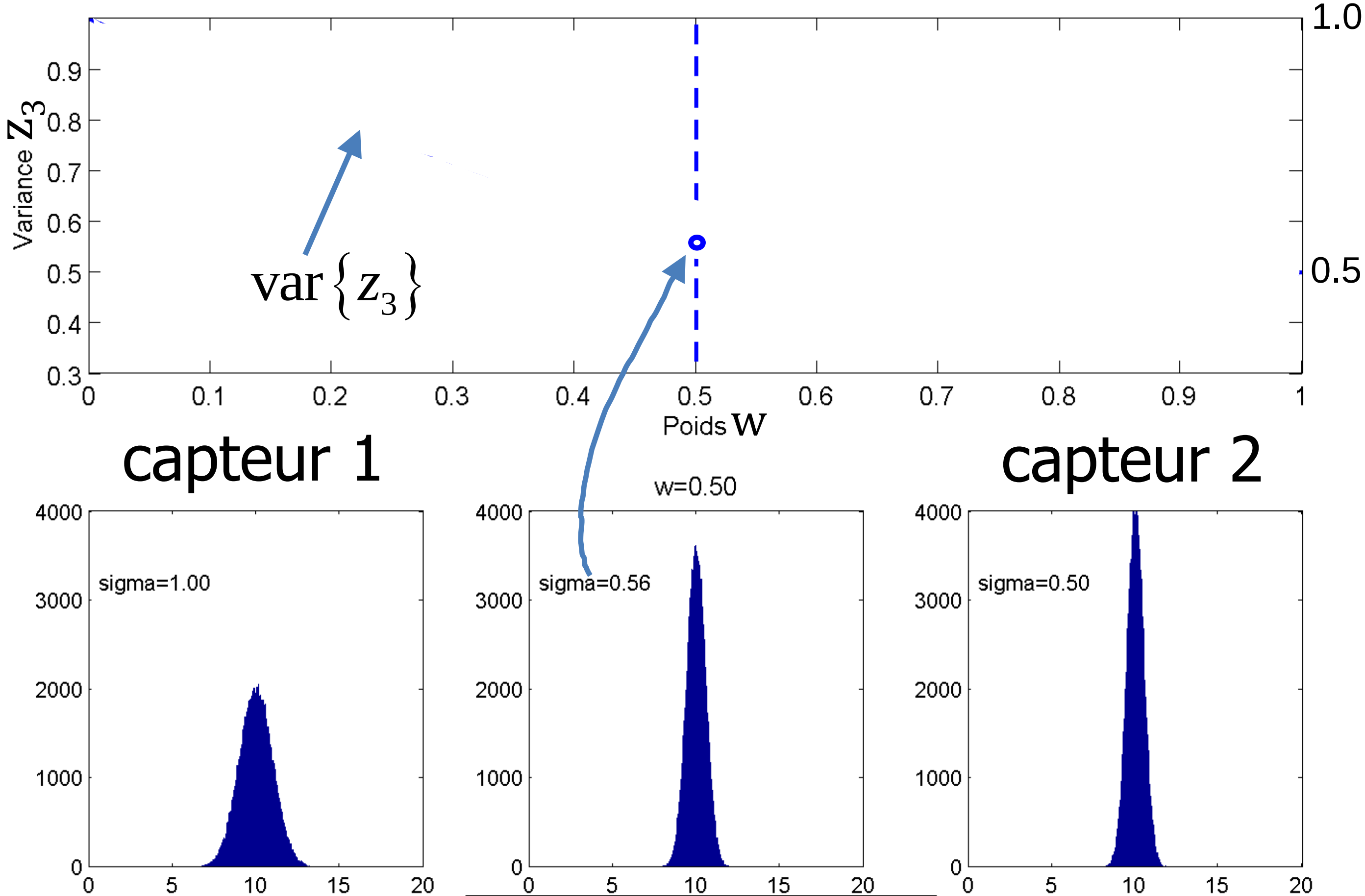
$z_1 = 10 + N(0, \sigma_1^2 = 1)$

$$\boxed{z_3 = (1-w) z_1 + w z_2}$$
$$0 \le w \le 1$$

$z_2 = 10 + N(0, \sigma_2^2 = 0.5^2)$

Fusion de capteurs : moyenne pondérée

Capteurs
non-identiques, $\sigma_1^2 > \sigma_2^2$



$z_1 = 10 + N(0, \sigma_1^2 = 1)$

$$\boxed{z_3 = (1-w) z_1 + w z_2}$$
$$0 \le w \le 1$$

w=0.50

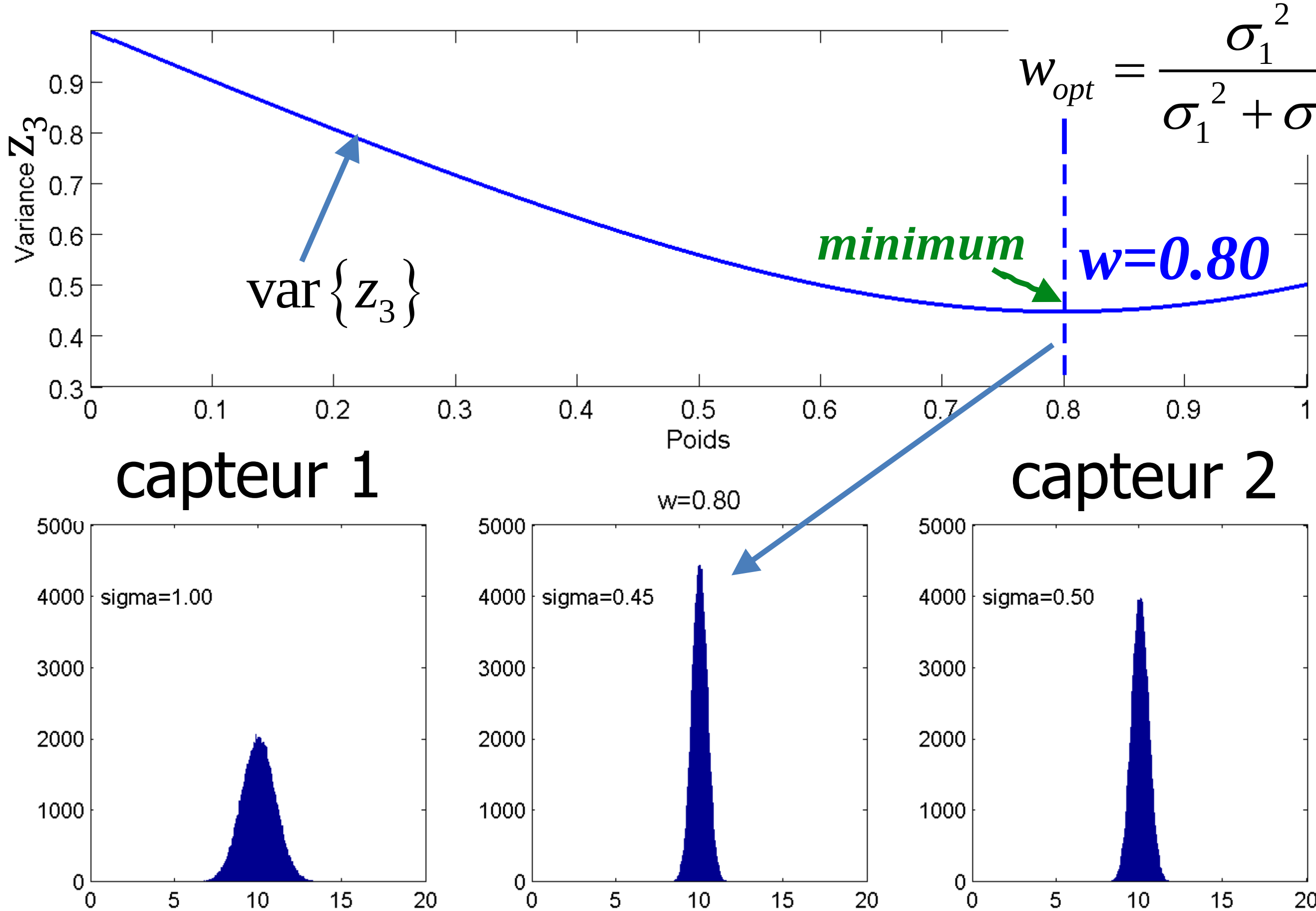
$z_2 = 10 + N(0, \sigma_2^2 = 0.5^2)$

UNIVERSITÉ LAVAL

Fusion de capteurs : moyenne pondérée

Capteurs **non-identiques, $\sigma_1^2 > \sigma_2^2$**

$$w_{opt} = \frac{\sigma_1^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}}$$



$z_1 = 10 + N(0, \sigma_1^2 = 1)$

$$\boxed{z_3 = (1-w)z_1 + wz_2}$$
$$0 \le w \le 1$$

$z_2 = 10 + N(0, \sigma_2^2 = 0.5^2)$

UNIVERSITÉ LAVAL

Preuve poids *w* optimal

$$z_3 = (1-w)z_1 + wz_2$$

variances

$$\sigma_3{}^2 = (1\text{-}w)^2\sigma_1{}^2 + w^2\sigma_2{}^2$$

(x_1 et x_2 sont indépendants)

Pour minimiser fonction, cherche $\dfrac{d}{dw}\sigma_3^2 = 0$

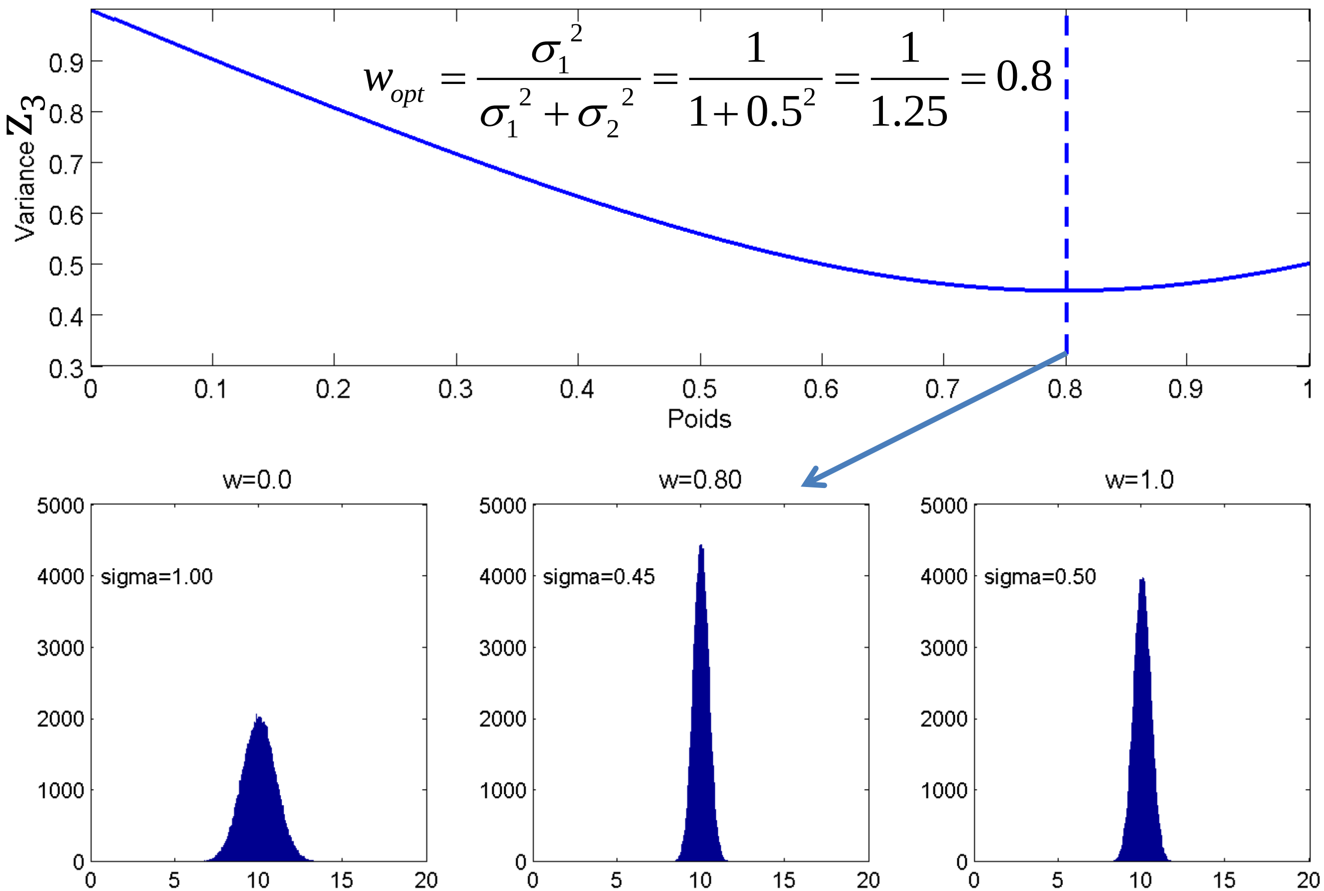
$$\frac{d}{dw}\sigma_3{}^2 = \frac{d}{dw}\left\{(1\text{-}w)^2\sigma_1{}^2 + w^2\sigma_2{}^2\right\} = 2(1\text{-}w)\sigma_1{}^2(-1) + 2w\sigma_2{}^2 = 0$$

$$2(w-1)\sigma_1{}^2 + 2w\sigma_2{}^2 = 0$$

Poids optimal minimisant variance z_3 : $\boxed{w = \dfrac{\sigma_1{}^2}{\sigma_1{}^2 + \sigma_2{}^2}}$

p.103 du manuel

Fusion de capteurs : moyenne pondérée



$z_1 = 10 + N(0, \sigma_1^2 = 1^2)$

$$\boxed{z_3 = (1-w) z_1 + w z_2}$$

$z_2 = 10 + N(0, \sigma_2^2 = 0.5^2)$

Fusion de capteurs : résumé

- J'ai deux mesures : z_1 et z_2
- J'ai les variances associées : σ_1^2 et σ_2^2

combinaison optimale

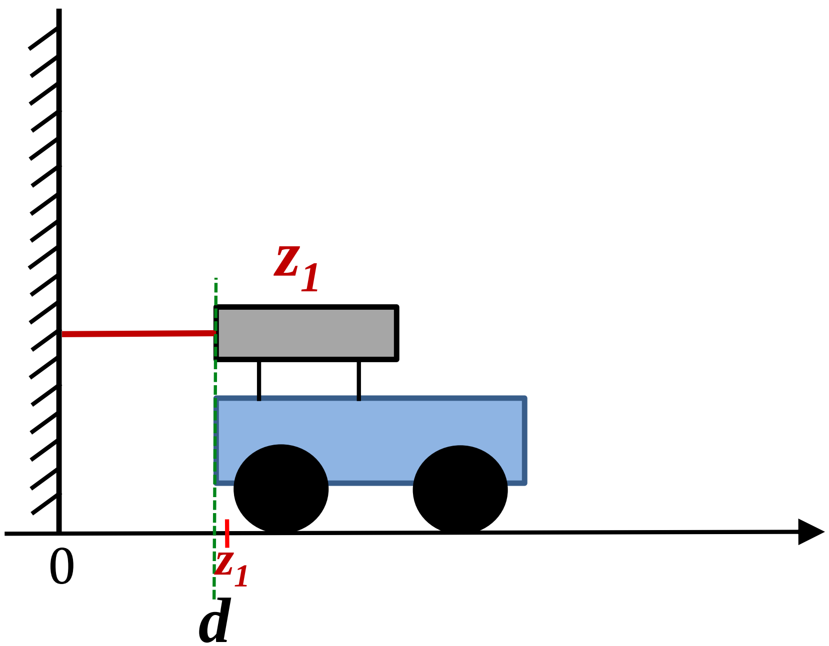
$$\boxed{z_3 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} z_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} z_2}$$

On fait **plus confiance** à la mesure ou l'évidence avec la **plus petite variance** σ^2

Concept #2
Moyenne au fil du temps

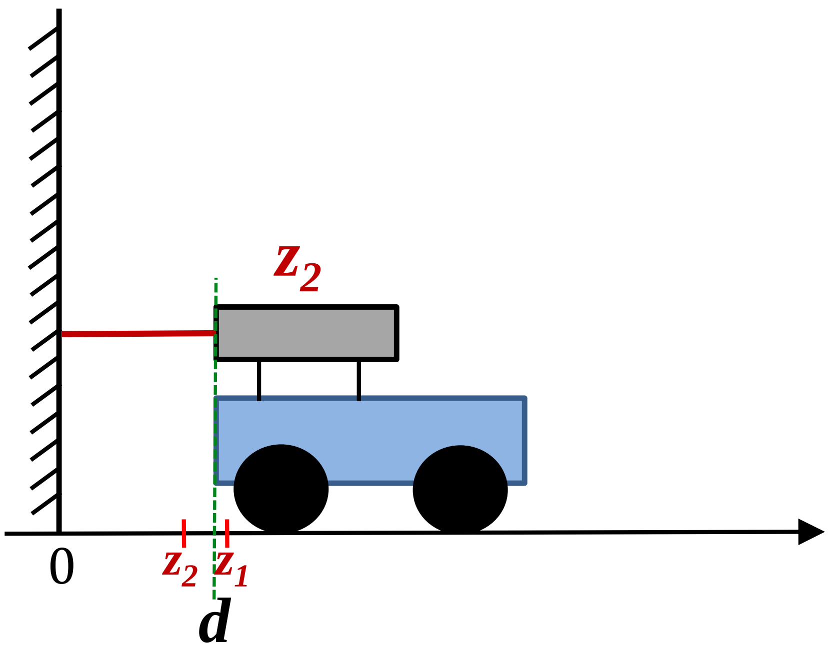
Moyenne au fil du temps

- Après chaque mesure z_t, avoir le meilleur estimé disponible de la position d d'un robot immobile
- Prendre plusieurs mesures du même capteur



Moyenne au fil du temps

- Après chaque mesure z_t, avoir le meilleur estimé disponible de la position d d'un robot immobile
- Prendre plusieurs mesures du même capteur

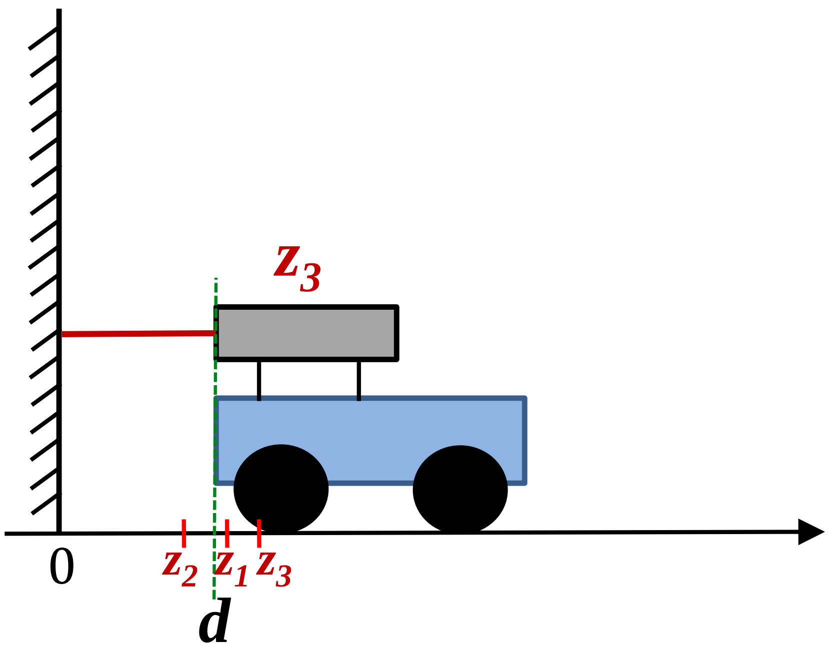


$$\hat{d}(1) = z_1$$

$$\hat{d}(2) = \frac{z_1 + z_2}{2}$$

Moyenne au fil du temps

- Après chaque mesure z_t, avoir le meilleur estimé disponible de la position *d* d'un robot immobile
- Prendre plusieurs mesures du même capteur



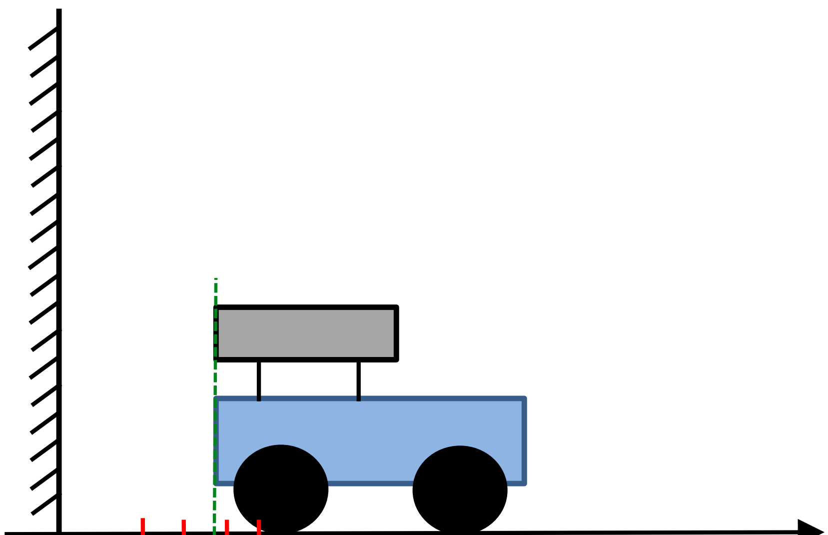
$$\hat{d}(1) = z_1$$

$$\hat{d}(2) = \frac{z_1 + z_2}{2}$$

$$\hat{d}(3) = \frac{z_1 + z_2 + z_3}{3}$$

Moyenne au fil du temps

- Après chaque mesure z_t, avoir le meilleur estimé disponible de la position d d'un robot immobile
- Prendre plusieurs mesures du même capteur



$$\hat{d}(1) = z_1$$

$$\hat{d}(2) = \frac{z_1 + z_2}{2}$$

$$\hat{d}(3) = \frac{z_1 + z_2 + z_3}{3}$$

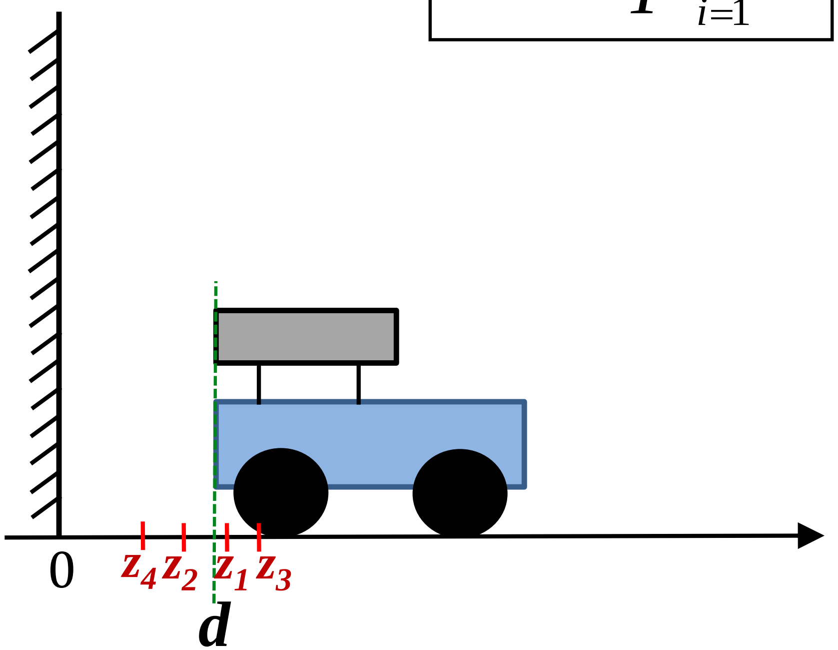
$$\hat{d}(4) = \frac{z_1 + z_2 + z_3 + z_4}{4}$$

etc…

Moyenne au fil du temps

- Avec le temps, la série s'allonge… et le temps de calcul/espace de stockage pour les z_i aussi!

$$\boxed{\hat{d}(t) = \frac{1}{T}\sum_{i=1}^{T} z_i}$$



$$\hat{d}(1) = z_1$$

$$\hat{d}(2) = \frac{z_1 + z_2}{2}$$

$$\hat{d}(3) = \frac{z_1 + z_2 + z_3}{3}$$

$$\hat{d}(4) = \frac{z_1 + z_2 + z_3 + z_4}{4}$$

Moyenne récursive

- Garder en mémoire 2 variables :
  - estimé moyen $\hat{x}(k)$ *(estimé de x à l'instant k)*
  - estimé $P(k)$ de la variance de $\hat{x}(k)$
- On connait la variance σ^2_z du capteur
- Après nouvelle mesure $z(k+1)$, on la combine avec $\hat{x}(k)$ pour avoir un nouvel estimé $\hat{x}(k+1), P(k+1)$

on a vu précédemment

$$w = \frac{\sigma_1^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}}$$

$$z_3 = (1-w)z_1 + wz_2$$

$$\mathrm{var}\{z_3\} = \mathrm{var}\{(1-w)z_1 + wz_2\}$$

reformulation:

$z_1 = \hat{x}(k)$

$z_2 = z(k+1)$

$z_3 = \hat{x}(k+1)$

$$w = \frac{P(k)}{P(k) + \sigma_Z^2}$$

$$\hat{x}(k+1) = \hat{x}(k) + w(z(k+1) - \hat{x}(k))$$

$$P(k+1) = (1-w)P(k)$$

variance diminue avec le temps

UNIVERSITÉ LAVAL

Preuve sur la variance $P(k)$

$$\boxed{\hat{x}(k+1) = \hat{x}(k) + w(z(k+1) - \hat{x}(k))}$$

$$\boxed{P(k) = Var\{\hat{x}(k)\}}$$

$$\boxed{\hat{x}(k+1) = (1-w)\hat{x}(k) + wz(k+1)}$$

$$\boxed{w = \frac{P(k)}{P(k) + \sigma_Z^2}}$$

$$Var\{\hat{x}(k+1)\} = Var\{(1-w)\hat{x}(k) + wz(k+1)\}$$

$$Var\{\hat{x}(k+1)\} = (1-w)^2 Var\{\hat{x}(k)\} + w^2 Var\{(z(k+1)\}$$

$$Var\{\hat{x}(k+1)\} = (1-w)^2 P(k) + w^2 \sigma_z^2$$

On choisi le poids optimal : $w = \dfrac{P(k)}{P(k) + \sigma_Z^2}$, donc : $\sigma_Z^2 = P(k)(\dfrac{1}{w} - 1)$

$$Var\{\hat{x}(k+1)\} = (1-w)^2 P(k) + w^2 P(k)(\frac{1}{w} - 1)$$

$$Var\{\hat{x}(k+1)\} = (1 - 2w + w^2)P(k) + \frac{w^2}{w} P(k) - w^2 P(k)$$

$$Var\{\hat{x}(k+1)\} = (1-w)P(k)$$

Exemple d'exécution

Soit les mesures suivantes : $Z = \{z_1 = 10, z_2 = 12, z_3 = 11\} \quad \sigma_z^2 = 1$

$$w = \frac{P(k)}{P(k) + \sigma_Z^2}$$

$$\hat{x}(k+1) = \hat{x}(k) + w(z(k+1) - \hat{x}(k))$$

$$P(k+1) = (1-w)P(k)$$

1

$$\hat{x}(1) = z(1) = 10$$

$$P(1) = \sigma_z^2 = 1$$

$z(2) = 12$

$$w = \frac{P(1)}{P(1) + \sigma_z^2} = \frac{1}{1+1} = \frac{1}{2}$$

2

$$\hat{x}(2) = \hat{x}(1) + w(z(2) - \hat{x}(1)) = 10 + \frac{1}{2}(12-10) = 11$$

$$P(2) = (1 - \frac{1}{2})P(1) = \frac{1}{2}$$

$z(3) = 11$

$$w = \frac{P(2)}{P(2) + \sigma_z^2} = \frac{1/2}{1/2+1} = \frac{1}{3}$$

3

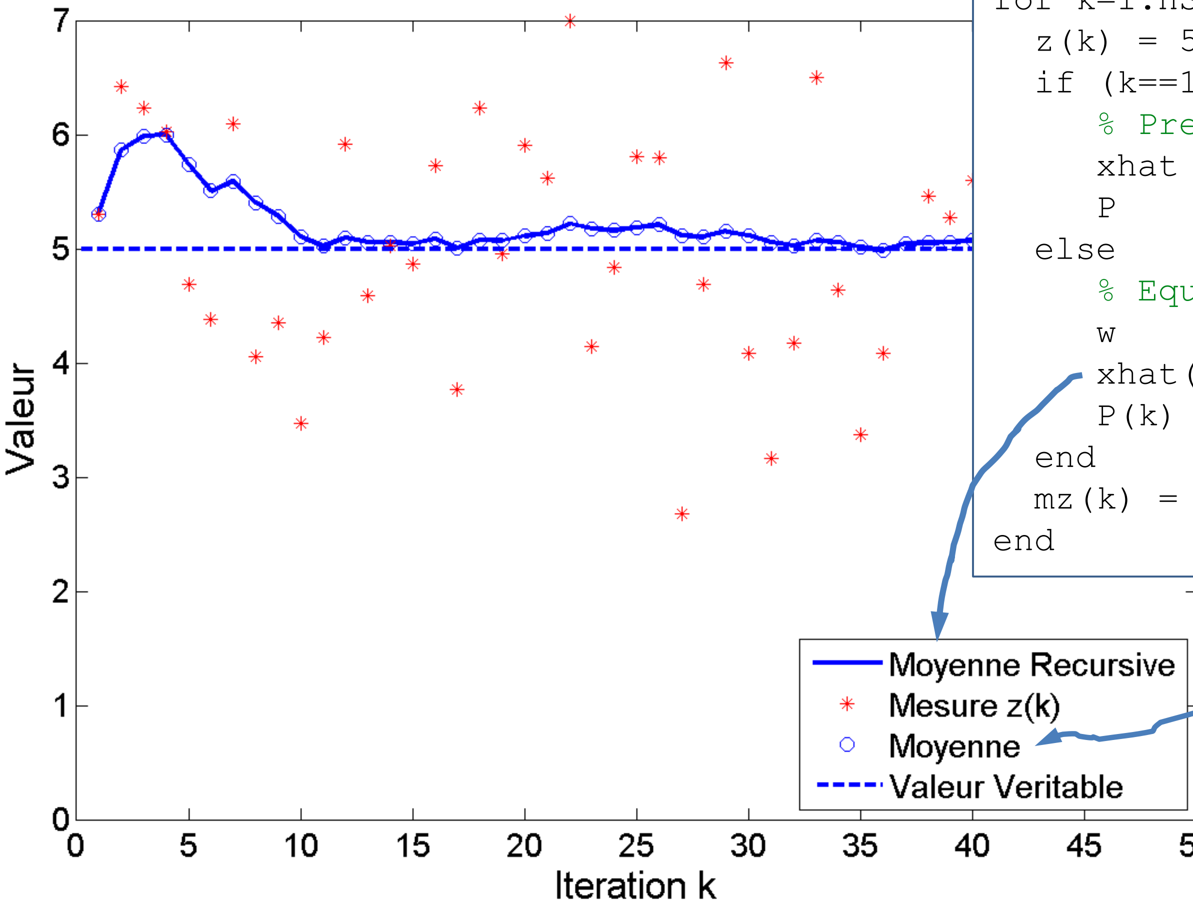
$$\hat{x}(3) = \hat{x}(2) + w(z(3) - \hat{x}(2)) = 11 + \frac{1}{3}(11-11) = 11$$

$$P(3) = (1 - \frac{1}{3})P(2) = \frac{1}{3}$$

On peut donc oublier les mesures z_i après les mises-à-jour

Implémentation filtre récursif *matlab*

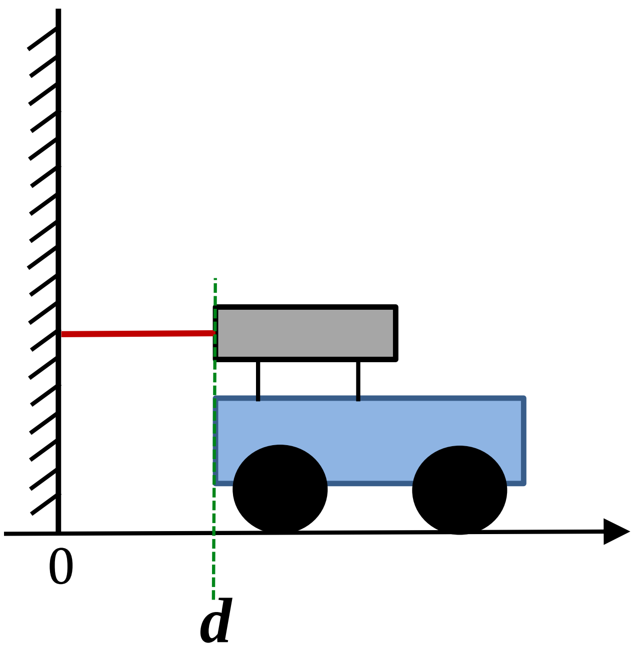
```
% Boucle de capture d'echantillons
for k=1:nStep
  z(k) = 5 + Scapteur*randn; % Z bruitee
  if (k==1)
     % Premier echantillon, on initialise
     xhat = z(k);
     P    = Scapteur^2;
  else
     % Equation (4.4) manuel p.104
     w       = P(k-1)/(P(k-1)+Scapteur^2);
     xhat(k) = xhat(k-1) + w*(z(k)-xhat(k-1));
     P(k)    = (1-w)*P(k-1);
  end
  mz(k) = mean(z(1:k));
end
```

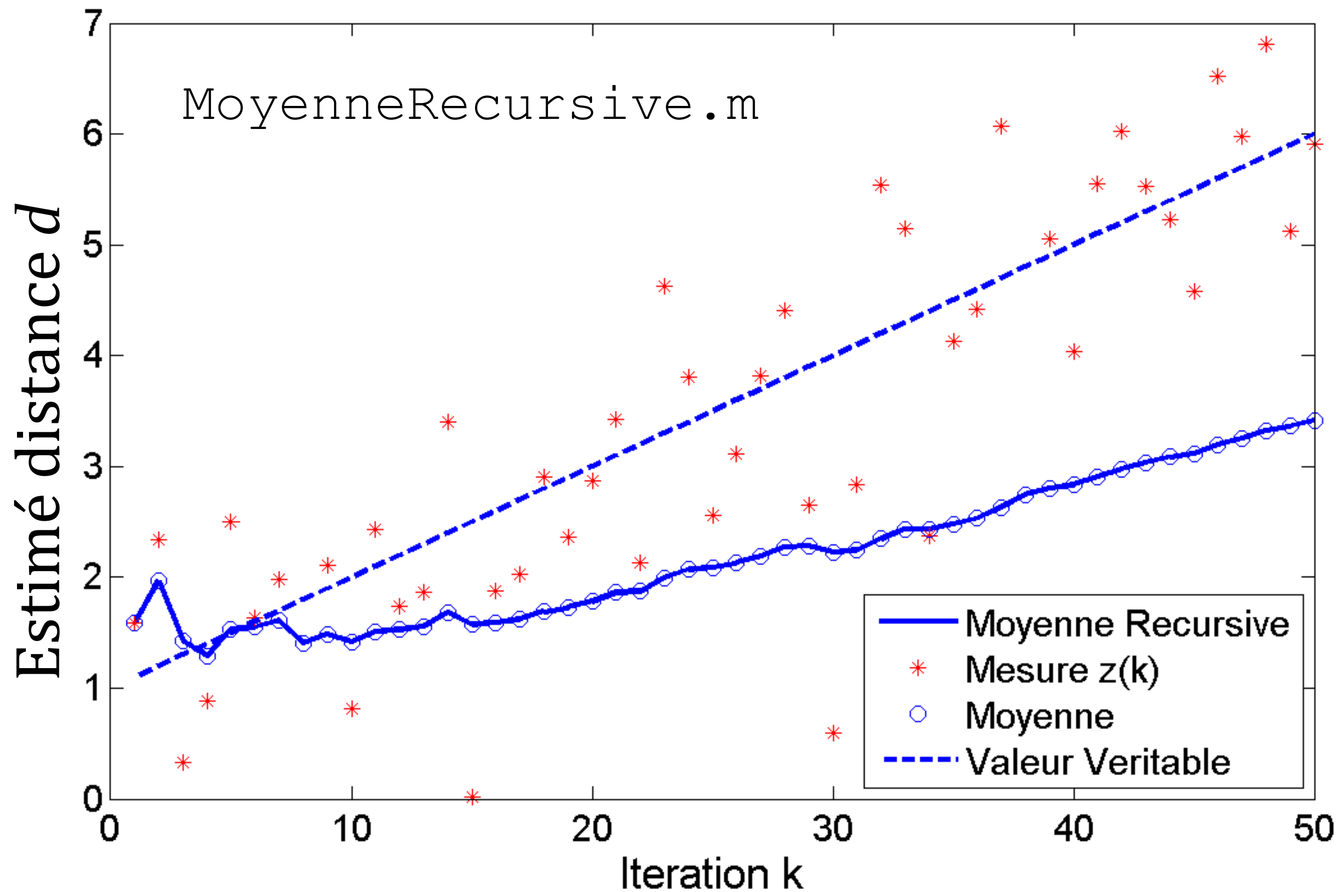


Notez comment la moyenne se rapproche de la valeur véritable, au fil des données

Moyenne si la valeur réelle change…

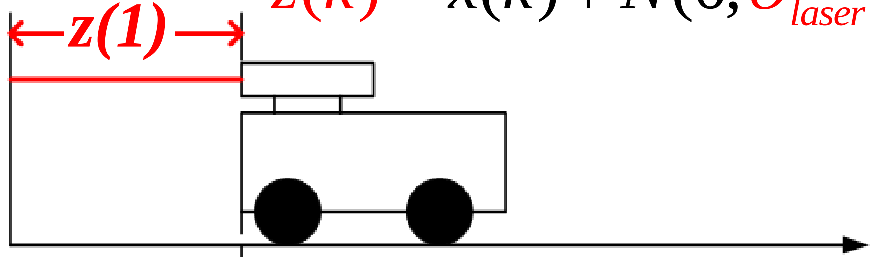
- Si le robot se déplace… la moyenne perd son sens





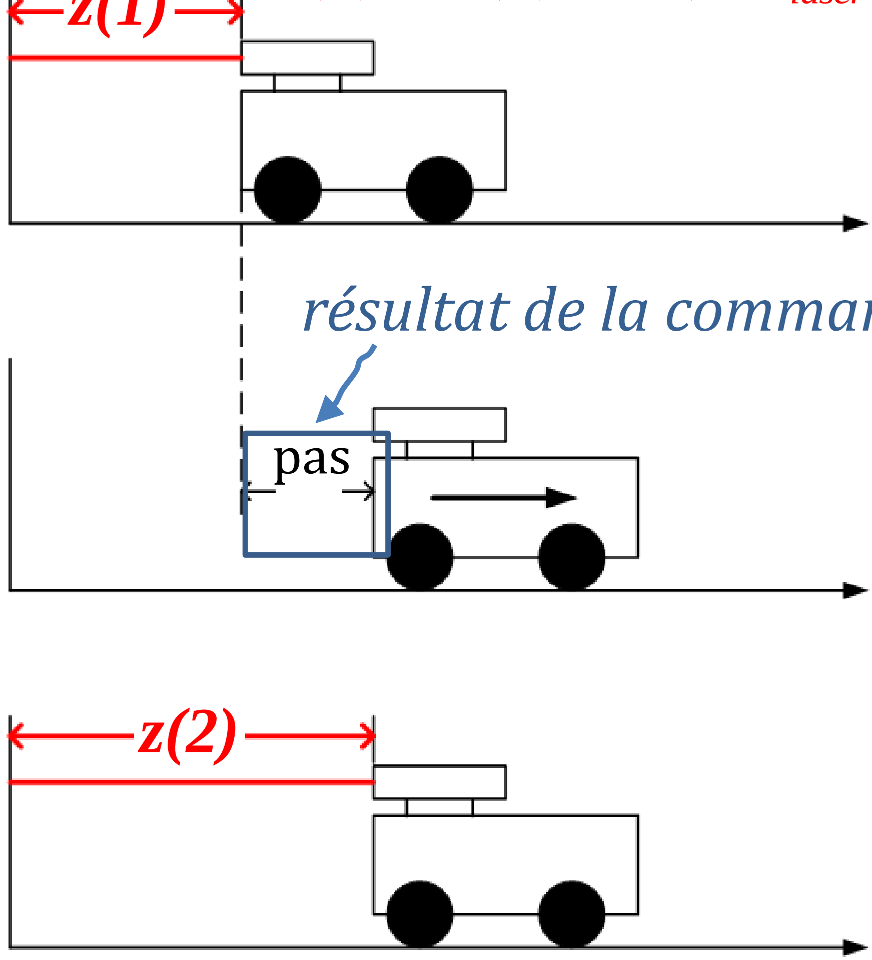
Combiner mesures différents endroits

$$z(k) = x(k) + N(0, \sigma_{laser}^2)$$



Combiner mesures différents endroits

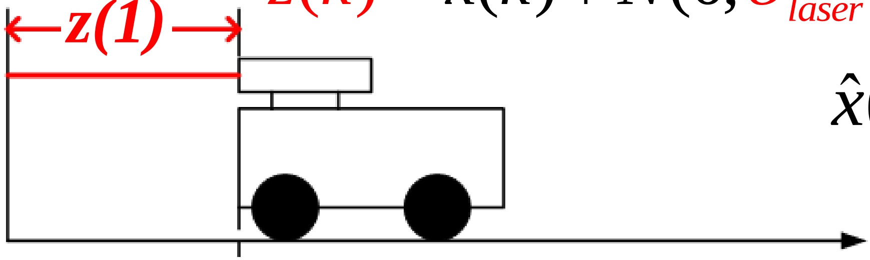
$$z(k) = x(k) + N(0, \sigma^2_{laser})$$



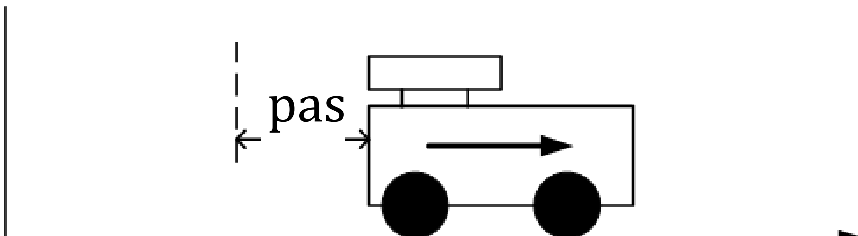
J'ai donc $\mathbf{z}_{1:2}=\{\mathbf{z}(1), \mathbf{z}(2)\}$
et $\boldsymbol{u}_{1:1}=\{\boldsymbol{u}(1)\}$

Combiner mesures différents endroits

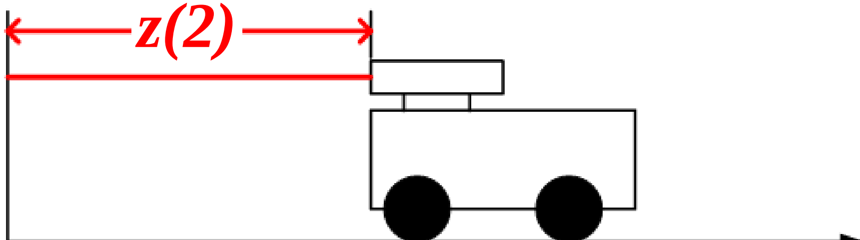
$$z(k) = x(k) + N(0, \sigma^2_{laser})$$



$$\hat{x}(1) = z(1) \qquad P(1) = \sigma^2_{laser}$$



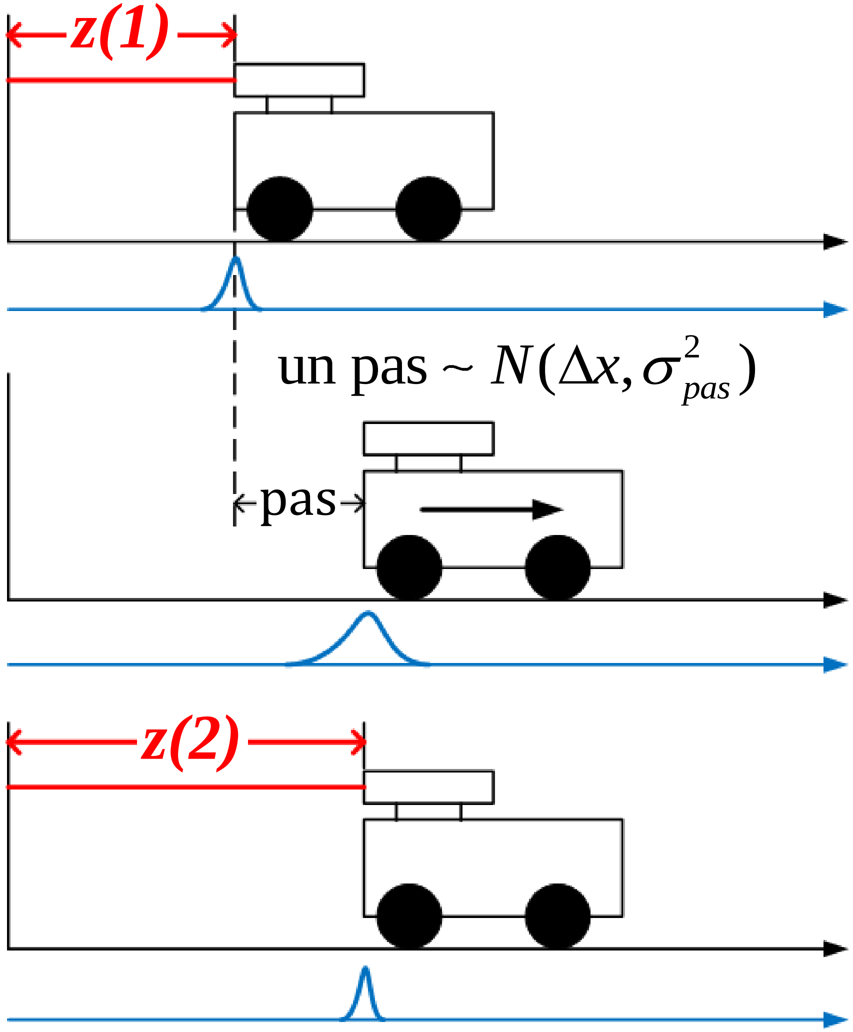
un pas $\sim N(\Delta x, \sigma^2_{pas})$



moyenne récursive

innovation

Combiner mesures différents endroits



$$\hat{x}(1) = z(1) \qquad P(1) = \sigma^2_{laser}$$

avance d'un pas

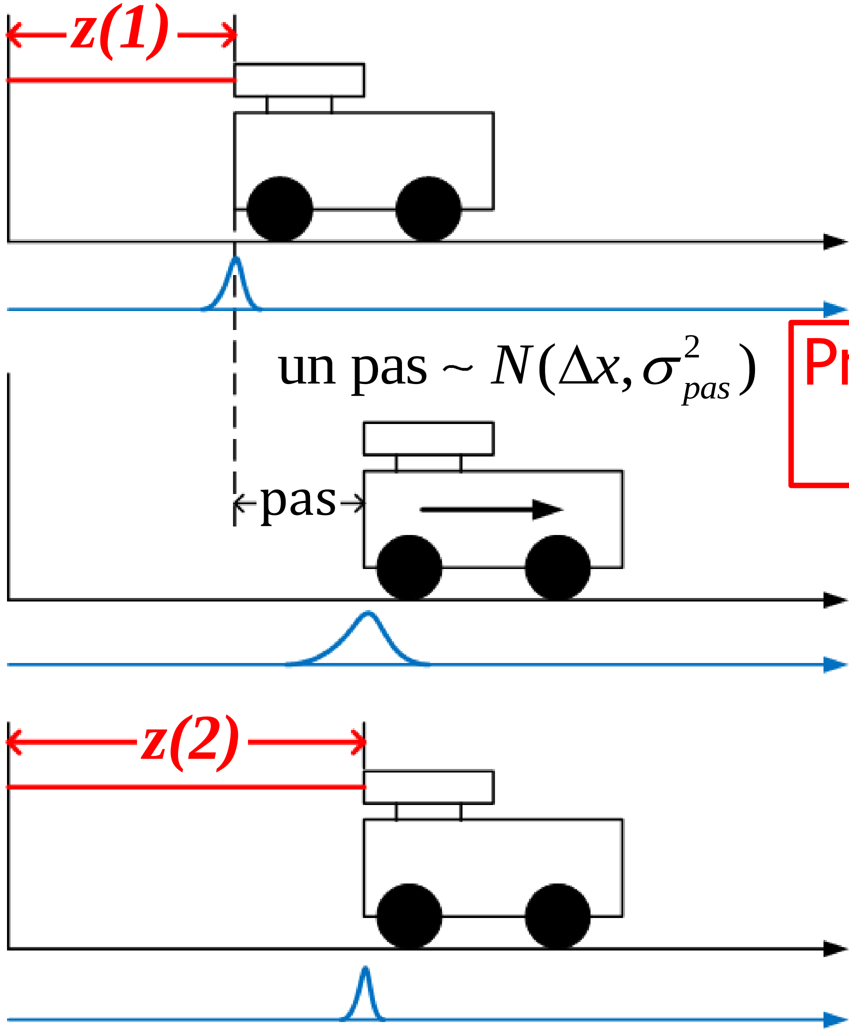
$$\hat{x}(2\,|\,1) = \hat{x}(1) + \Delta x \qquad P(2\,|\,1) = P(1) + \sigma^2_{pas}$$

prend mesure

moyenne récursive

$$\left\{\begin{aligned} w &= \frac{P(2\,|\,1)}{P(2\,|\,1) + \sigma^2_{laser}} \\ \hat{x}(2) &= \hat{x}(2\,|\,1) + w\big(\underbrace{\boxed{z(2) - \hat{x}(2\,|\,1)}}_{\text{innovation}}\big) \\ P(2) &= (1-w)P(2\,|\,1) \end{aligned}\right.$$

Combiner mesures différents endroits



$$\hat{x}(1) = z(1) \qquad P(1) = \sigma^2_{laser}$$

avance d'un pas

Prédiction

$$\hat{x}(2|1) = \hat{x}(1) + \Delta x \qquad P(2|1) = P(1) + \sigma^2_{pas}$$

prend mesure

Mise à jour

moyenne récursive

$$\begin{cases} w = \dfrac{P(2|1)}{P(2|1) + \sigma^2_{laser}} \\ \hat{x}(2) = \hat{x}(2|1) + w(z(2) - \hat{x}(2|1)) \\ P(2) = (1-w)P(2|1) \end{cases}$$

Estimation d'état du robot

- On estime l'endroit le plus probable où le robot se situe à l'instant k:

$$\hat{x}(k)$$

- Il faut aussi la confiance de cet estimé (variance), pour savoir comment incorporer les mesures $\mathbf{z}$:

$$P(k) \approx \mathrm{var}\left\{\hat{x}(k)\right\}$$

Estimation d'état du robot

- Quelles sont les possibles variables d'intérêt ?
  - pose statique du robot : $[x \;\; y \;\; \theta]$
  - état dynamique du robot : $\begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}$
  - certains paramètres d'un capteur qui évoluent lentement dans le temps (biais des gyros)
  - l'endroit et la signature des objets dans l'environnement
- Idéalement l'état est choisi pour être **complet** : propriété Markov
  - rarement vrai, mais constitue une bonne approximation

Estimation d’état :
Filtrage Bayésien

Problème de localisation

- Je suis parti de $\boldsymbol{x_0}$
- J'ai exécuté les commandes de déplacement / mesures odométriques* $\boldsymbol{u_1 \ldots u_t}$
- J'ai pris les mesures $\boldsymbol{z_1 \ldots z_t}$ *(extéroceptives en général)*
- Où suis-je à l'instant *t*? ($\boldsymbol{x_t}$) *(filtrage)*

**les mesures odométriques sont traitées comme des commandes de déplacement*

Problème de localisation

$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$$

$\boldsymbol{bel()}$: *belief* ou croyance (raccourci de notation)

$\boldsymbol{x_t}$: pose du robot au temps présent

$\boldsymbol{u_{1:t}}$: toutes les commandes passées envoyées aux actionneurs

$$\boldsymbol{u_{1:t}}=\{u_1, u_2, \ldots, u_t\}$$

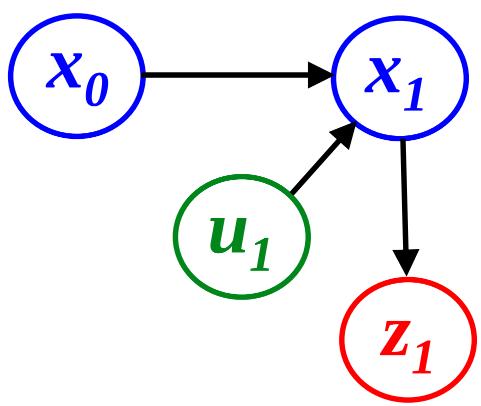
$\boldsymbol{z_{1:t}}$: toutes les mesures passées provenant des capteurs

Simple en apparence, mais imaginez 1000 commandes $\boldsymbol{u_t}$ et 1000 mesures $\boldsymbol{z_t}$... une équation avec au moins 2000 termes?

Solution : ajouter des simplifications!

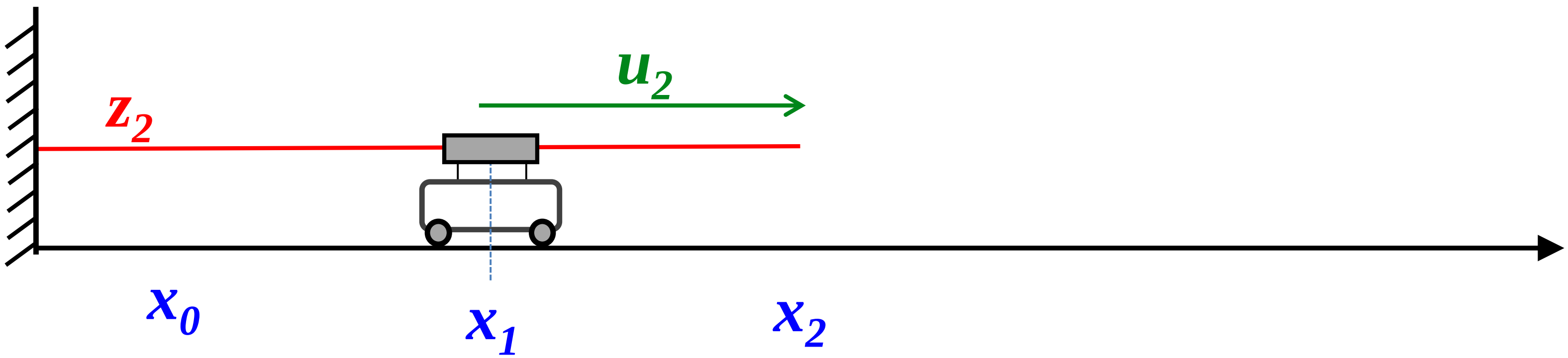
Simplification : État x complet (Markovien)

Graphe des dépendances



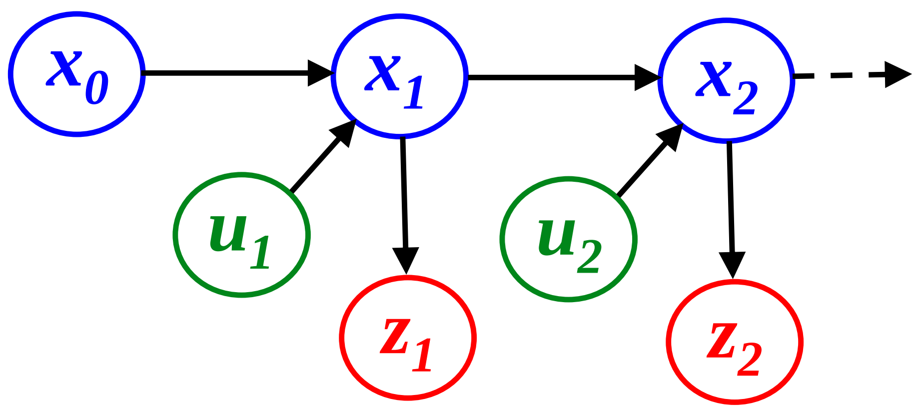
B $\xleftarrow{\text{dépend de...}}$ A

Simplification : État x complet (Markovien)



Ces dépendances vont nous permettre de résoudre $bel(x_t)$ de façon récursive

Graphe des dépendances



et ainsi de suite…

$B \xleftarrow{\text{dépend de...}} A$

Simplification par indépendances des variables

x : état du robot (position, angle, etc.)
u : commandes envoyées aux actionneurs
z : mesures provenant des capteurs

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

motion model/ modèle de déplacement

Incertitude augmente

(après la commande u_t, avant la mesure z_t)

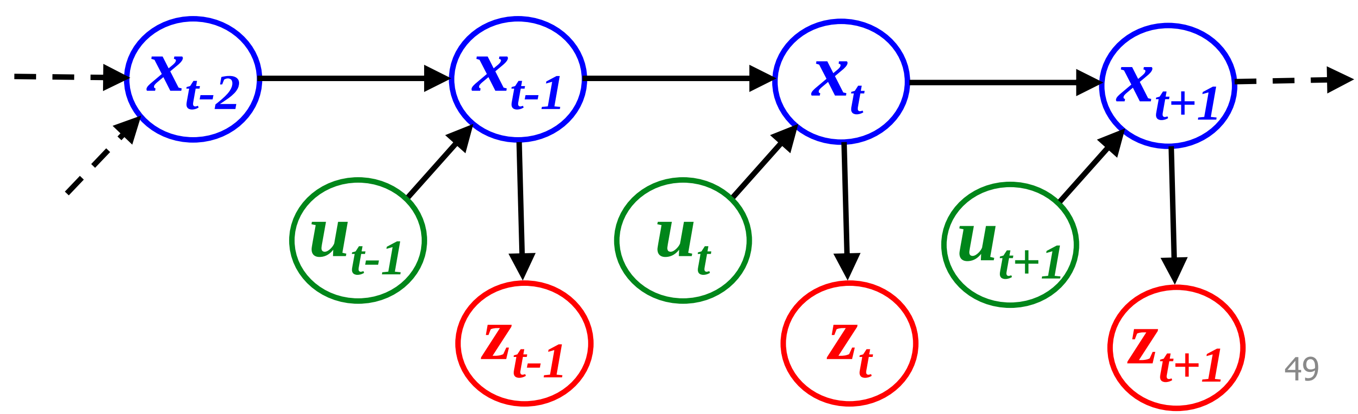
$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

sensor model/ modèle de capteur

Incertitude diminue

(Après tout, les mesures z ne dépendent que de la pose du robot)

$p(A \mid B) = p(A)$
si B n'apporte aucune information sur A



Filtrage Bayésien : formule générique

```
Algorithme FiltreBayes(bel(x_{t-1}), u_t, z_t)
   for all x_t do
```

$$bel_{pred}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$ **(1) Commande u_t**

$$bel(x_t) = \eta \, p(z_t \mid x_t) bel_{pred}(x_t)$$ **(2) Mesure z_t**

```
   endfor
   return bel(x_t)
```

- On doit avoir :
  - *belief* original $bel(x_0)$
  - probabilité de transition d'état : $p(x_t|u_t, x_{t-1})$
  - modèle du capteur $p(z_t \mid x_t)$

$\left.\begin{array}{l} p(x_t|u_t, x_{t-1}) \\ p(z_t \mid x_t) \end{array}\right\}$ comportement du système

On cherche : $bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$

Preuve filtrage Bayésien récursif

Théorème de Bayes : $p(A \mid B, C)\, p(B \mid C) = p(B \mid A, C)\, p(A \mid C)$

(avec extra conditionnelle C)

$$p(x_t \mid z_t, z_{1:t-1}, u_{1:t})\, p(z_t \mid z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t})$$

extrait la dernière mesure

constante, car ne dépend pas de xt

$$bel(x_t) = p(x_t \mid z_t, z_{1:t-1}, u_{1:t}) = \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t})$$

État x_t complet (Markov)

$$p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

$$p(x_t \mid z_t, z_{1:t-1}, u_{1:t}) = \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t})$$

On cherche : $bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t})$

Preuve filtrage Bayésien récursif

Théorème de Bayes : $p(A \mid B, C)\, p(B \mid C) = p(B \mid A, C)\, p(A \mid C)$
(avec extra conditionnelle C)

$$p(x_t \mid z_t, z_{1:t-1}, u_{1:t})\, p(z_t \mid z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t})$$

extrait la dernière mesure

constante, car ne dépend pas de xt

$$p(x_t \mid z_t, z_{1:t-1}, u_{1:t}) = \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t})$$

État x_t complet (Markov)
$p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$

$$bel(x_t) = p(x_t \mid z_t, z_{1:t-1}, u_{1:t}) = \eta\, p(z_t \mid x_t)\, \overbrace{p(x_t \mid z_{1:t-1}, u_{1:t})}^{(2)}$$

Probabilités totales
$p(A \mid C) = \int p(A \mid B, C)\, p(B \mid C)\, dB$

$$p(x_t \mid z_{1:t-1}, u_{1:t}) = \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t})\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1}$$

État x_{t-1} complet (Markov)
$p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$

$$p(x_t \mid z_{1:t-1}, u_{1:t}) = \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1}$$

Théorème d'indépendance
$p(A \mid B) = p(A)$ si A, B indép.

x_{t-1} et u_t sont indépendants si u_t est présumé aléatoire

$$p(x_t \mid z_{1:t-1}, u_{1:t}) = \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})\, dx_{t-1} = \overbrace{\int p(x_t \mid x_{t-1}, u_t)\, bel(x_{t-1})\, dx_{t-1}}^{Bel_{pred}(x_t)\ (1)}$$

Filtrage Bayésien

Algorithme récursif

```
Algorithme FiltreBayes(bel(x_{t-1}), u_t, z_t)
   for all x_t do
```

distribution

Prédiction

$$bel_{pred}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad (1)$$

Mise-à-jour

$$bel(x_t) = \eta\, p(z_t \mid x_t) bel_{pred}(x_t) \quad (2)$$

```
   endfor
   return bel(x_t)
```

bel( ) : *belief* ou croyance (raccourci de notation)

u_t : commande au temps t

z_t : mesure au temps t, après la commande u_t

x_t : pose du robot au temps t, après u_t et z_t

η : facteur de normalisation

Difficile à implémenter <u>exactement</u> car pas toujours de solution analytique de l'intégrale (1)/produit (2)

Quand aura-t-on une solution analytique?

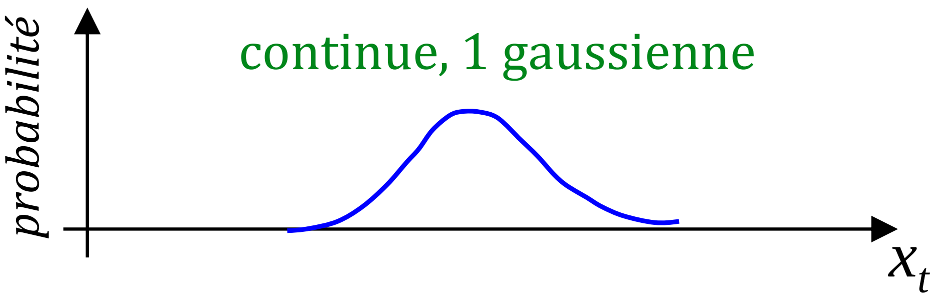
- Les équations à calculer sont

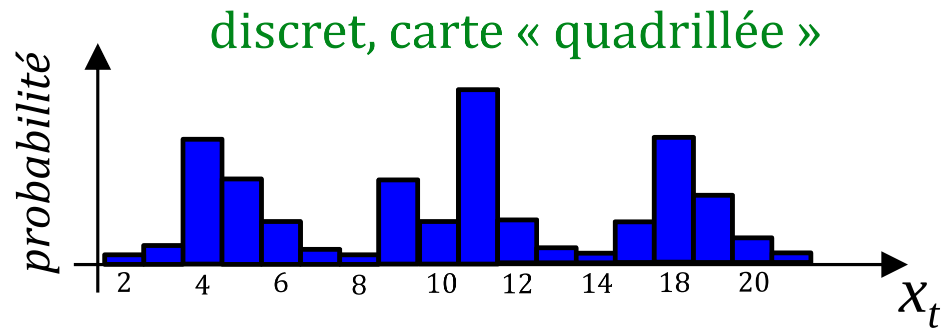
$$\boxed{bel_{pred}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}}$$

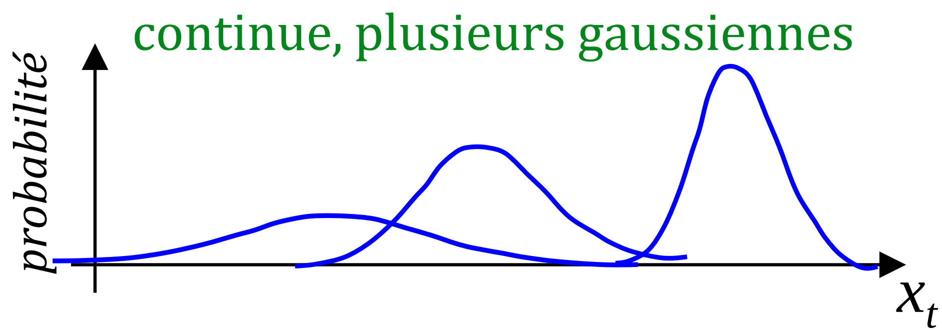
$$\boxed{bel(x_t) = \eta\, p(z_t \mid x_t) bel_{pred}(x_t)}$$

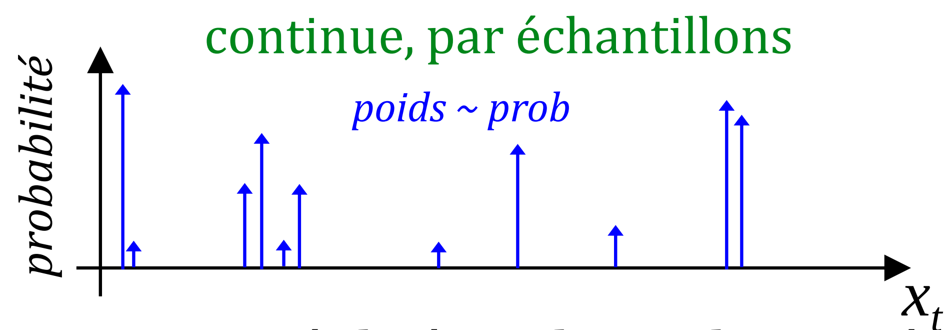
- Solution analytique à ces équations si :
  - *bel*( ) est gaussien
  - les équations du systèmes sont linéaires
  - les bruits sont gaussiens
- Dans les autres cas, il faudra approximer

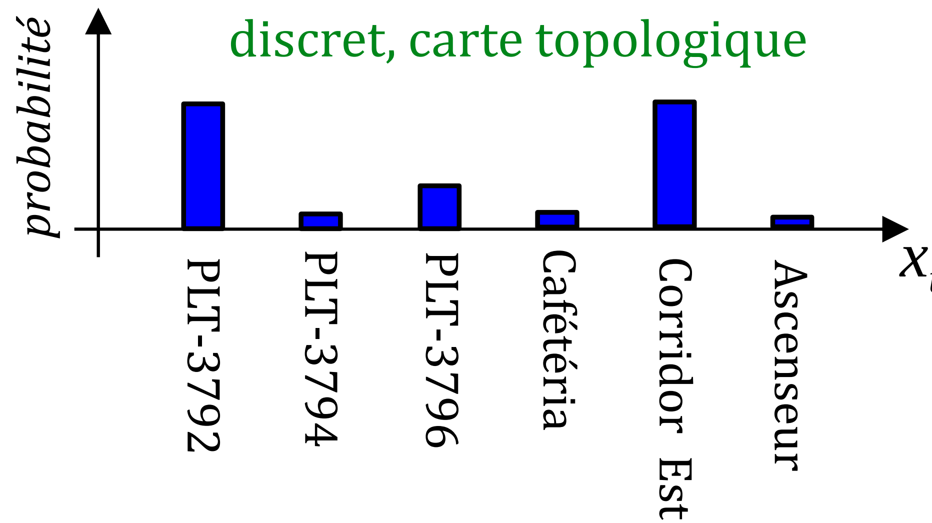
Exemple 1D représentation du *bel*(x_t)











La réalité va dicter la représentation de *Bel*(x_t)
Cette représentation dictera le type de filtre Bayésien utilisé

Différents filtres de Kalman

- Fusion de capteurs
  - combiner évidences de façon optimal

estimés + mesures

concept clef #1

- Fusion de mesures dans le temps
  - moyenne récursive

concept clef #2

- Fusion de capteurs avec déplacement : estimation d'état (pour distributions Gaussiennes)
  - Filtre de Kalman (linéaire)
  - Filtre de Kalman <u>étendu</u> (non-linéaire)
  - Filtre de Kalman <u>non-parfumé</u> (non-linéaire)

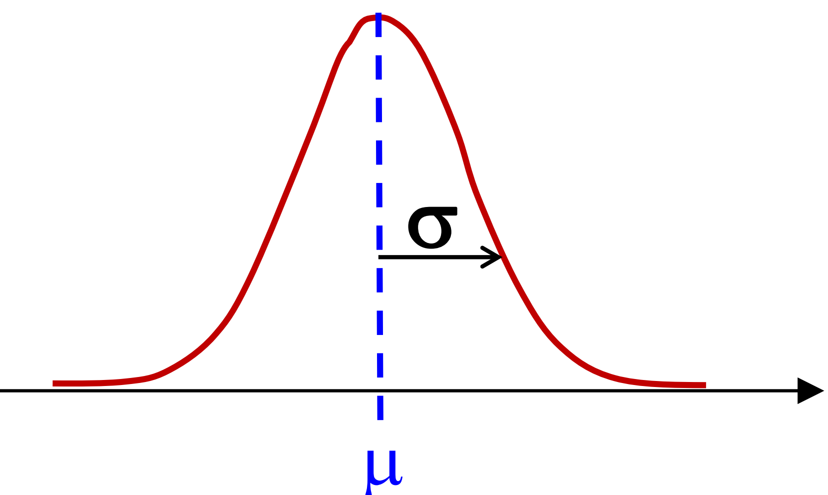
#1+ #2

- Ce sont tous des filtres Bayésiens

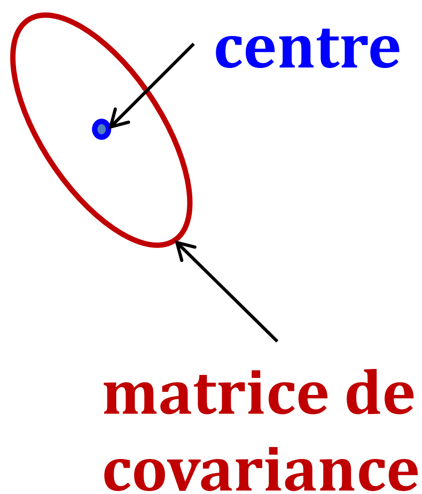
Filtre de Kalman

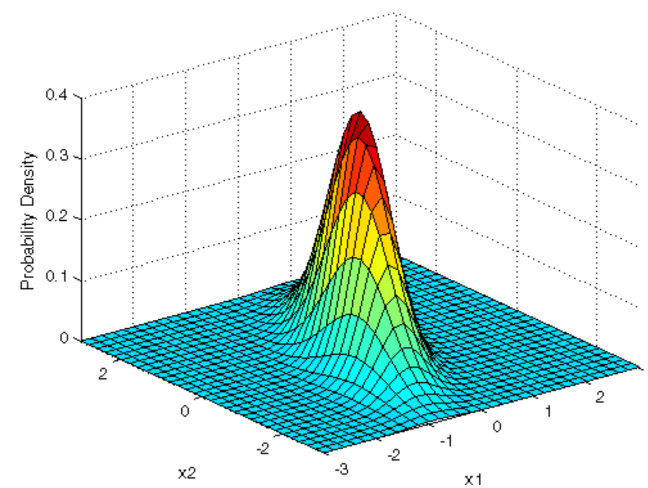
- Utiliser la connaissance de la dynamique du système pour réduire l'impact du bruit
- L'état+incertitude du système seront représentés par des gaussiennes multidimensionnelles

1D



2D





Gaussienne multivariée dimension *k*

$$\boxed{f_X(\vec{x}) = \frac{1}{\sqrt{(2\pi)^k \, |\Sigma|}} \exp\left(-\frac{1}{2} (\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu}) \right)}$$

déterminant

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}, \quad \vec{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_k \end{bmatrix}$$

$$\boxed{\Sigma_{ij} = \mathrm{cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]}$$

matrice carrée des covariances $\Sigma = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & \sigma_{x_1 x_3} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 & \sigma_{x_2 x_3} \\ \sigma_{x_1 x_3} & \sigma_{x_2 x_3} & \sigma_{x_3}^2 \end{bmatrix}$

(symétrique, définie positive)

Soit M une matrice symétrique réelle d'ordre n. Elle est dite **définie positive** si elle vérifie l'une des trois propriétés équivalentes suivantes :

1. Pour toute matrice colonne **non nulle** $\mathbf{x}$ à n éléments réels, on a

   $${}^t\mathbf{x} M \mathbf{x} > 0.$$

   (autrement dit, la forme quadratique définie par M est strictement positive pour $\mathbf{x} \neq 0$)

source : wikipedia

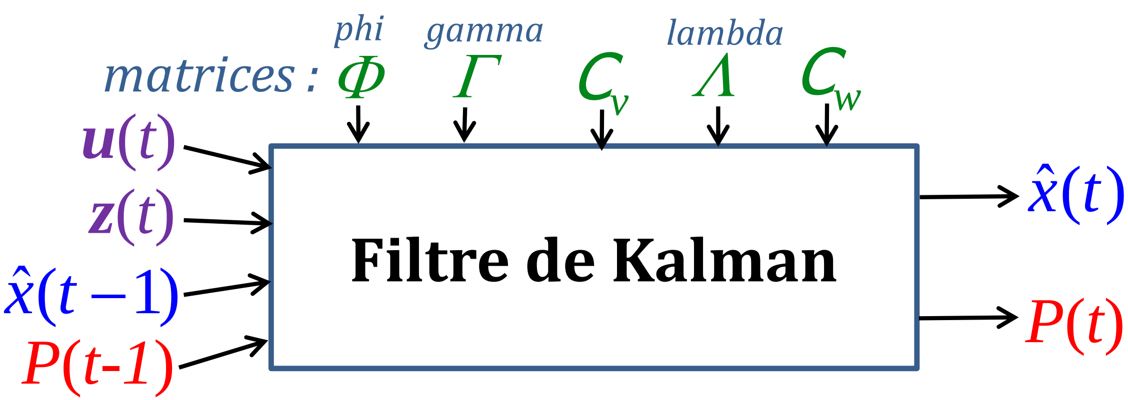
UNIVERSITÉ LAVAL

Filtre Bayésien avec gaussiennes + équations linéaires :

Filtre de Kalman

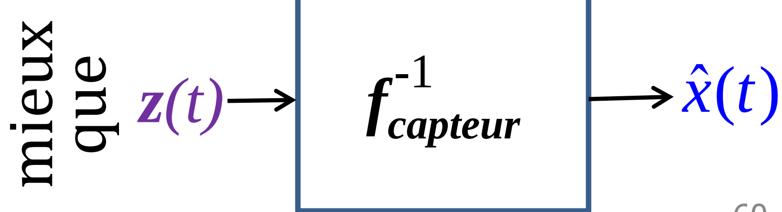
Filtre de Kalman

- « Boîte noire » d'équations *(toujours les mêmes, peu importe le système)*
- Paramètres qui décrivent le comportement du système
- Système dynamique, mesures incomplètes* ou bruitées
- En entrée : commandes $\boldsymbol{u(t)}$ et mesures $\boldsymbol{z(t)}$ + estimé précédent $\hat{x}(t-1)$ et sa covariance estimée $P(t\text{-}1)$
- En sortie : nouvel estimé de la pose $\hat{x}(t)$ + covariance $P(t)$



* Je mesure (observe) x, y mais je veux connaitre θ.

- Pourquoi? Pour avoir $\hat{x}(t)$ plus précis qu'avec seulement $\boldsymbol{z(t)}$



Équations du filtre de Kalman (aperçu)

(Équations matricielles : système linéaire)

(1) $\hat{x}(k+1|k) = \Phi\hat{x}(k) + \Gamma u(k)$

(2) $P(k+1|k) = \Phi P(k)\Phi^T + C_v(k+1)$

(3) $\hat{z}(k+1|k) = \Lambda\hat{x}(k+1|k)$

(4) $r(k+1) = z(k+1) - \hat{z}(k+1|k)$

(5) $K(k+1) = P(k+1|k)\Lambda^T\{\Lambda P(k+1|k)\Lambda^T + C_w(k+1)\}^{-1}$

(6) $\hat{x}(k+1) = \hat{x}(k+1|k) + K(k+1)r(k+1)$

(7) $P(k+1) = (I - K(k+1)\Lambda)P(k+1|k)$

Exécuté à chaque itération k :

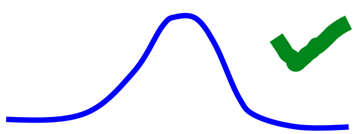
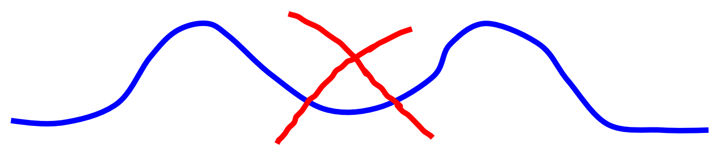
- 1 vecteur commande $u(k)$
- un vecteur de mesures $z(k)$

UNIVERSITÉ LAVAL

Filtre Kalman : estimateur optimal

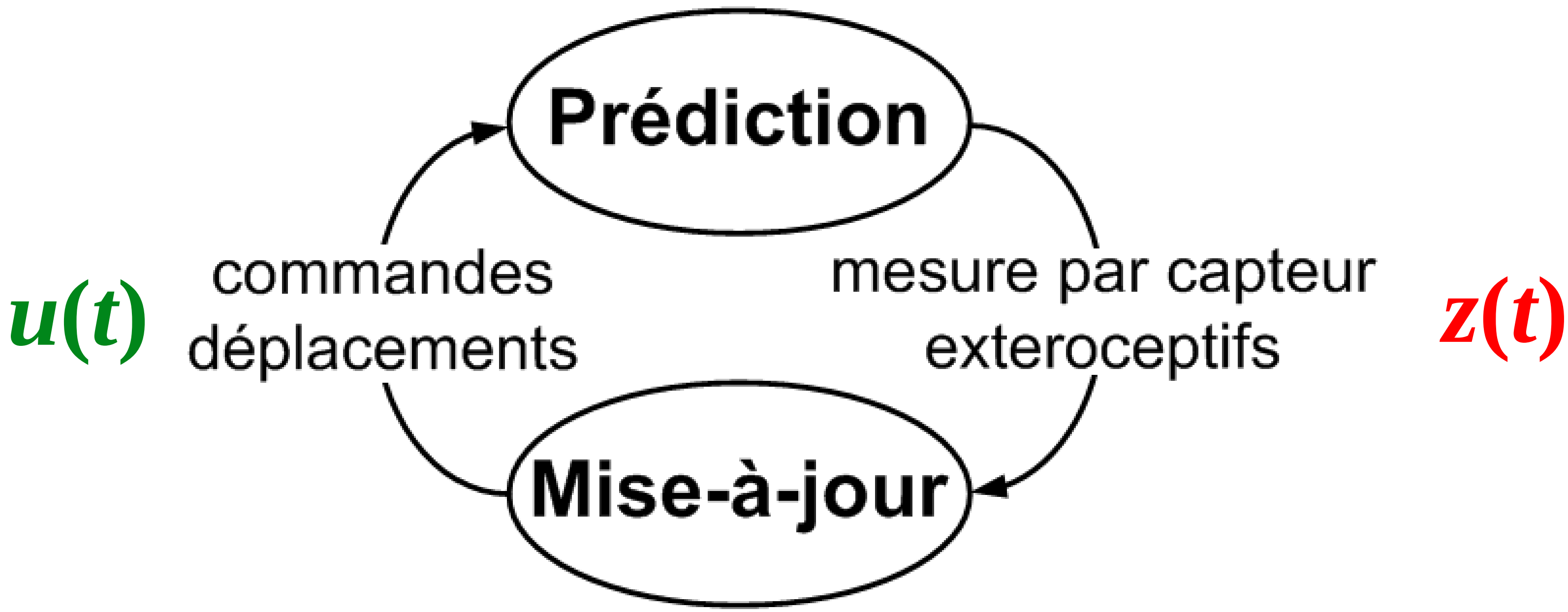
Pré-requis:

- Modèle du système est <u>linéaire</u> $\hat{x}(k+1|k) = \Phi\hat{x}(k) + \Gamma u(k)$
- Modèle du capteur est <u>linéaire</u> $\hat{z}(k+1|k) = \Lambda\hat{x}(k+1|k)$
- Bruits sont gaussiens, avec <u>moyennes nulles</u>. capteur non-biaisé
- État est caractérisé par une moyenne et covariance → distribution unimodale

- Covariance est un bon indicateur de la confiance
- État $x(k)$ est complet : propriété de Markov

Estimation d’état



un tour de la boucle par cycle
déplacement-mesure

Modèle linéaire du système

- Modèle de la dynamique du **système**

Bruit Gaussien

$$x(k+1) = \Phi(k)x(k) + \Gamma(k)\underbrace{u(k)}_{\text{actions}} + v(k)$$

$$v(k) \sim N(0, \sigma_v^2)$$

modèle matriciel de transition d'état

modèle matriciel des commandes

($\Phi(k)$ signifie que Φ peut varier en fonction du temps)

(Ceci n'est pas l'équation du filtre, mais comment le système fonctionne. On n'ajoute pas du bruit dans le filtre. Il en va de même pour les acétates suivantes sur les modèles).

Modèle linéaire du système

- Modèle de la dynamique du **système**

Bruit Gaussien

$$x(k+1) = \Phi(k)x(k) + \Gamma(k)\underbrace{u(k)}_{\text{actions}} + v(k)$$

(erreurs de modélisation, perturbations)

- Exemple

État : $\boldsymbol{x}$=[$\boldsymbol{d}$] (*mètres*)
Commande : $\boldsymbol{u}$=[$\delta_{\boldsymbol{d}}$] (*mètres*)

$$d(k+1) = d(k) + \delta_d(k) + v(k)$$

$$x(k+1) = x(k) + u(k) + v(k)$$

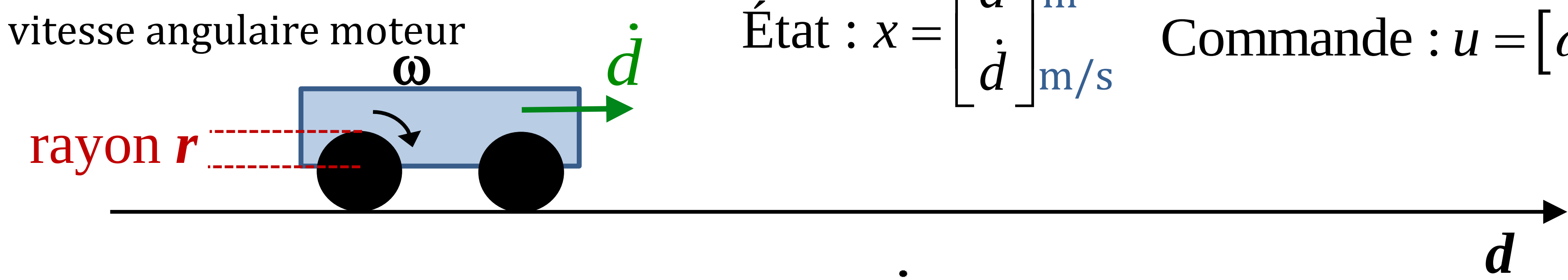
$$\Phi = [1] \qquad \Gamma = [1] \qquad v \sim N(0, \sigma^2_{pas})$$

Modèle linéaire du système (2)

- Modèle dynamique **système**

Bruit Gaussien

$$x(k+1) = \Phi(k)x(k) + \overbrace{\Gamma(k)u(k)}^{\text{actions}} + v(k)$$



État : $x = \begin{bmatrix} d \\ \dot{d} \end{bmatrix} \begin{matrix} \text{m} \\ \text{m/s} \end{matrix}$ Commande : $u = [\omega]$ rad/s

$$d(k+1) = d(k) + \dot{d}(k)\Delta t + v_r(k)$$

roches…

$$\dot{d}(k+1) = r(\omega(k) + v_\omega(k))$$

$$x(k+1) = \begin{bmatrix} d(k+1) \\ \dot{d}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 0 \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} d(k) \\ \dot{d}(k) \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ r \end{bmatrix}}_{\Gamma} \underbrace{[\omega(k)]}_{u} + \begin{bmatrix} v_r(k) \\ r v_\omega(k) \end{bmatrix}$$

Modèle linéaire du capteur

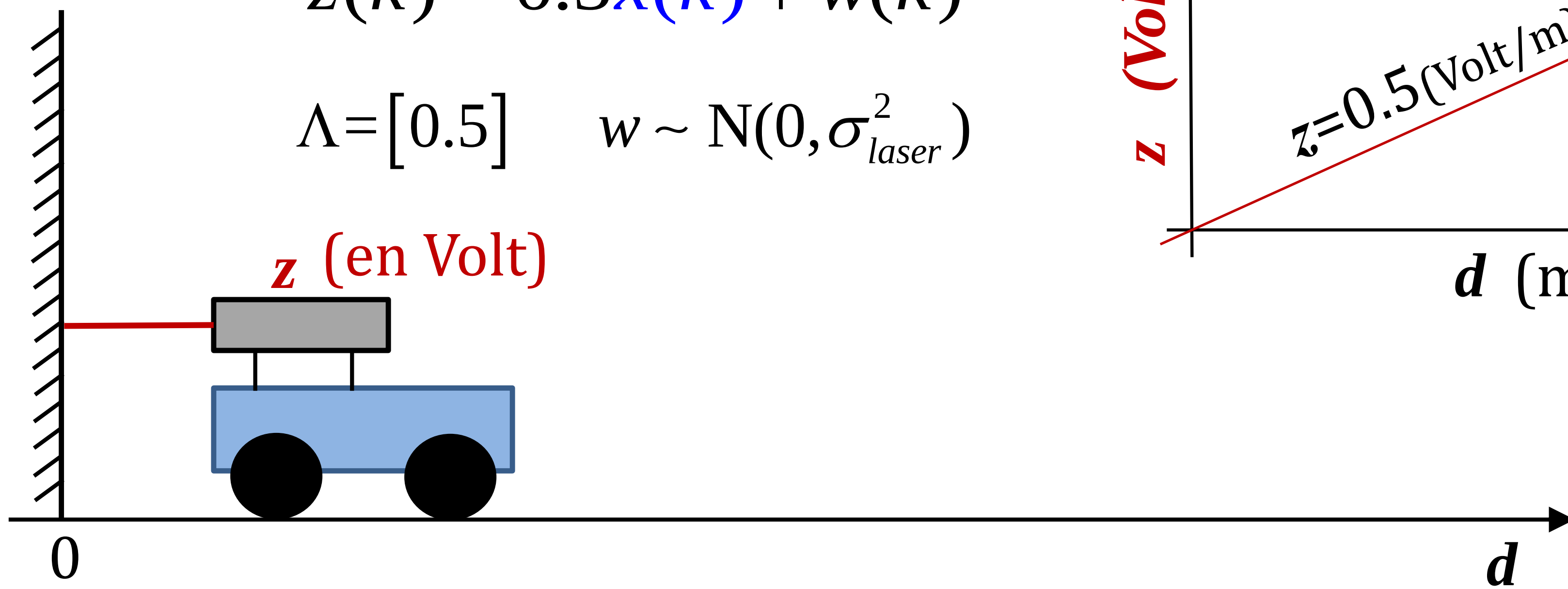
État : $\boldsymbol{x}=[\boldsymbol{d}]$ (*mètres*)

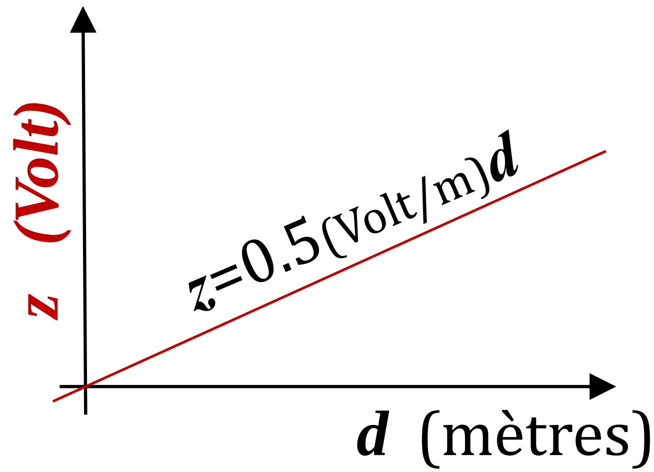
$$z(k) = \Lambda(k)x(k) + w(k)$$

Bruit Gaussien

$$z(k) = 0.5x(k) + w(k)$$

$$\Lambda = [0.5] \qquad w \sim N(0, \sigma^2_{laser})$$



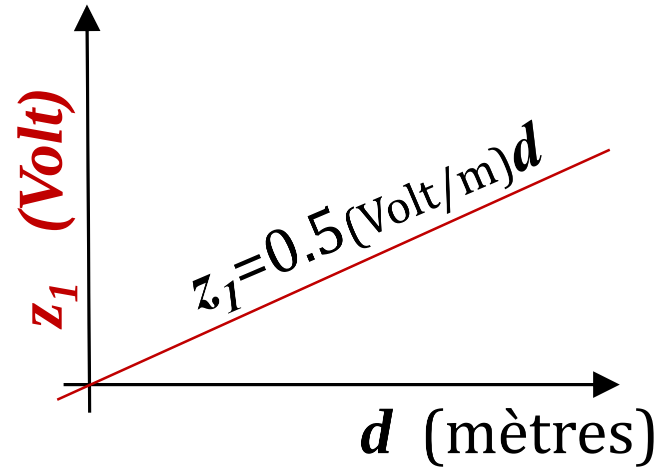


Modèle linéaire du capteur (2)

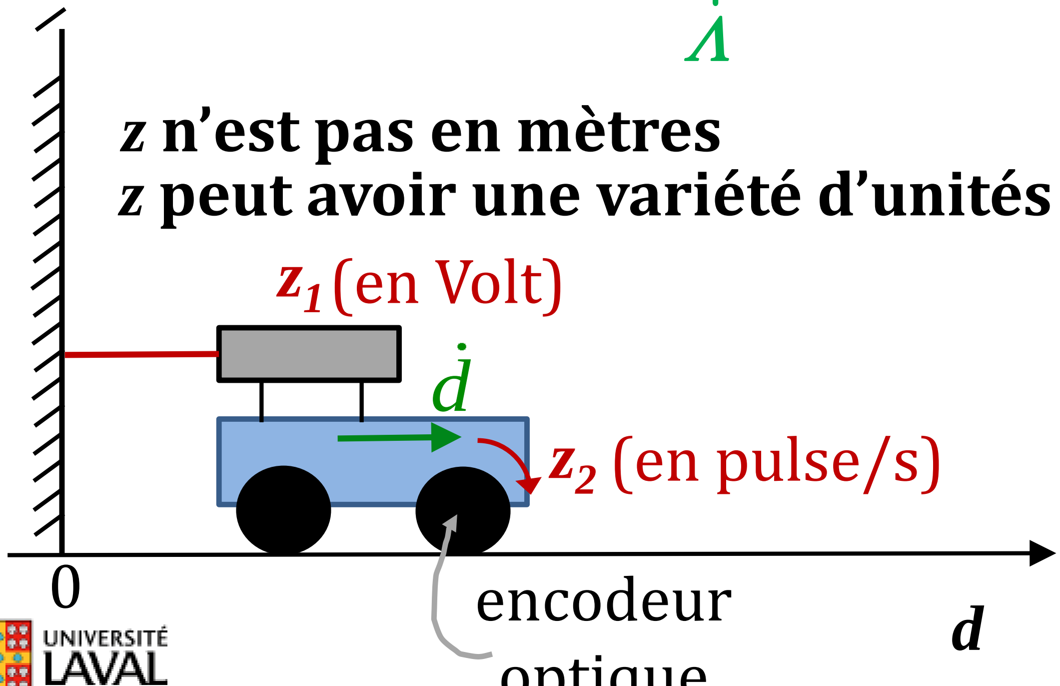
États : $x = \begin{bmatrix} d \\ \dot{d} \end{bmatrix}$

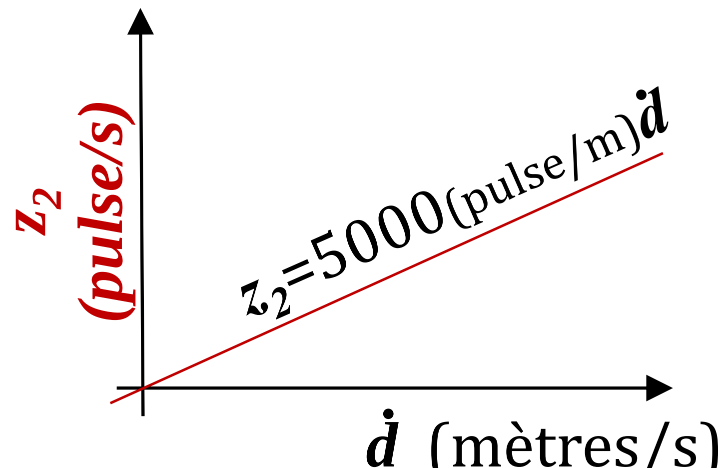
$$z(k) = \Lambda(k)\,x(k) + w(k) \longleftarrow \text{Bruit Gaussien}$$

$$\begin{matrix}\text{(en Volt)} \\ \text{(en pulse/s)}\end{matrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0.5 & 0 \\ 0 & 5000 \end{bmatrix}}_{\Lambda} \begin{bmatrix} d \\ \dot{d} \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$



z n'est pas en mètres
z peut avoir une variété d'unités





Modèle linéaire du capteur (3)

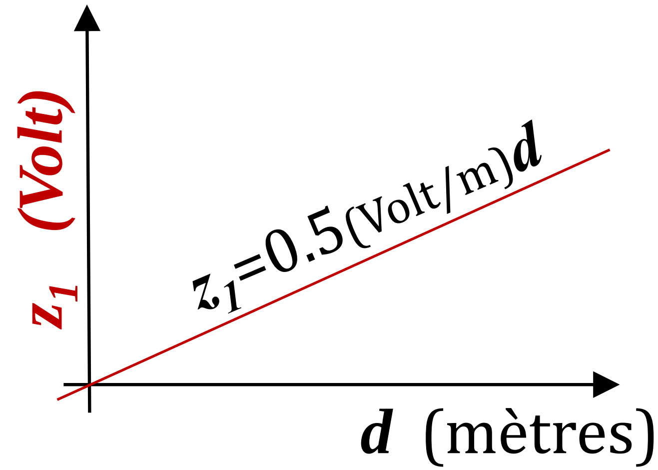
États : $x = \begin{bmatrix} d \\ \dot{d} \end{bmatrix}$

$$z(k) = \Lambda(k)x(k) + w(k)$$

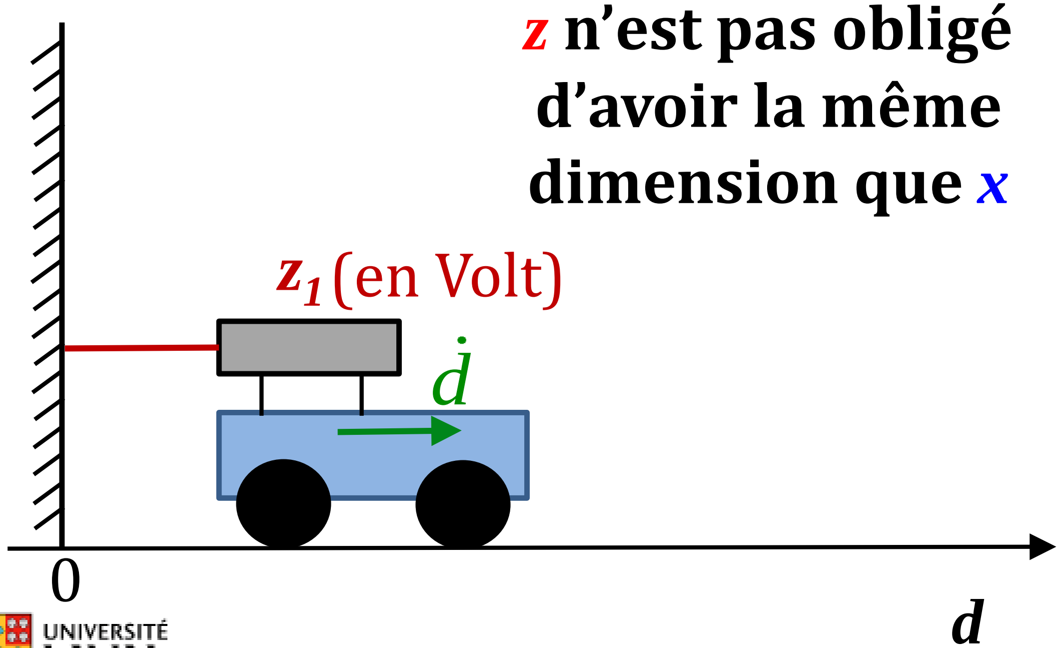
Bruit Gaussien

2 vs. 1

$$[z_1] = [0.5 \quad 0]\begin{bmatrix} d \\ \dot{d} \end{bmatrix} + [w_1]$$



z n'est pas obligé d'avoir la même dimension que x

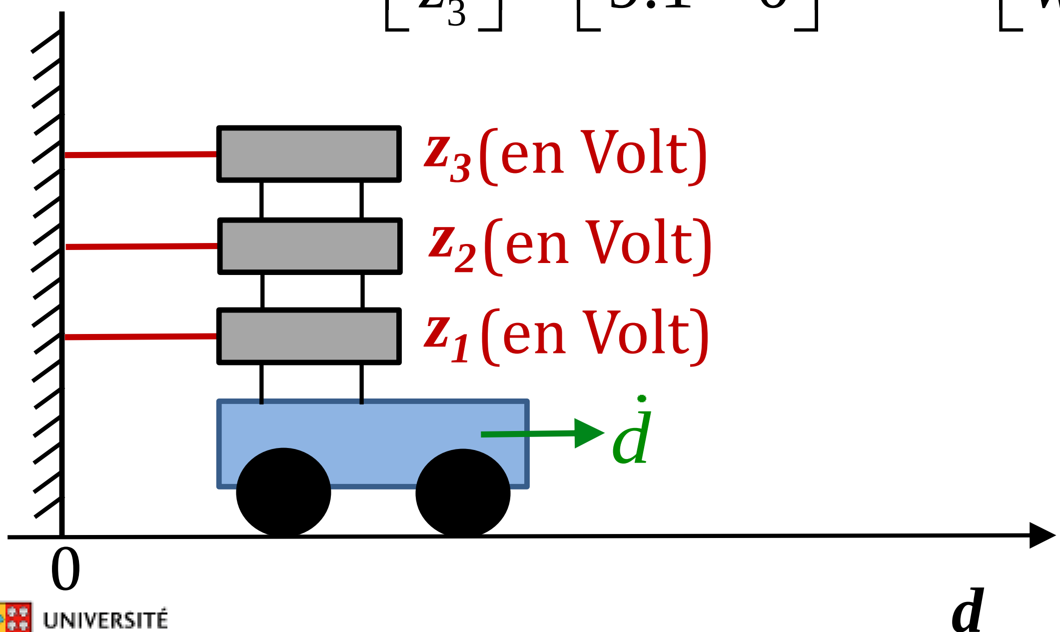


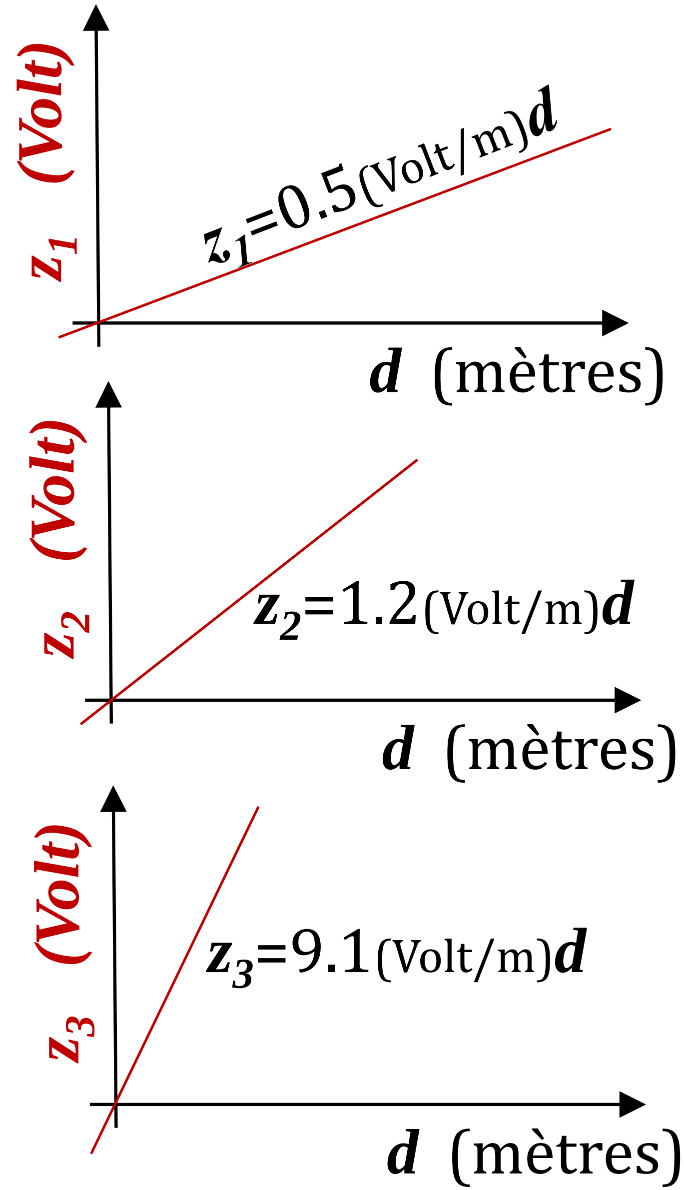
Modèle linéaire du capteur (4)

États : $x = \begin{bmatrix} d \\ \dot{d} \end{bmatrix}$ $\qquad z(k) = \Lambda(k)x(k) + w(k)$

2 vs. 3

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 1.2 & 0 \\ 9.1 & 0 \end{bmatrix} \begin{bmatrix} d \\ \dot{d} \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$





Filtre Kalman

- Ce que l'on cherche à estimer :
  - estimé de la pose au moment k : $\hat{x}(k)$
  - estimé de la covariance de la pose au moment k : $\boldsymbol{P(k)}$

- Ce que l'**on doit connaître** de notre système
  - Variance sur déplacement $C_v(k)$
  - Variance sur capteurs $C_w(k)$

  fourni par le manufacturier / vous le mesurez

  - Modèle transition d'état $\boldsymbol{\Phi}$
  - Modèle des commandes $\boldsymbol{\Gamma}$

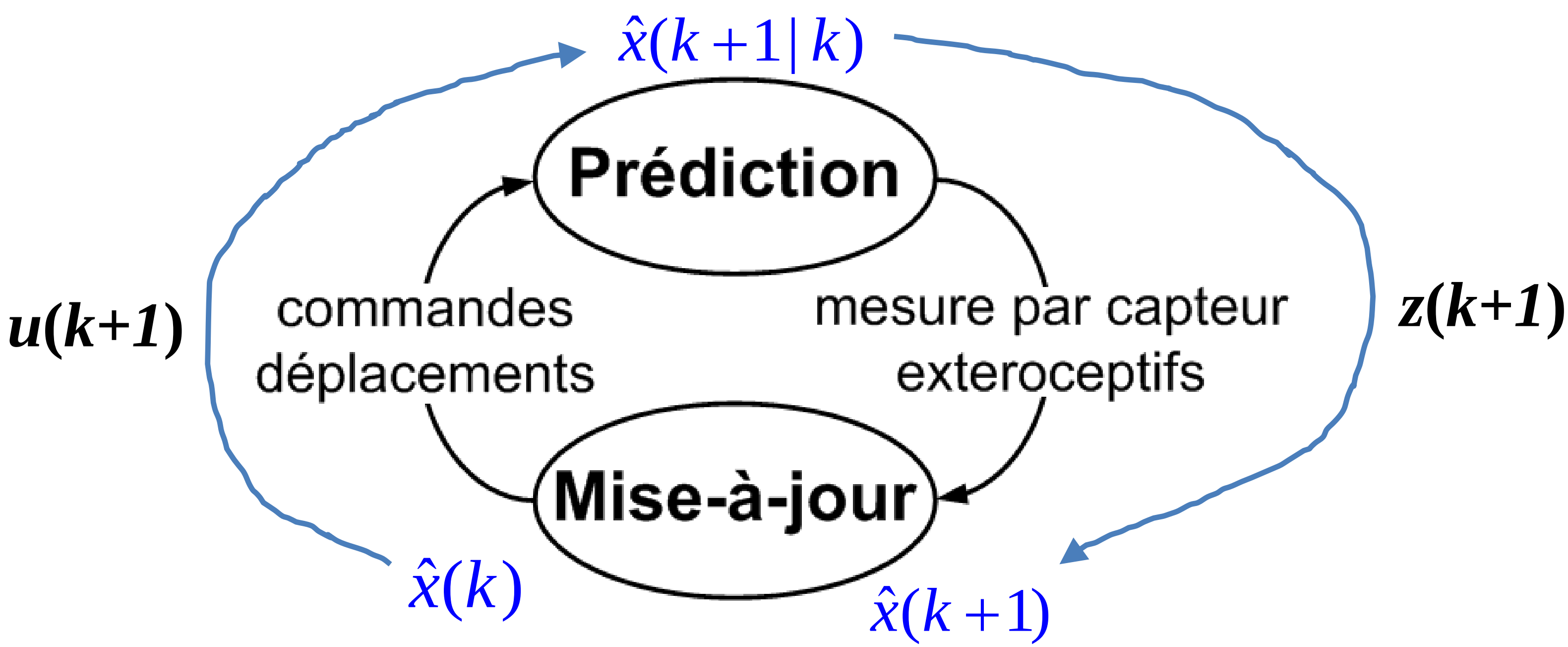
  c'est votre modèle physique du système

  - Modèle du capteur $\boldsymbol{\Lambda}$

  fourni par le manufacturier / calibration manuelle

Notation

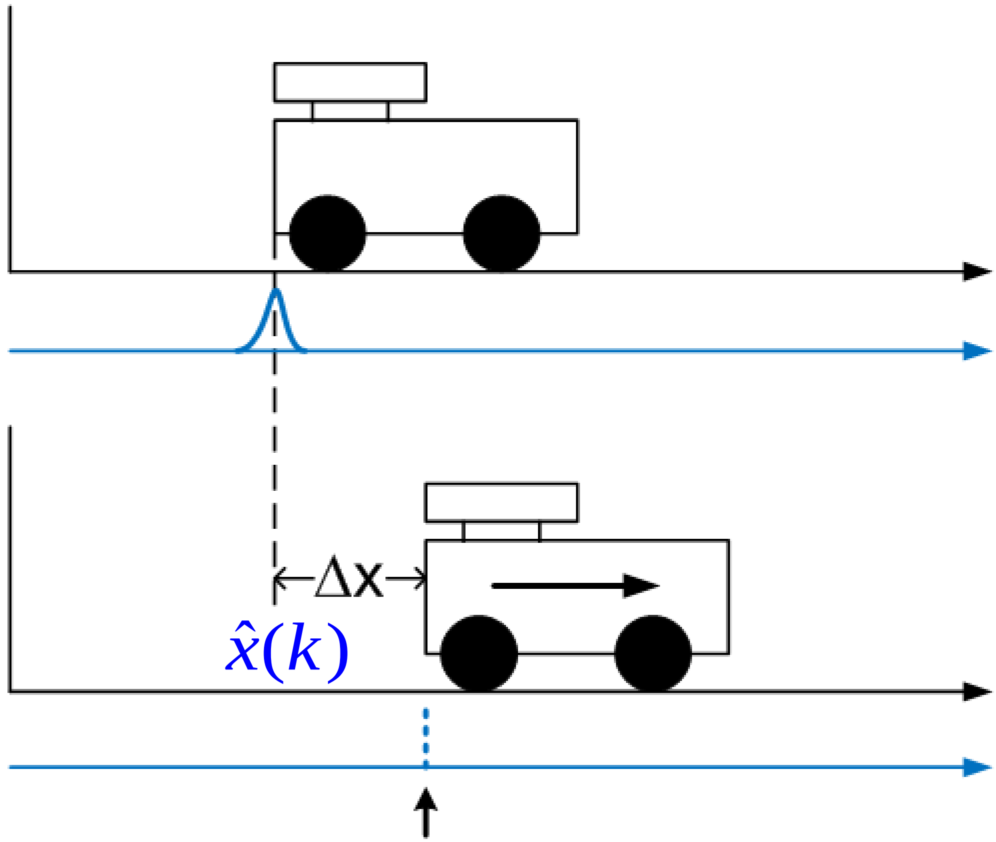
$\hat{x}(k)$	estimé de la pose au moment k
$\hat{x}(k+1\|k)$	estimé de la pose, après déplacement $u(k+1)$, avant mesure $z(k+1)$
$\hat{x}(k+1)$	estimé de la pose, après mesure $z(k+1)$



Filtre Kalman : prédiction sur x

avant mesure $z(k+1)$ $\left\{ \hat{x}(k+1|k) = \Phi(k)\hat{x}(k) + \Gamma(k)u(k) \right.$

cas chariot 1 D



pour $\Phi(k) = [1], \quad \Gamma(k) = [1] \;\Rightarrow\; \hat{x}(k+1|k) = \hat{x}(k) + \Delta x$

Filtre Kalman générique : prédiction sur P

- Variance estimée sur la prédiction $\hat{x}(k)$:

$$P(k) = E\{(\hat{x}(k) - x(k))(\hat{x}(k) - x(k))^T\}$$

Position réelle (et toujours inconnue)

$$\boxed{P(k+1|k) = \Phi(k)P(k)\Phi(k)^T + \underbrace{C_v(k)}}$$

Bruit sur déplacement,
perturbations

- Par exemple, 1 dimension et Φ=[1]

$$P(k+1|k) = \Phi P(k)\Phi^T + C_v = P(k) + C_v$$

- Par exemple, 1 dimension et Φ=[a]

$$P(k+1|k) = \Phi P(k)\Phi^T + C_v = aP(k)a + C_v = a^2 P(k) + C_v$$

Prédiction sur P : véhicule accéléré

Modèle physique : $y_t = y_{t-1} + \dot{y}_{t-1}\Delta T + \frac{1}{2}\ddot{y}_{t-1}\Delta T^2 \qquad \dot{y}_t = \dot{y}_{t-1} + \ddot{y}_{t-1}\Delta T$

Modèle linéaire Kalman : $x_t = \Phi x_{t-1} + \Gamma u$

État : $x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$ Commande : $u = [\ddot{y}]$ $\Phi = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$ $\Gamma = \begin{bmatrix} \frac{1}{2}\Delta T^2 \\ \Delta T \end{bmatrix}$

$$\boxed{P(k+1|k) = \Phi P(k) \Phi^T + C_v}$$

Initialisation : $P(1) = \begin{bmatrix} 400 & 0 \\ 0 & 16 \end{bmatrix}$ $\Delta T = 0.5$

Incertitude sur position augmente

$$\Phi P(1)\Phi^T = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 400 & 0 \\ 0 & 16 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 404 & 8 \\ 8 & 16 \end{bmatrix}$$

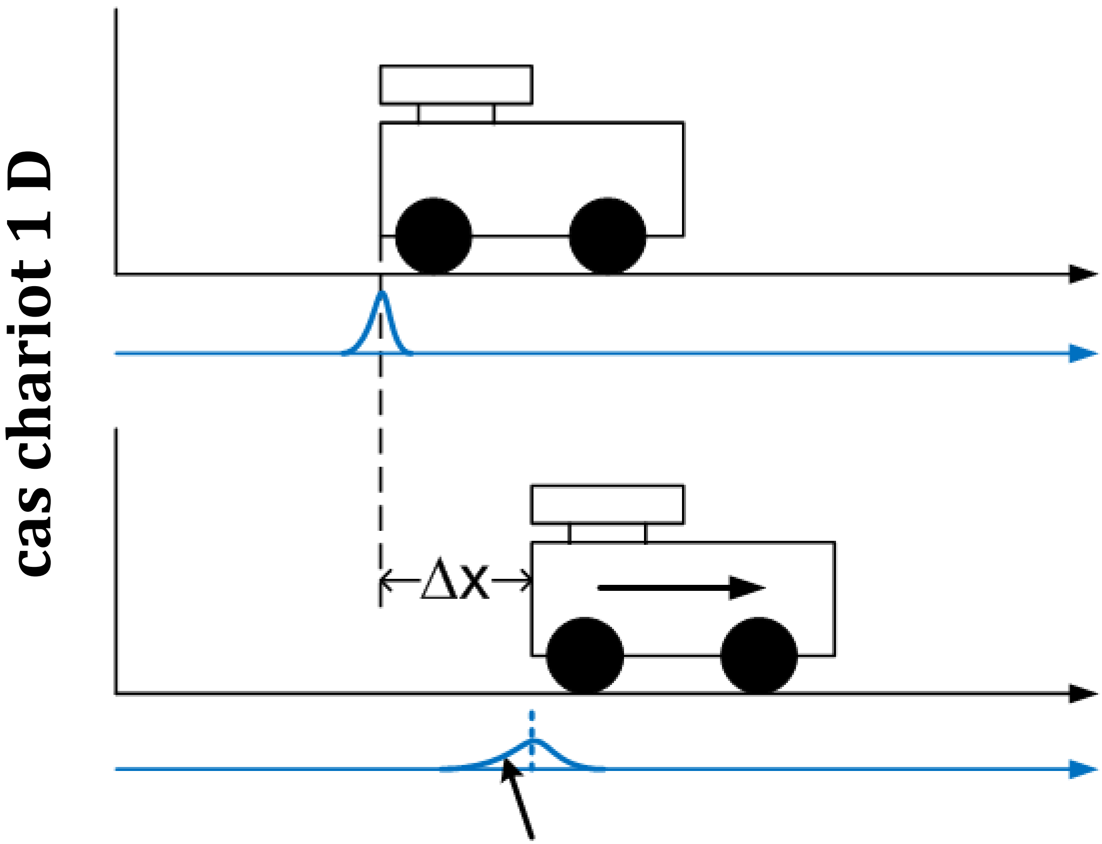
Incertitude sur vitesse inchangée

Ajoute $C_v = \begin{bmatrix} {}^v\sigma_y^2 & {}^v\sigma_{y\dot{y}} \\ {}^v\sigma_{y\dot{y}} & {}^v\sigma_{\dot{y}}^2 \end{bmatrix} = \begin{bmatrix} {}^v\sigma_y^2 & 0 \\ 0 & {}^v\sigma_{\dot{y}}^2 \end{bmatrix}$ (Déterminée expérimentalement)

Incertitude P augmente partout

Filtre Kalman : prédiction sur P

$$P(k+1|k) = \Phi(k)P(k)\Phi(k)^T + C_v(k)$$



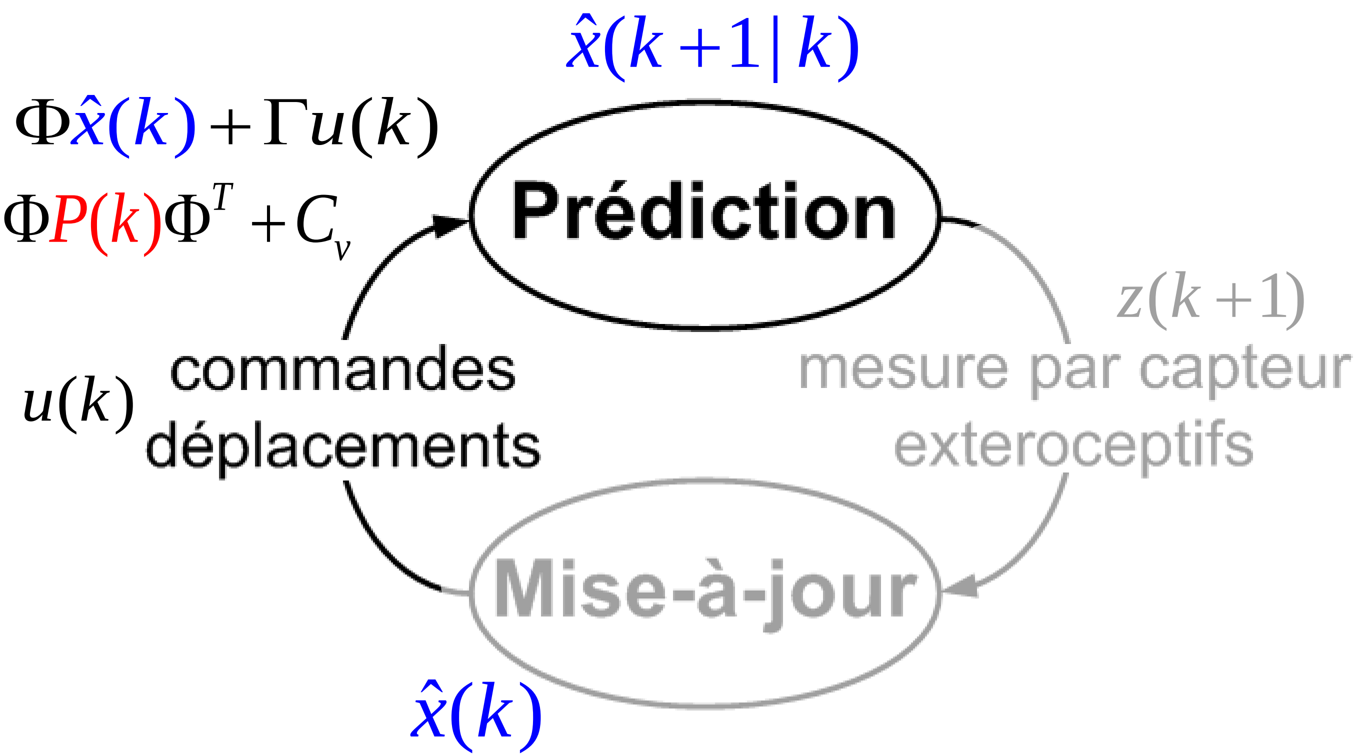
$$\Phi(k) = [1] \;\Rightarrow P(k) + \sigma^2{}_{pas}$$

Estimation d'état : prédiction

note : pour alléger la notation, $\Phi=\Phi(k)$, $\Gamma=\Gamma(k)$

$$\hat{x}(k+1|k)=\Phi\hat{x}(k)+\Gamma u(k)$$

$$P(k+1|k)=\Phi P(k)\Phi^T+C_v$$

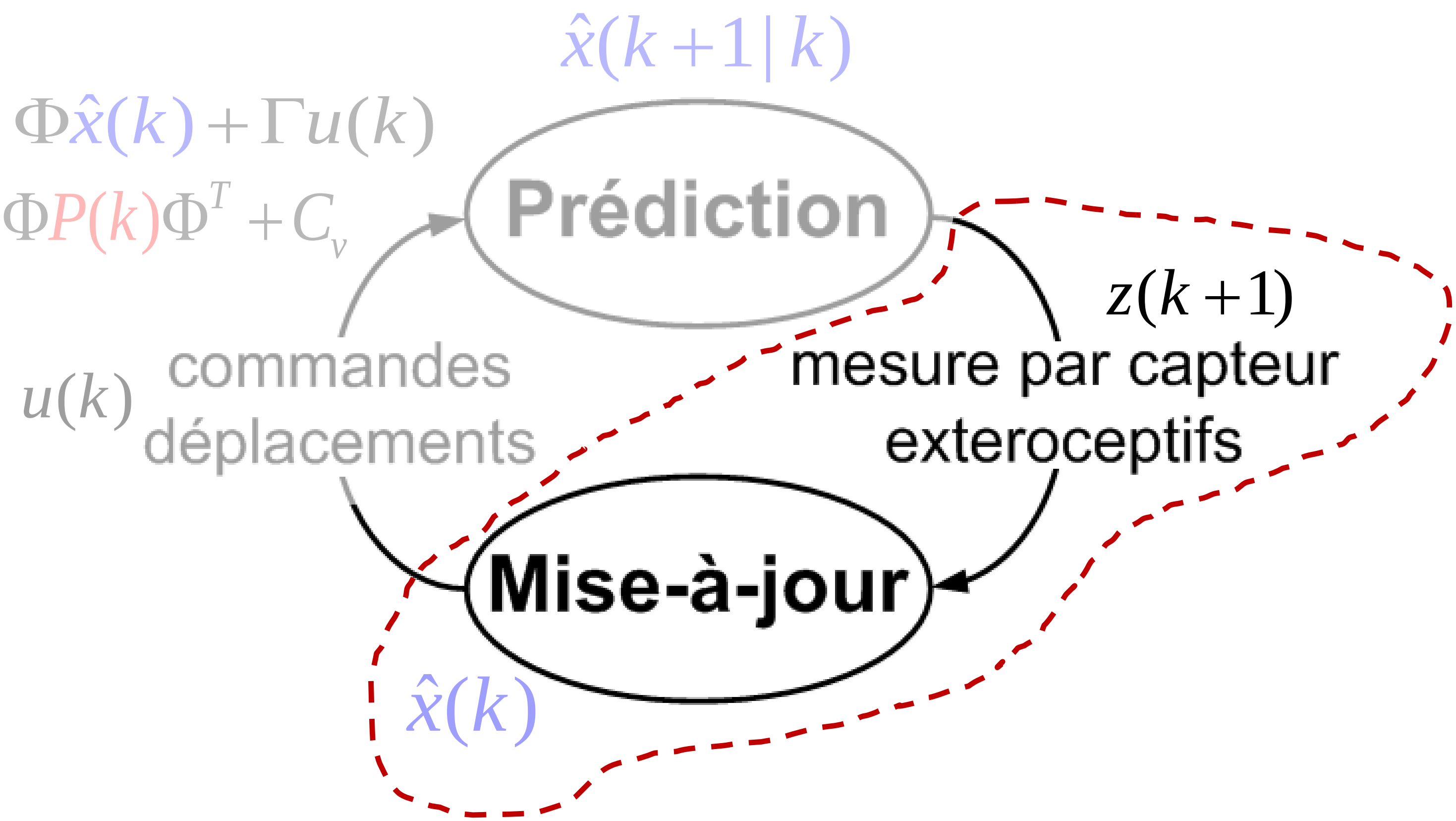


Estimation d'état : mise-à-jour

note : pour alléger la notation, $\Phi=\Phi(k)$, $\Gamma=\Gamma(k)$

$$\hat{x}(k+1|k) = \Phi\hat{x}(k) + \Gamma u(k)$$

$$P(k+1|k) = \Phi P(k)\Phi^T + C_v$$



Filtre Kalman : prédire mesure z

- Prédire la mesure du capteur :

$$\hat{z}(k+1|k) = \overbrace{\Lambda(k+1)}^{\text{fonction linéaire du capteur}} \underbrace{\hat{x}(k+1|k)}_{\text{État que l'on pense être}}$$

- Compare avec mesure prise $z(k+1)$: **innovation *r***

$$r(k+1) = \underbrace{z(k+1)}_{\text{mesure obtenue}} - \underbrace{\hat{z}(k+1|k)}_{\text{mesure prédite}}$$

Calcul poids optimal : gain K

- K est similaire à w_{opt} vu précédemment
- Permet de combiner de <u>façon optimale</u> l'estimé de position $\hat{x}(k+1|k)$ avec l'innovation $r(k+1)$, en tenant compte de :
  - estimé de la covariance de l'estimé de la pose : $P(k+1|k)$
  - covariance de la mesure : matrice $C_w(k+1)$

$$\boxed{K(k+1) = P(k+1|k)\Lambda(k+1)^T\{\Lambda(k+1)P(k+1|k)\Lambda(k+1)^T + C_w(k+1)\}^{-1}}$$

exemple : $\Lambda(k+1) = [1], \quad C_w = \sigma^2_{laser}$

(voir moyenne récursive)

$$K(k+1) = P(k+1|k)[1]^T\{[1]P(k+1|k)[1]^T + \sigma^2_{laser}\}^{-1} = \frac{P(k+1|k)}{P(k+1|k)+\sigma^2_{laser}}$$

UNIVERSITÉ LAVAL

Calcul poids optimal : gain K

- Comparaison faite dans l'espace des mesures (via Λ), et non pas dans l'espace de l'état

$$K(k+1) = P(k+1|k)\Lambda(k+1)^T\{\underbrace{\Lambda(k+1)P(k+1|k)\Lambda(k+1)^T}_{\text{« passe » la covariance sur l'état à travers la fonction de capteur}} + \underbrace{C_w(k+1)}_{\text{covariance des bruits dans les unités des capteurs}}\}^{-1}$$

- **On n'inverse donc jamais la fonction de capteur**
- *(Avantage : Si la fonction de capteur n'est pas bijective, on ne peut pas l'inverser. Dans ce cas-ci linéaire, ce ne sera jamais un problème. Mais pour des cas non-linéaires (EKF) cela pourrait arriver : capteur infrarouge!)*

Combiner innovation + estimé

$$\underbrace{\hat{x}(k+1)}_{\text{Estimé après mesure}} = \underbrace{\hat{x}(k+1|k)}_{\text{Estimé avant mesure}} + \underbrace{K(k+1)}_{\text{gain Kalman : matrice distribue les corrections}} \underbrace{r(k+1)}_{\text{innovation : information du capteur}}$$

Cas à une dimension : $0 \leq K \leq 1$

Fait entièrement confiance à l'estimé $P << C_w$

Fait entièrement confiance à la mesure $P >> C_w$

Espace des mesures vs. espace d'état

calcul du gain : $K(k+1) = P(k+1|k)\Lambda^T\{\Lambda P(k+1|k)\Lambda^T \;+\; C_w\}^{-1}$

$\hat{x}(k+1|k) = 4.3\ \mathrm{m}$

$P = (0.2m)^2 = 0.04\ \mathrm{m}^2$

$\Lambda = 3\ \mathrm{V/m}$

$\sigma_{\mathrm{V}} = 0.3\ \mathrm{V}$

$z(k) = 13.8\ \mathrm{V}$ (x=4.6 m)

$K(k+1) = 0.04\mathrm{m}^2 \times 3\ \mathrm{V/m}\{3\ \mathrm{V/m} \times 0.04\ \mathrm{m}^2 \times 3\ \mathrm{V/m} \;+\; 0.09\ \mathrm{V}^2\}^{-1}$

$K(k+1) = 0.2667\ \mathrm{m/V}$

prédiction de mesure :

$\hat{z}(k+1|k) = \Lambda\hat{x}(k+1|k) = 3\ \mathrm{V/m} \times 4.3\ \mathrm{m} = 12.9\ \mathrm{V}$

innovation :

$r(k+1) = z(k+1) - \hat{z}(k+1|k) = 13.8\ \mathrm{V} - 12.9\ \mathrm{V} = 0.9\ \mathrm{V}$

mise-à-jour:

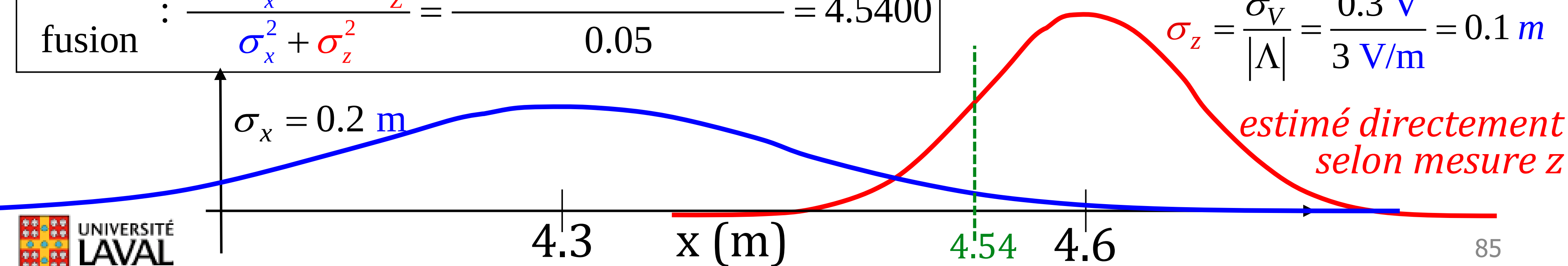
$\hat{x}(k+1) = \hat{x}(k+1|k) + K(k+1)r(k+1) = 4.3\ \mathrm{m} + 0.2667\ \mathrm{m/V} \times 0.9\ \mathrm{V} = 4.540\ \mathrm{m}$

Rapporté dans l'espace de l'état

même chose

méthode fusion : $\dfrac{4.6\sigma_x^2 + 4.3\sigma_Z^2}{\sigma_x^2 + \sigma_z^2} = \dfrac{4.6 \times 0.04 + 4.3 \times 0.01}{0.05} = 4.5400$

$\sigma_z = \dfrac{\sigma_{\mathrm{V}}}{|\Lambda|} = \dfrac{0.3\ \mathrm{V}}{3\ \mathrm{V/m}} = 0.1\ m$



Ajuster la covariance *P*

- Nous avons **acquis** de la précision sur la pose

- La covariance ***P*** va **nécessairement** diminuer

$$P(k+1) = (I - K(k+1)\Lambda(k+1))P(k+1|k)$$

Équations du filtre de Kalman

- Prédiction (lors du déplacement)

(1) $\hat{x}(k+1|k) = \Phi\hat{x}(k) + \Gamma u(k)$ déplacement du robot

(2) $P(k+1|k) = \Phi P(k)\Phi^T + C_v$ augmentation covariance P, après déplacement

- Mise-à-jour (mesure par capteur extéroceptif)

(3) $\hat{z}(k+1|k) = \Lambda\hat{x}(k+1|k)$ prédire ce que je devrait mesurer

(4) $r(k+1) = z(k+1) - \hat{z}(k+1|k)$ compare avec mesure : innovation $r(k+1)$

(5) $K(k+1) = P(k+1|k)\Lambda^T\{\Lambda P(k+1|k)\Lambda^T + C_w(k+1)\}^{-1}$ gain optimal

(6) $\hat{x}(k+1) = \hat{x}(k+1|k) + K(k+1)r(k+1)$ combine estimé pose + mesure

(7) $P(k+1) = (I - K(k+1)\Lambda)P(k+1|k)$ ajuste covariance P, après gain info

note : pour alléger la notation, $\Phi=\Phi(k)$, $\Gamma=\Gamma(k)$, $\Lambda=\Lambda(k)$