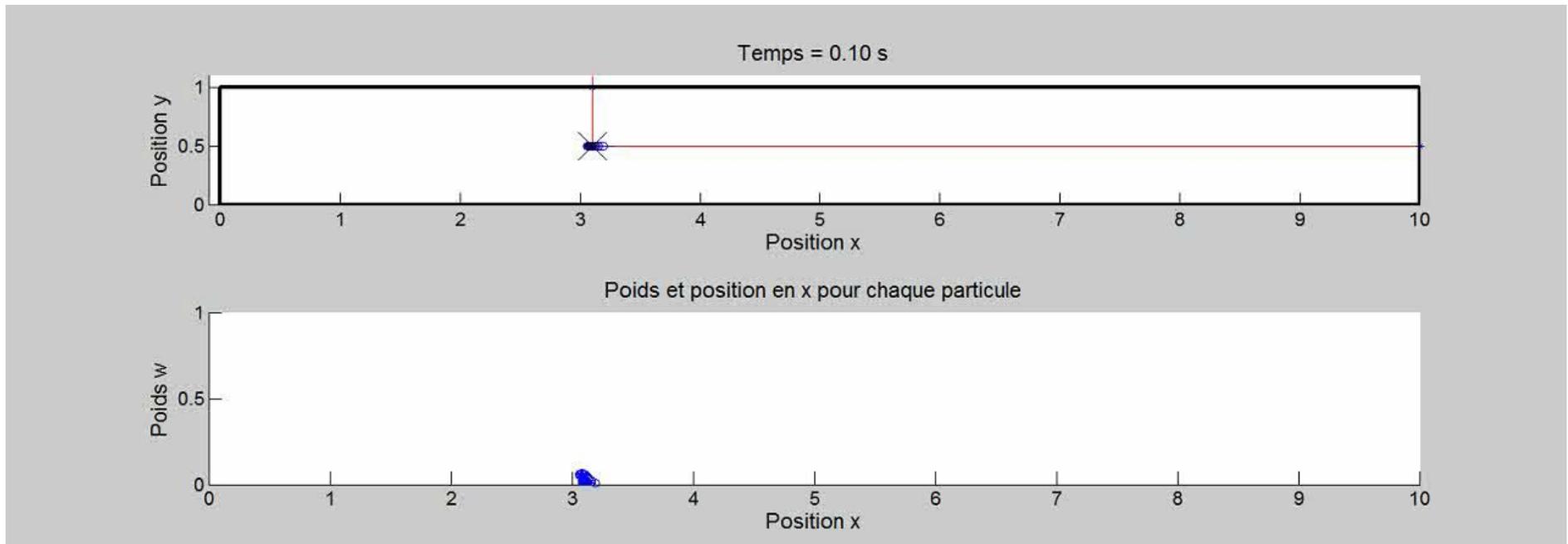


Appauvrissement avec le temps

- Avec les mises-à-jour, certaines particules finissent par avoir un poids $w_i \approx 0$
- Perte du pouvoir de représentation — *mesure laser*



Sample Importance Resampling : SIR

- Critère d'appauvrissement :

(N_{eff} = nombre effectif de particules)

$$N_{eff} = \frac{1}{\sum_{i=1}^c w_i^2}$$

$$w_1 = \frac{1}{4}, w_2 = \frac{1}{4}, w_3 = \frac{1}{4}, w_4 = \frac{1}{4} \rightarrow \frac{1}{\sum_1^4 \left(\frac{1}{4}\right)^2}$$

Sample Importance Resampling : SIR

- Critère d'appauvrissement : $N_{eff} = \frac{1}{\sum_{i=1}^c w_i^2}$
(N_{eff} = nombre effectif de particules)

$$w_1 = \frac{1}{4}, w_2 = \frac{1}{4}, w_3 = \frac{1}{4}, w_4 = \frac{1}{4} \Rightarrow \frac{1}{\sum_1^4 \left(\frac{1}{4}\right)^2} = \frac{1}{\sum_1^4 \frac{1}{16}} = \frac{1}{\frac{4}{16}} = \frac{16}{4} = 4$$

Sample Importance Resampling : SIR

- Critère d'appauvrissement : $N_{eff} = \frac{1}{\sum_{i=1}^c w_i^2}$
(N_{eff} = nombre effectif de particules)

$$w_1 = \frac{1}{4}, w_2 = \frac{1}{4}, w_3 = \frac{1}{4}, w_4 = \frac{1}{4} \Rightarrow \frac{1}{\sum_1^4 \left(\frac{1}{4}\right)^2} = \frac{1}{\sum_1^4 \frac{1}{16}} = \frac{1}{\frac{4}{16}} = \frac{16}{4} = 4$$

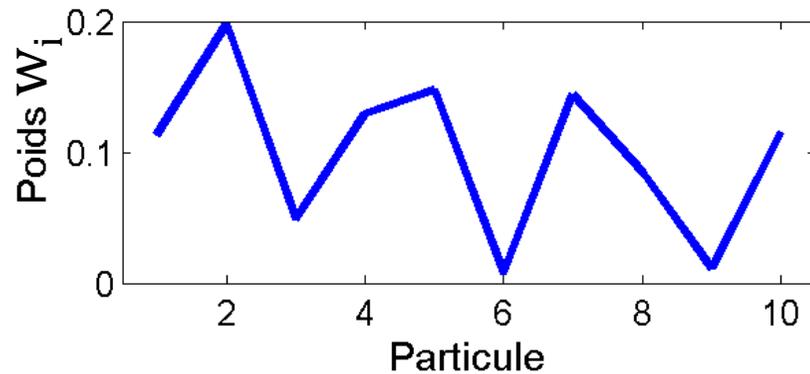
$$w_1 = 0.85, w_2 = 0.05, w_3 = 0, w_4 = 0.1 \Rightarrow N_{eff} = 1.3605$$

Sample Importance Resampling : SIR

- Si $N_{eff} < N_{seuil}$, il y a appauvrissement
 - peu de particules avec un poids w_i non-négligeable
- Normal, car à la longue l'état X_i diverge
- Doit donc rétablir une nouvelle population de particules
- Comment? En échantillonnant avec remplacement dans la population actuelle, avec probabilité modulée avec w_i

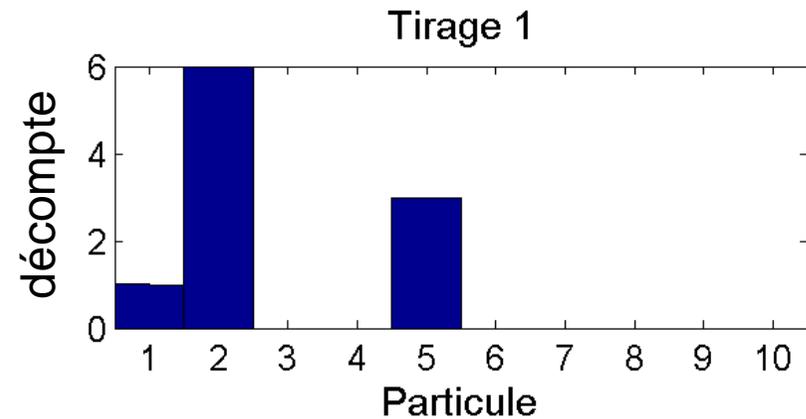
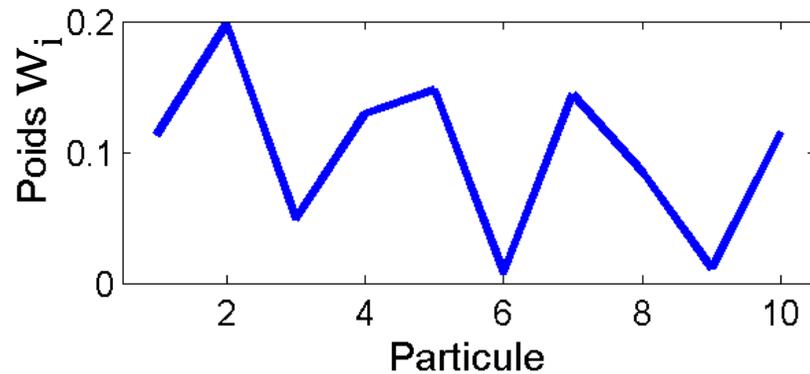
Sample Importance Resampling

- Exemple pour $C=10$ particules



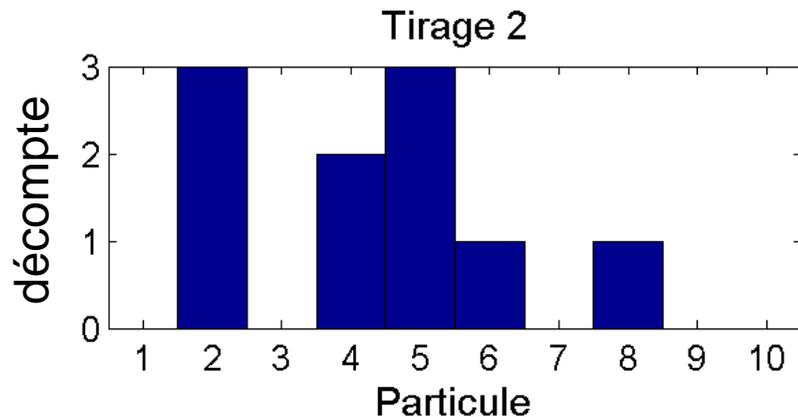
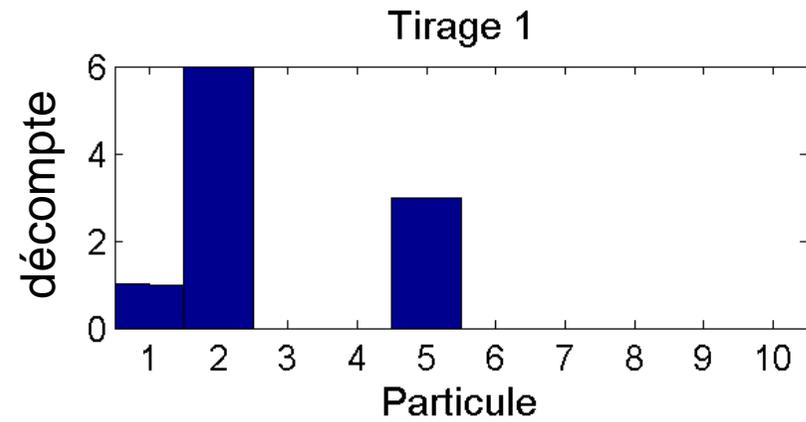
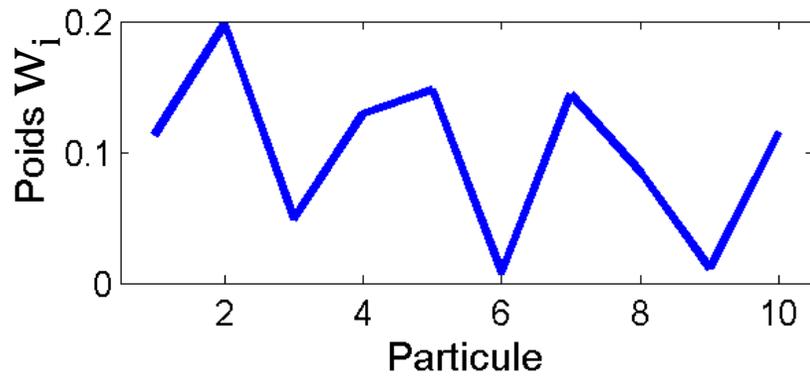
Sample Importance Resampling

- Exemple pour $C=10$ particules



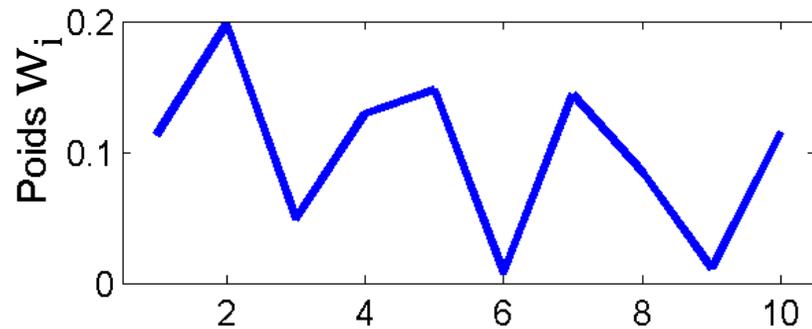
Sample Importance Resampling

- Exemple pour $C=10$ particules

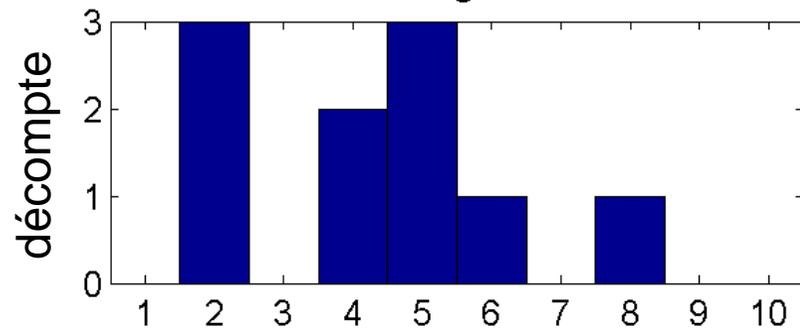


Sample Importance Resampling

- Exemple pour $C=10$ particules

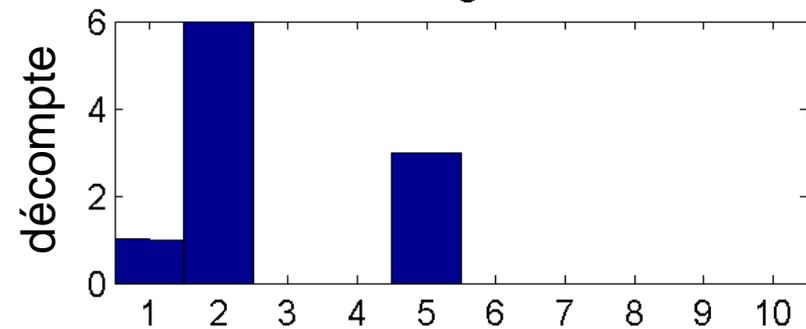


Particule
Tirage 2

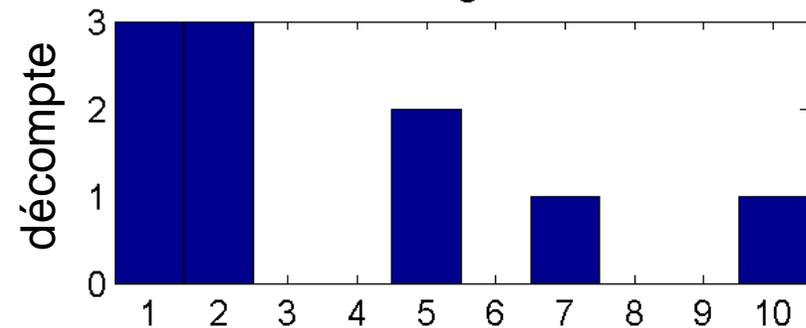


Particule

Tirage 1



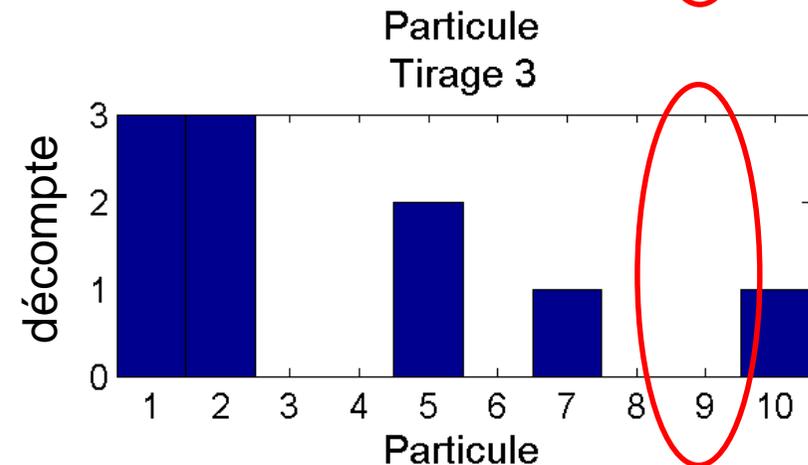
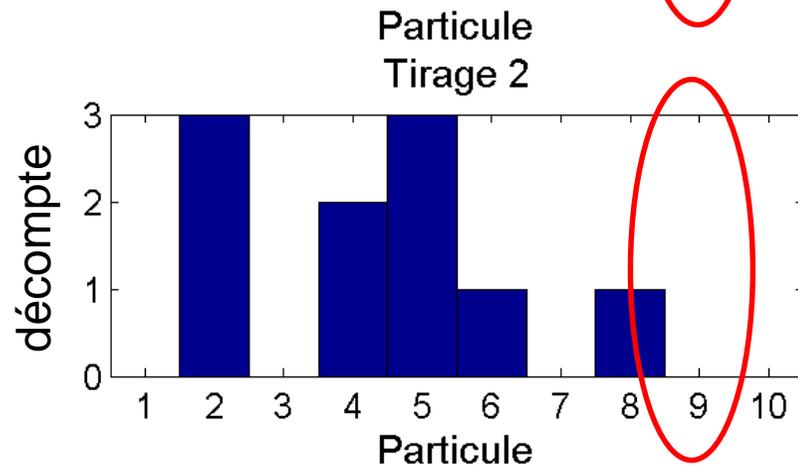
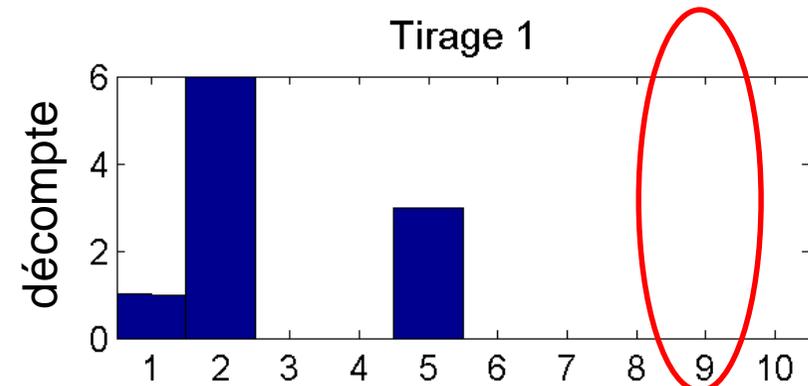
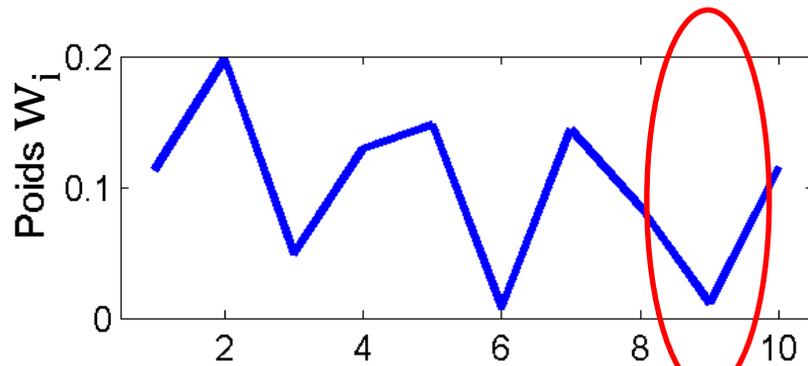
Particule
Tirage 3



Particule

Sample Importance Resampling

- Exemple pour $C=10$ particules



Particule 9 jamais pigée, car w_9 est très faible

Note : on ne fait qu'un seul tirage dans le filtre

Sample Importance Resampling : SIR

- Algorithme efficace de ré-échantillonnage :

Input: double $W[N]$

Require: $\sum_{i=1}^N W_i = 1$

$Q = \text{cumsum}(W)$; { calculate the running totals $Q_j = \sum_{l=0}^j W_l$ }

$t = \text{rand}(N+1)$; {t is an array of $N+1$ random numbers.}

$T = \text{sort}(t)$; {Sort them ($O(n \log n)$ time)}

$T(N+1) = 1$; $i=1$; $j=1$; {Arrays start at 1}

while ($i \leq N$) **do**

if $T[i] < Q[j]$ **then**

 Index[i]=j;

$i=i+1$;

else

$j=j+1$;

end if

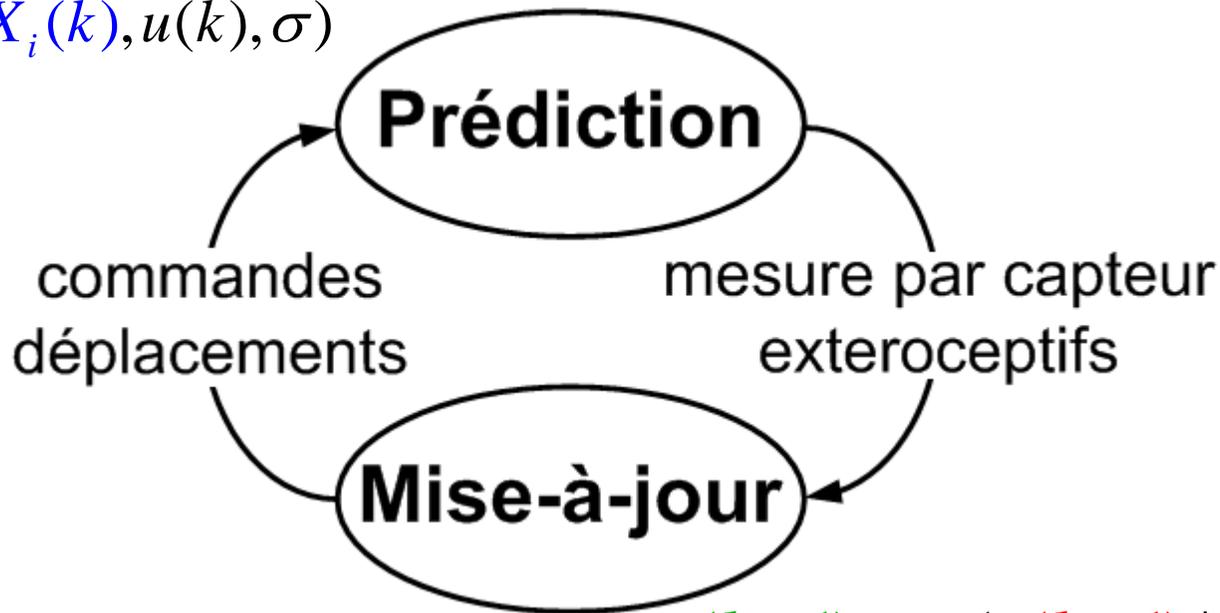
end while

Return(Index)

Prédiction + mise-à-jour

- Même mécanique de base que Kalman

$$X_i(k+1) = f_X(X_i(k), u(k), \sigma)$$



$$w_i(k+1) = p(z(k+1) | X_i(k+1))w_i(k)$$

Normaliser les w_i

Resampling, si $N_{eff} < N_{seuil}$

Filtre à particules

```
while (explore)
```

```
  for i=1:C
```

```
     $X_i(k+1) = f_X(X_i(k), u(k), \sigma)$       Prédiction
```

```
  end
```

```
   $z(k+1) = \text{mesure}()$  ;
```

```
  for i=1:C
```

```
     $w_i(k+1) = p(z(k+1) | X_i(k+1))w_i(k)$       Mise-à-jour
```

```
  end
```

```
  for i=1:C
```

```
     $w_i(k+1) = \frac{w_i(k+1)}{\sum_j \{w_j(k+1)\}}$       Normalisation
```

```
  end
```

```
  if ( $N_{eff} < N_{seuil}$ )
```

```
     $X_i(k+1) = \text{resample}(X_i(k+1), w_i(k+1))$  ;
```

```
     $w_i(k+1) = 1/C$ 
```

```
  end
```

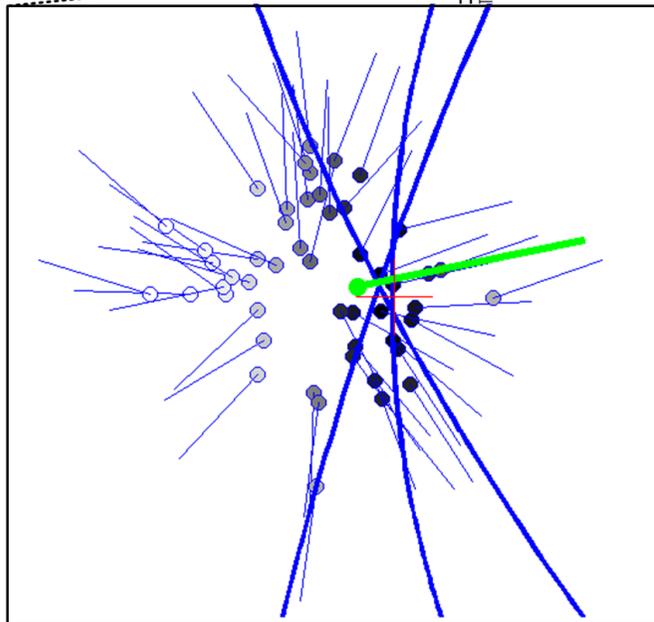
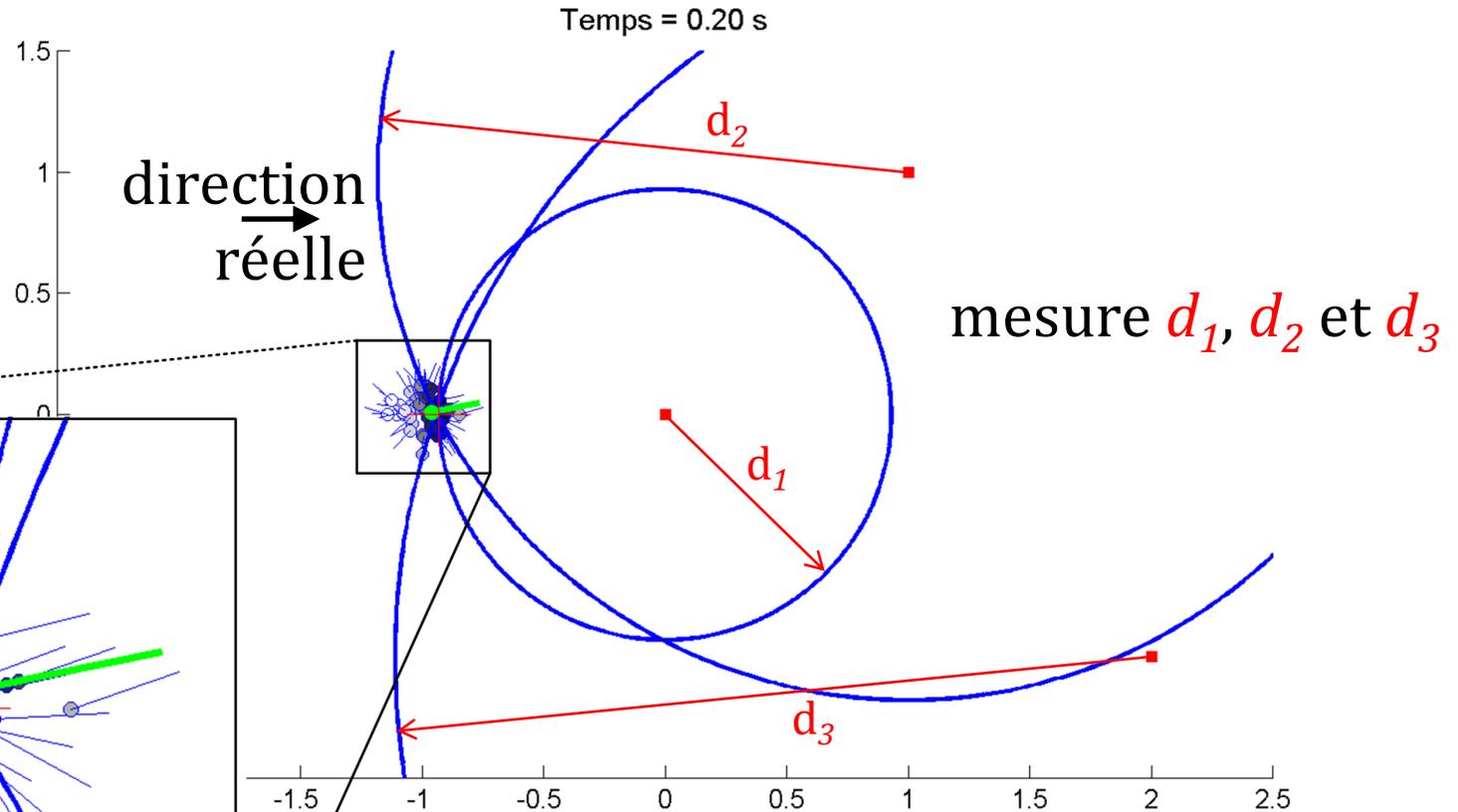
```
  k=k+1
```

$$N_{eff} = \frac{1}{\sum_{i=1}^N w_i^2} < N_{seuil}$$

Ré-échantillonnage

Exemple 1 : particules + trilatération

- Moyenne pondérée position + angle
- Repères
- Poids w élevé
- Poids w moyen
- Poids w faible

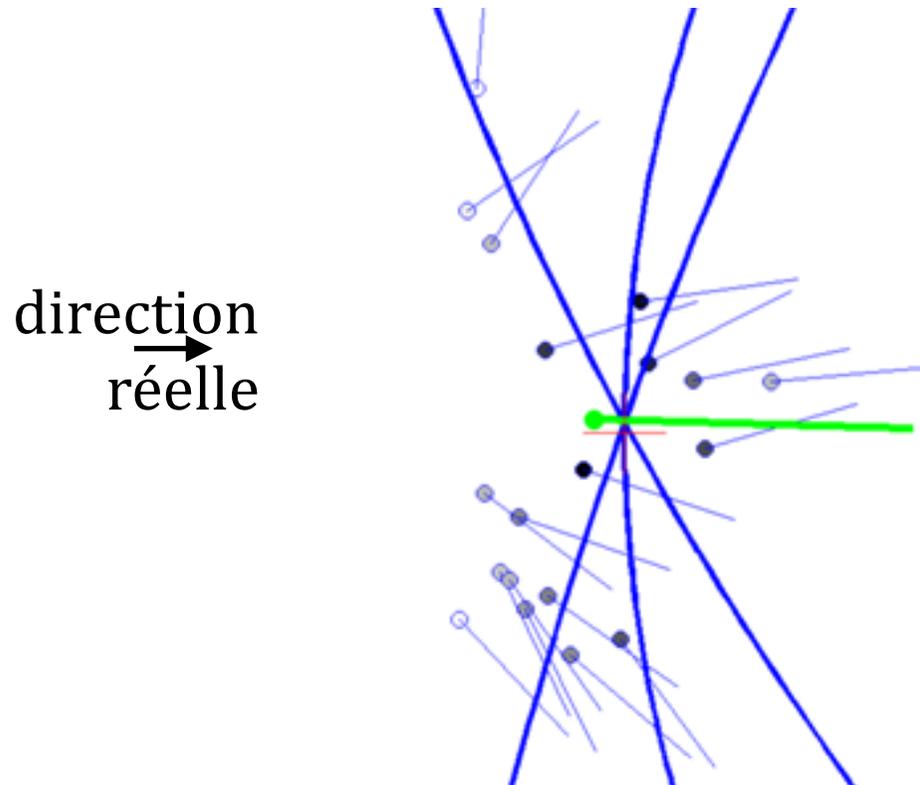


Après 1 déplacement

Initialisation de $C=50$ particules $X_i=[x \ y \ \theta]^T$, position initiale x, y connue, angles θ au hasard. Le système arrive à trouver un estimé de θ .

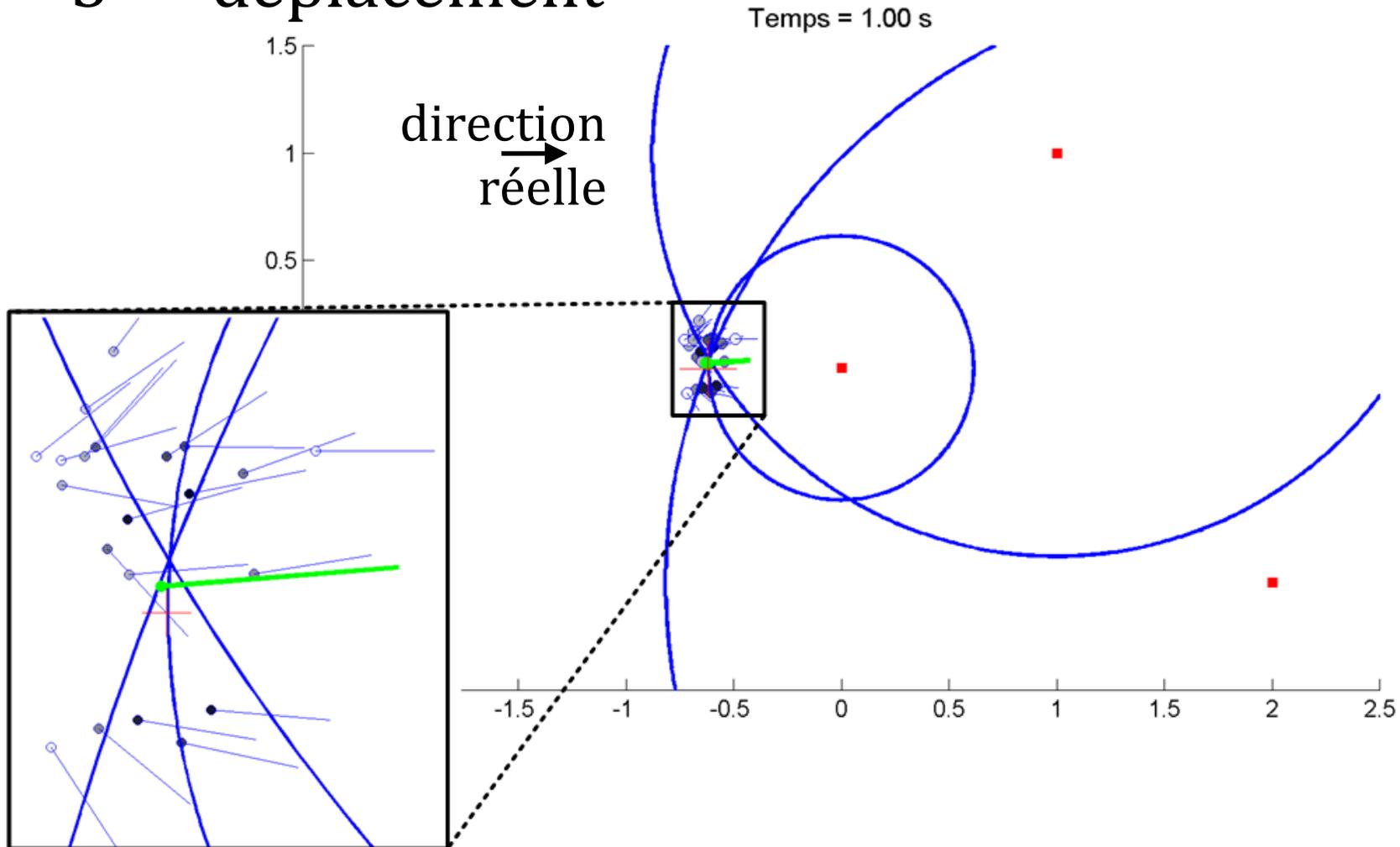
Exemple 1 : particules + trilatération

- Deuxième déplacement : presque toutes les particules avec le mauvais θ sont disparues



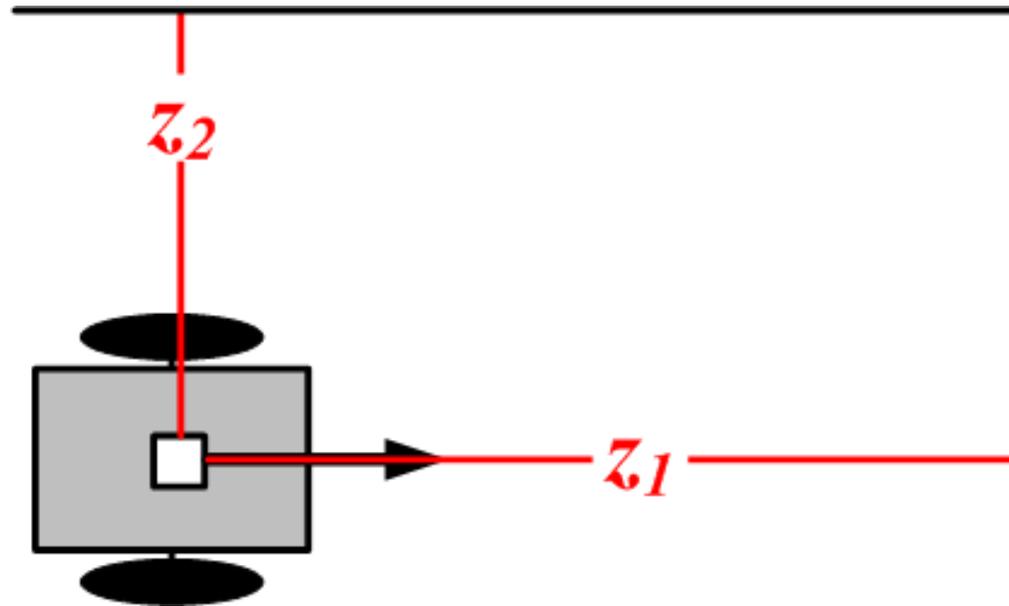
Exemple 1 : particules + trilatération

- 5^{ème} déplacement



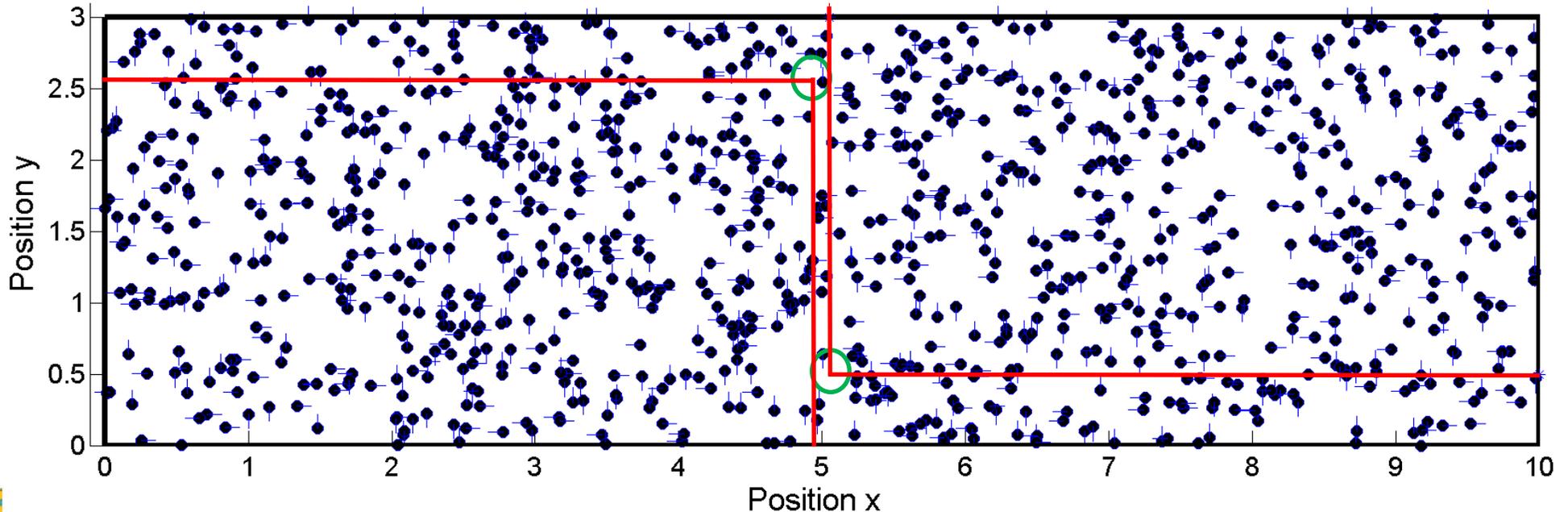
Exemple 2 : monde rectangle (similarité)

- Robot avec deux mesures laser : z_1 et z_2
- Pas de compas
- Carte connue



Exemple 2 : monde rectangle (similarité)

- Initialisé avec $C=400$ particules au hasard
 - position : uniformément (Problème de localisation globale)
 - angles : 0, 90, 180 ou 270 degrés
- Ligne rouge : mesures laser

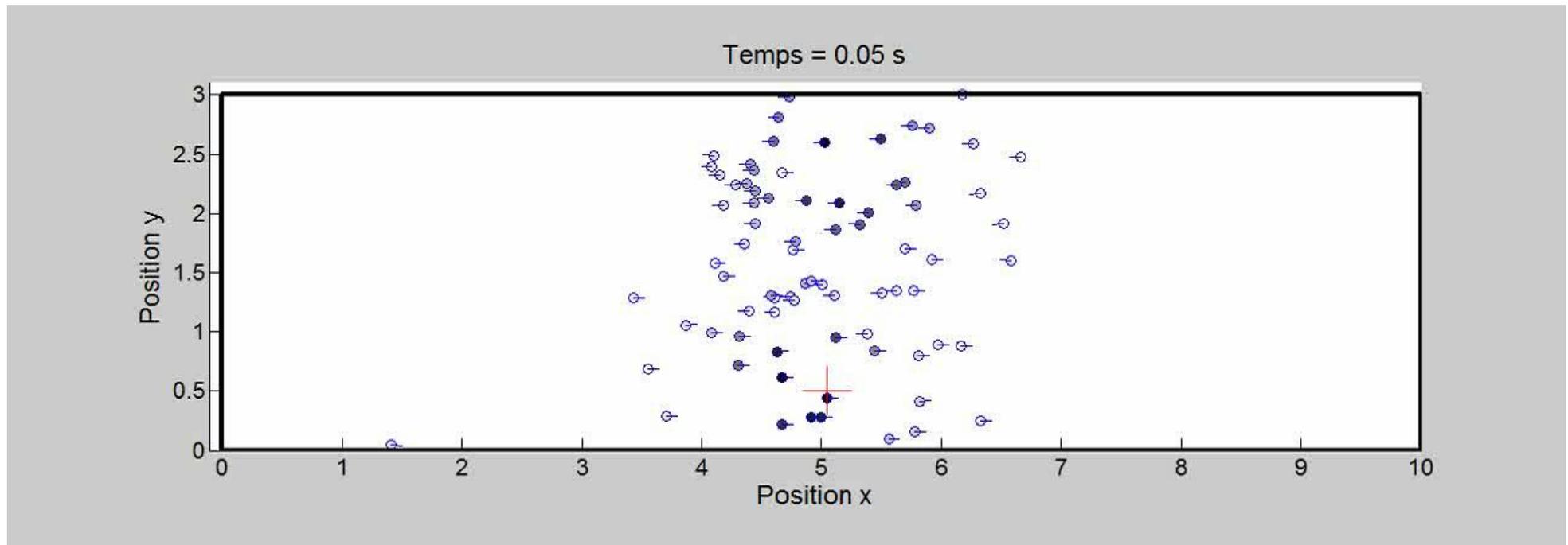


Exemple 2 : monde rectangle (similarité)

- Symétrie + absence de compas = ambiguïté

+ Position réelle du robot

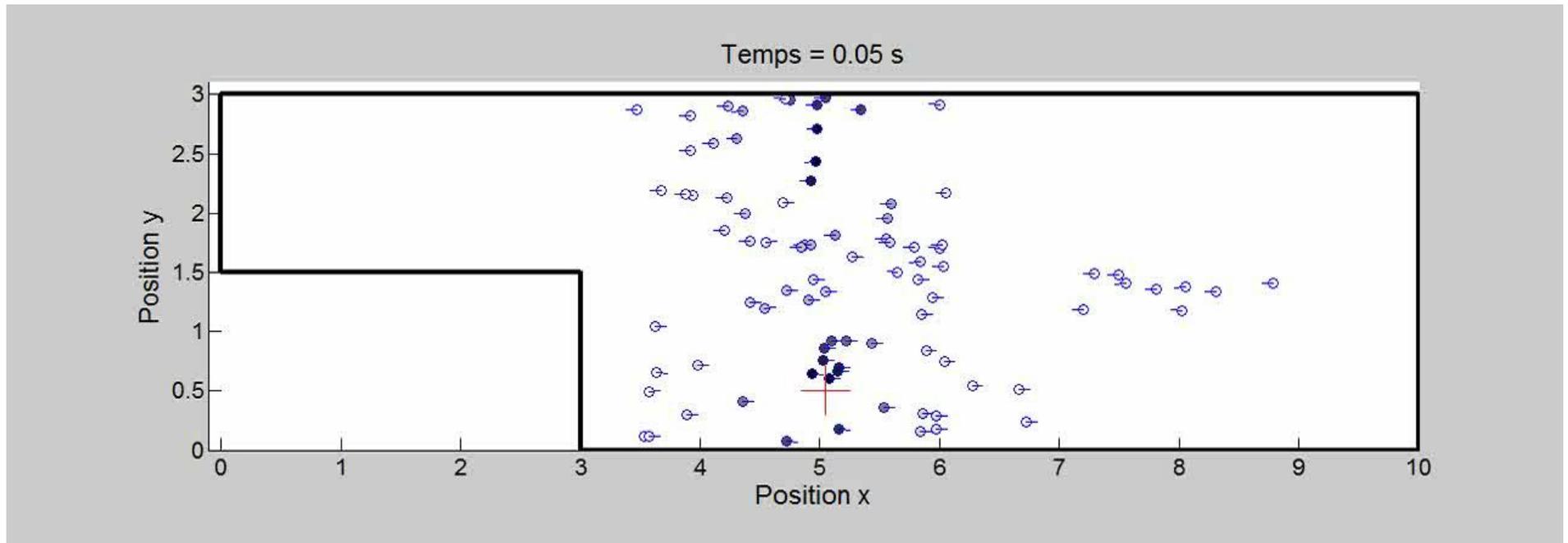
● Poids w élevé
● Poids w moyen
○ Poids w faible



Exemple 3 : monde asymétrique

- Asymétrie de la carte = levée de l'ambiguïté

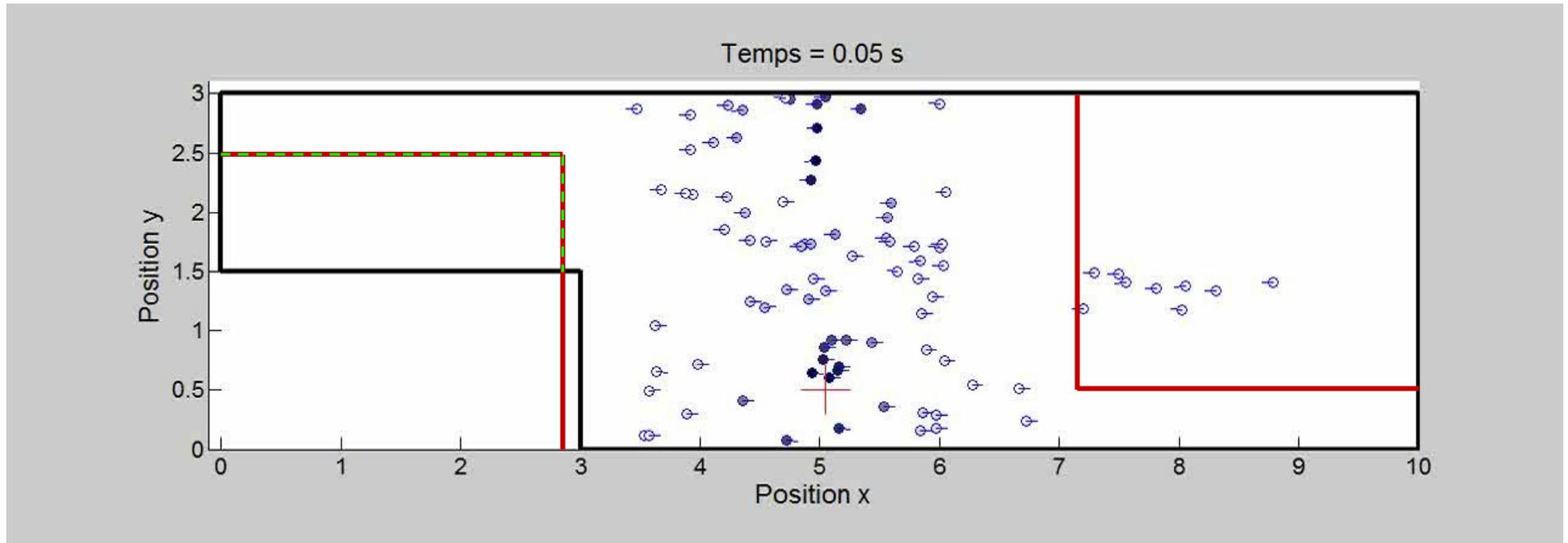
● Poids w élevé
● Poids w moyen
○ Poids w faible



Exemple 3 : monde asymétrique

- Asymétrie de la carte = levée de l'ambiguïté

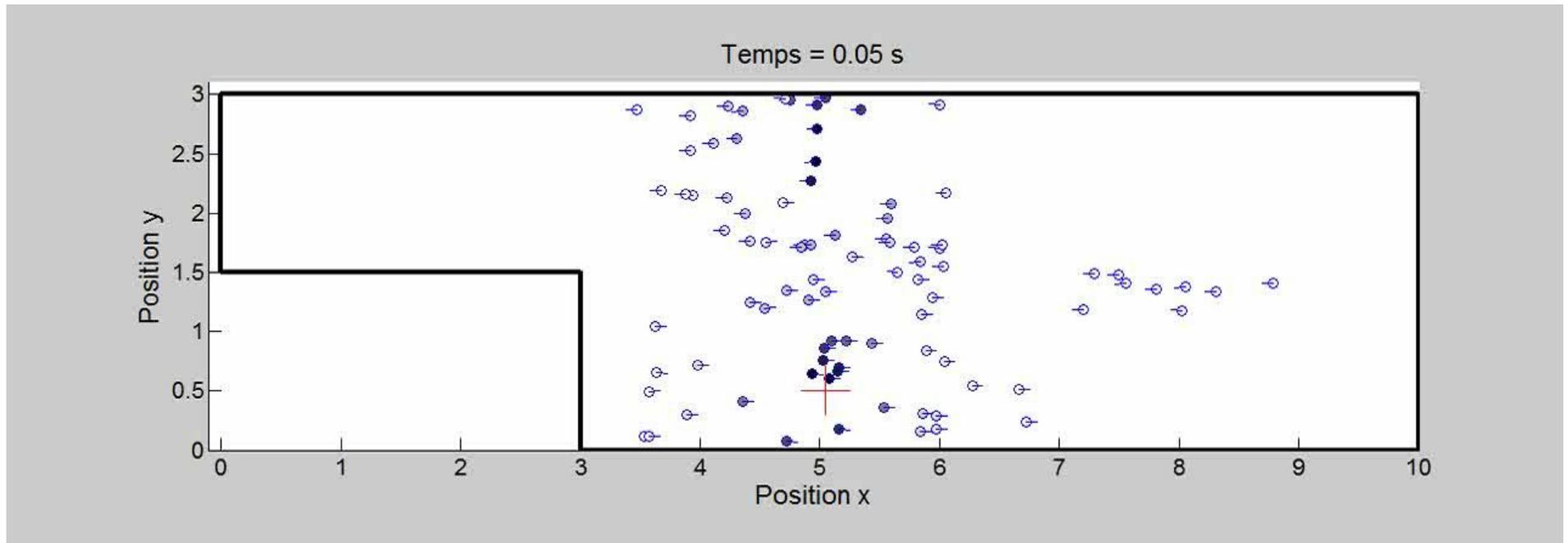
● Poids w élevé
● Poids w moyen
○ Poids w faible



Exemple 3 : monde asymétrique

- Asymétrie de la carte = levée de l'ambiguïté

● Poids w élevé
● Poids w moyen
○ Poids w faible

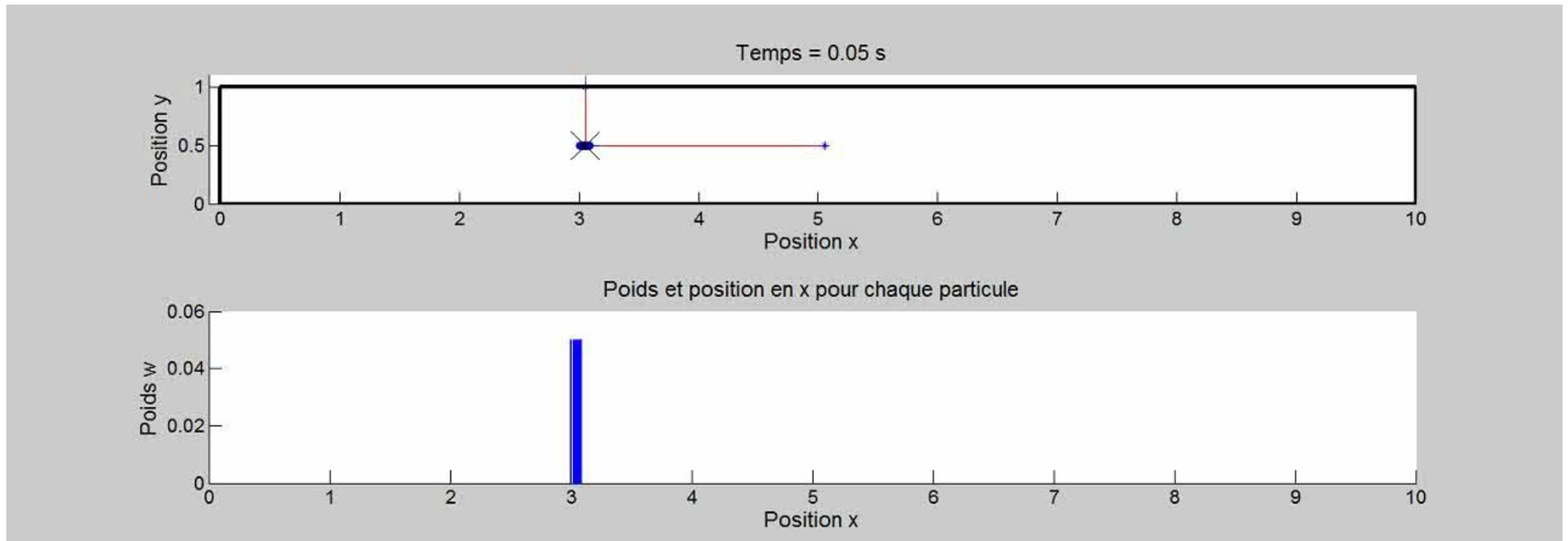


Corridor + laser courte portée

- Ici, le laser a une courte portée : 2 m.
- Perte de précision en x, loin du fond du couloir.

✗ Position réelle du robot

- Poids w élevé
- Poids w moyen
- Poids w faible



Bien adapté à la localisation globale

Quand on ne sait pas du tout où est la position de départ...



Trajectoire parcourue 

tiré de Probabilistic Robotics, par Thrun, Burgard et Fox

Bien adapté à la localisation globale



Trajectoire parcourue 

tiré de Probabilistic Robotics, par Thrun, Burgard et Fox

Bien adapté à la localisation globale



Trajectoire parcourue —

tiré de Probabilistic Robotics, par Thrun, Burgard et Fox

Fusion odom + gyro + GPS + carte!

- Bruit sur GPS : $\sigma = 50 \text{ m}$
- Bruit sur déplacement $\sigma = 2 \text{ m}$
- Odométrie + rotations du véhicule
- Heuristique : si la position n'est pas sur une route, réduire le poids de la particule de 90% :

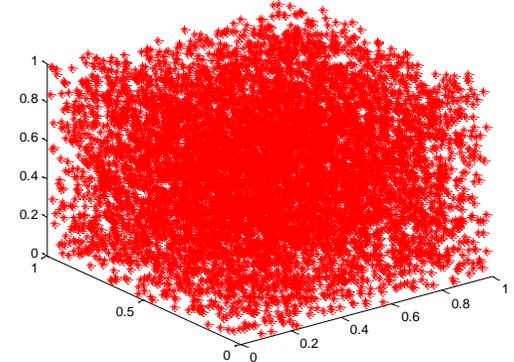
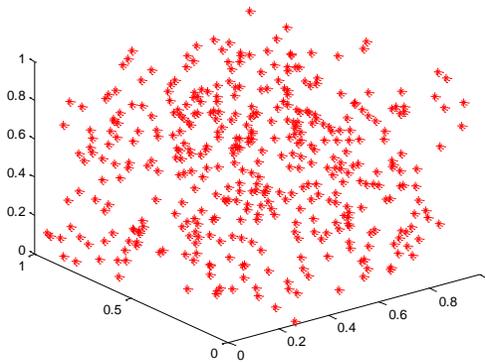
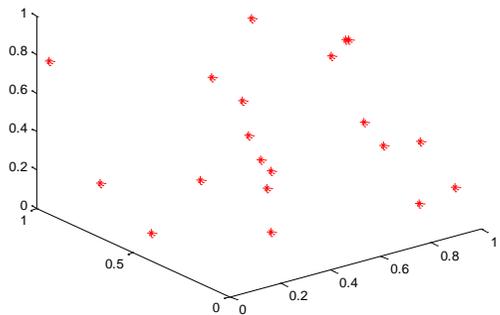
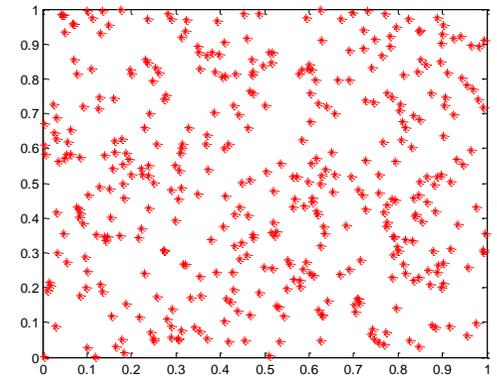
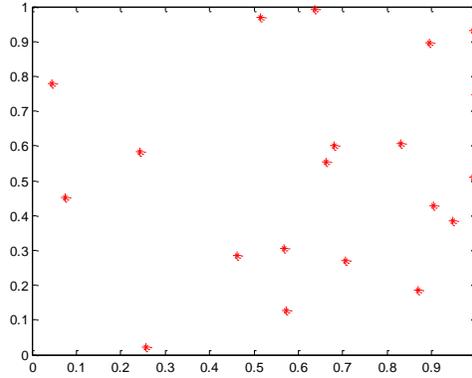
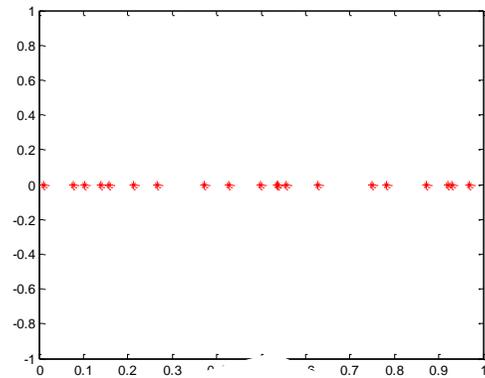
if (PasSurRoute($x(i)$))

$$w(i) = 0.1w(i)$$

end

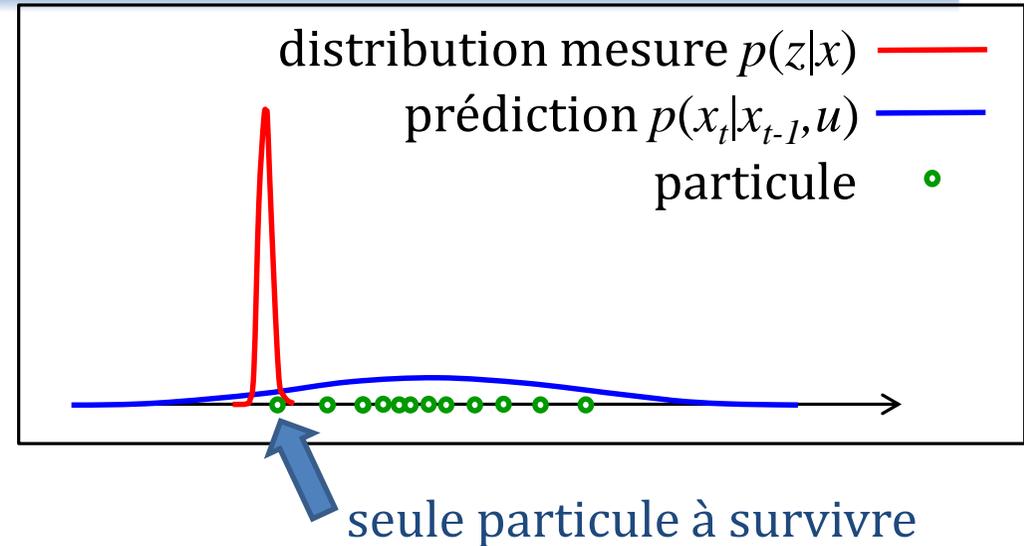
Gardez à l'esprit...

- Le nombre de particule C influence la précision
- Plus la dimensionnalité de l'état est grande, plus C doit être grand.



Problème si capteur trop précis...

- Peu de particules se trouvent dans la région couverte par la mesure
- Perte de variété (*deprivation*)
- Solutions:



- Placer des particules à l'endroit de la mesure

Lenser, Veloso. *Sensor resetting localization for poorly modelled mobile robots*. ICRA, 2000.

- Piger des particules dans une meilleure distribution combinant l'estimé de position et la mesure

- gmapping, FASTSLAM 2.0.

- Utiliser un bruit de capteur plus grand que la réalité

Variations pour filtres à particules

- Ajouter des particules uniformément après la mise-à-jour
 - *Augmente robustesse, survit au kidnapping*
- Ajouter des particules près de l'endroit indiqué par les capteurs
 - *prévient le problème de capteurs trop précis*
 - *difficile si l'inversion de la fonction de capteur ne nous donne pas une position*
- Augmenter ou diminuer le nombre C de particules en fonction du temps
 - *Adapting the sample size in particle filters through KLD-Sampling, D. Fox, IJRR, 2003.*

Filtres : mesures à différents rythmes?

- Commandes aux 100 ms
- Capteur 1 aux 300 ms
- Capteur 2 aux 500 ms

Autant pour les filtres à particules
que les filtres de Kalman

Propage commande
Mise-à-jour mesure z1
Propage commande
Propage commande
Propage commande
Mise-à-jour mesure z1
Propage commande
Propage commande
Mise-à-jour mesure z2
Propage commande
Mise-à-jour mesure z1
Propage commande
Propage commande
Propage commande
Mise-à-jour mesure z1
Propage commande
Mise-à-jour mesure z2
Propage commande
Propage commande
Mise-à-jour mesure z1

Types de problème de localisation

Problèmes de localisation

- **Localisation faible**

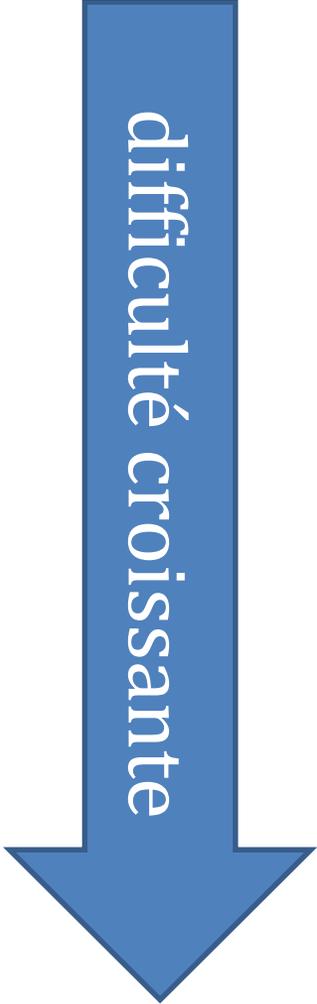
- « Ai-je déjà été ici ? »
- marqueur
- algorithmes de planification $bug0$, $bug1$, $bug2$ (qu'on ne voit plus)
- reconnaissance de lieu

- **Localisation forte**

- « Où suis-je par rapport à un repère global ? »
- $p(x_t / z_{1:t} u_{1:t})$
 - x_t : pose actuelle du robot
 - $z_{1:t}$: mesures des capteurs et $u_{1:t}$: commandes

Localisation forte

- **Tracking (unimodale)**
 - je connais ma position initiale x_0
 - implémenté par Kalman, ~~meanshift tracker~~
- **Globale (multimodale)**
 - je ne connais pas ma position initiale x_0
 - implémenté par filtre à particule, etc.
- **Kidnapping**
 - on téléporte le robot sans l'avertir
 - algorithme qui diverge/échoue
 - (il faut détecter ces échecs)



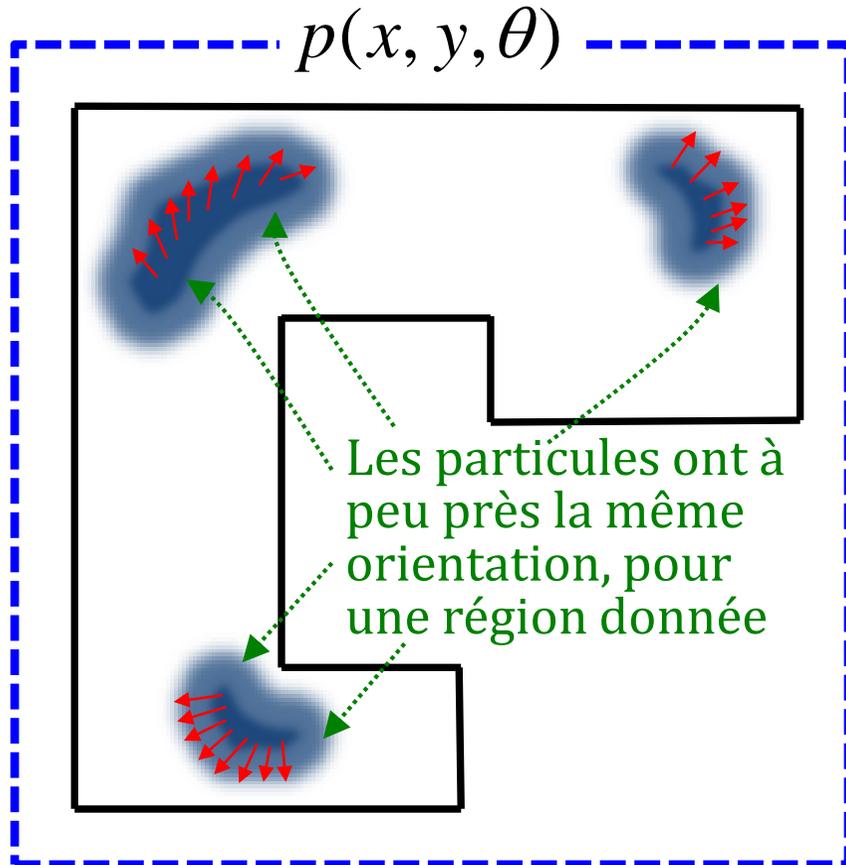
difficulté croissante

Rao Blackwellized Particle Filter (RBPF)

Filtre à particules Rao-Blackwellized

- FP : Si le nombre de variables d'état est grand, le nombre de particules nécessaires devient grand
 - Fléau de la dimension
- Kalman : la distribution à représenter doit être unimodale, idéalement proche normale
- Filtre particule *Rao-Blackwellized* :
 - On tente de combiner les deux...
 - Va exploiter la structure présente dans le problème

Exemple



$$\begin{aligned} p(x, y, \theta) &= p(x | y, \theta) p(y, \theta) \\ &= p(y | x, \theta) p(x, \theta) \\ \longrightarrow &= p(\theta | x, y) p(x, y) \end{aligned}$$

Y a-t-il une factorisation
« meilleure » que les
autres, en fonction de
notre problème?

Si on fait l'hypothèse que
sachant x, y , l'orientation
 θ est dist. unimodale, oui!

Factoriser en deux problèmes

Distribution
unimodale : Kalman

$$p(x, y, \theta) = p(\theta | x, y) p(x, y)$$

Distribution
multimodale : FP

Chaque particule sera donc un
filtre de Kalman!

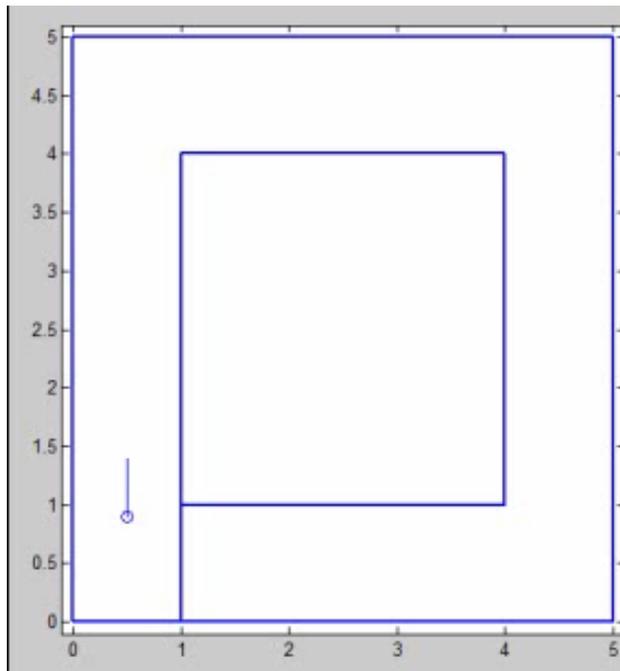
(On y reviendra +tard pour FASTSLAM)

Iterative Closest Point : **ICP**

Positionnement relatif par scan 2D

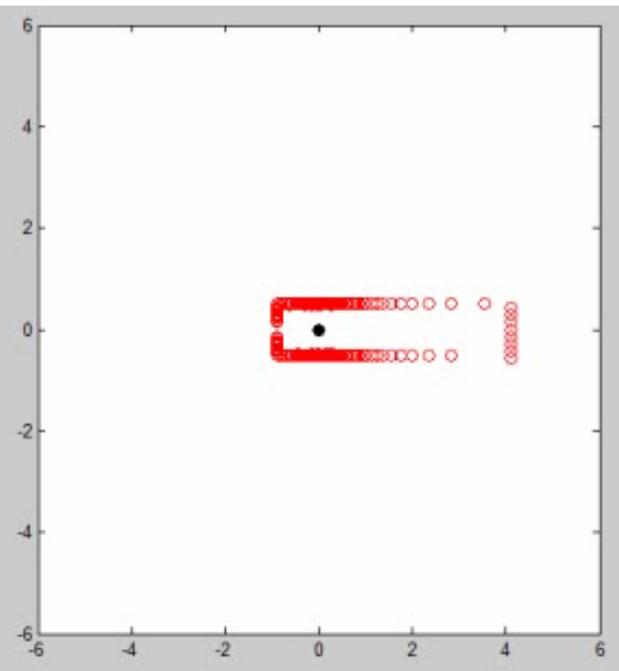
- Utiliser directement les données Z d'un capteur de distance pour estimer la position relative entre deux scans.
- Scénario :

**Monde m +
position du robot**



coordonnées globales

Scan laser

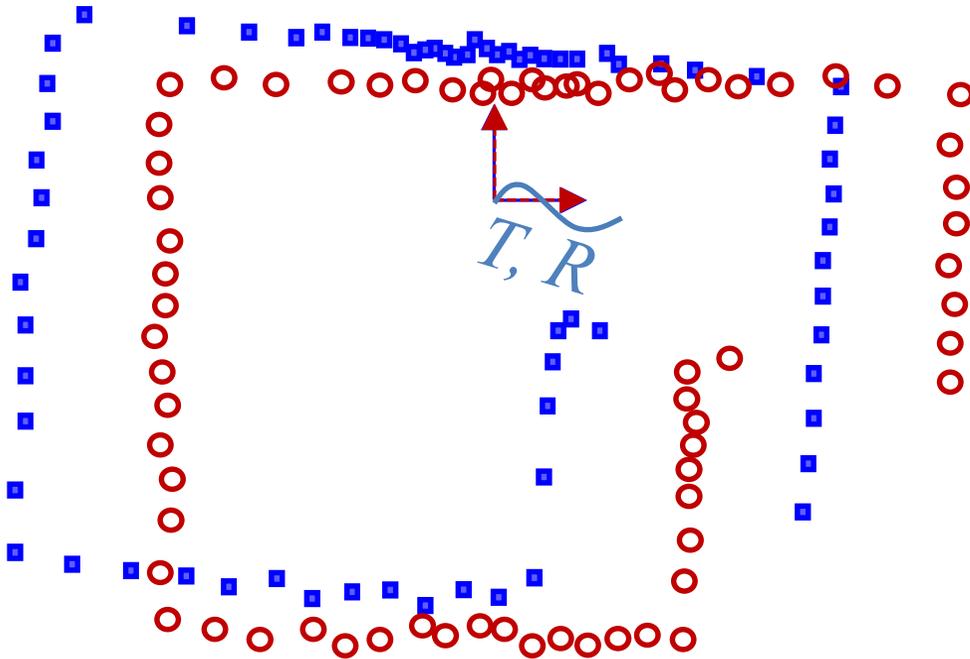


coordonnées locales (robot)



Opération de calage de points

- Deux scans 2D/3D, A et B , pris à des endroits différents mais proches
- Problème : trouver R et T entre A et B

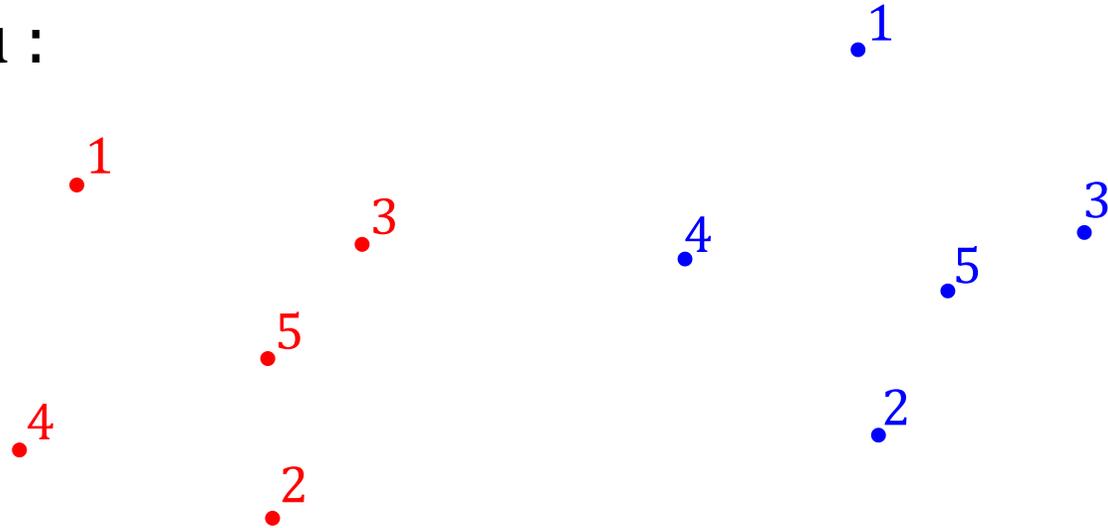


- Similaire à odométrie visuelle
- **Sauf que les points sont anonymes ici**
- **et les mesures peuvent ne pas être pour une même partie de surface**



Problème de calage de point (simple)

- Ici on assume que l'appariement entre points **A** et **B** est exact et connu :

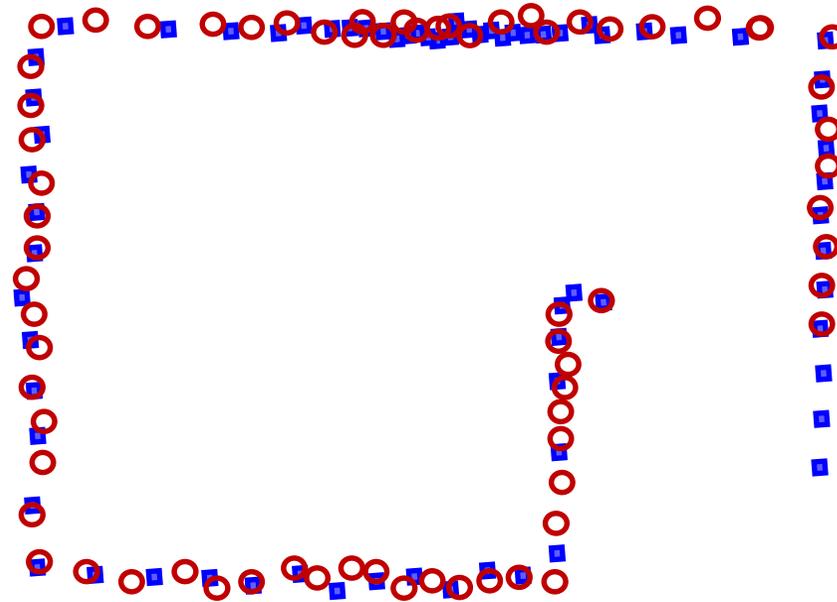


- Cherche : $\operatorname{argmin}_{R,T} \sum_i \| A_i - (T + RB_i) \|^2$

- Se résout en 10 lignes matlab, sans itération (SVD)

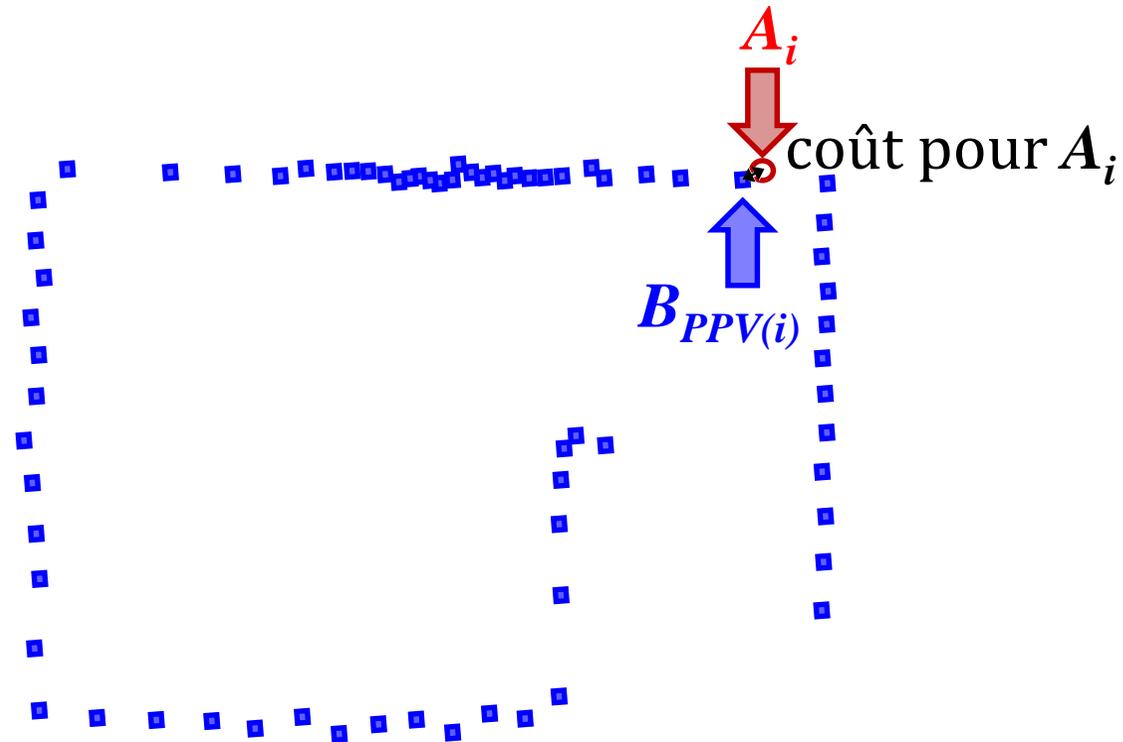
Plus proche voisin pour calcul coût

- Les points n'ont pas de signatures
- Apparier points par critère du plus proche voisin $PPV(i)$.



$$\operatorname{argmin}_{R,T} \sum_{i \in A} \| A_i - TRB_{PPV(i)} \|^2$$

Plus proche voisin pour calcul coût



$$\operatorname{argmin}_{R,T} \sum_{i \in A} \| A_i - TRB_{PPV(i)} \|^2$$

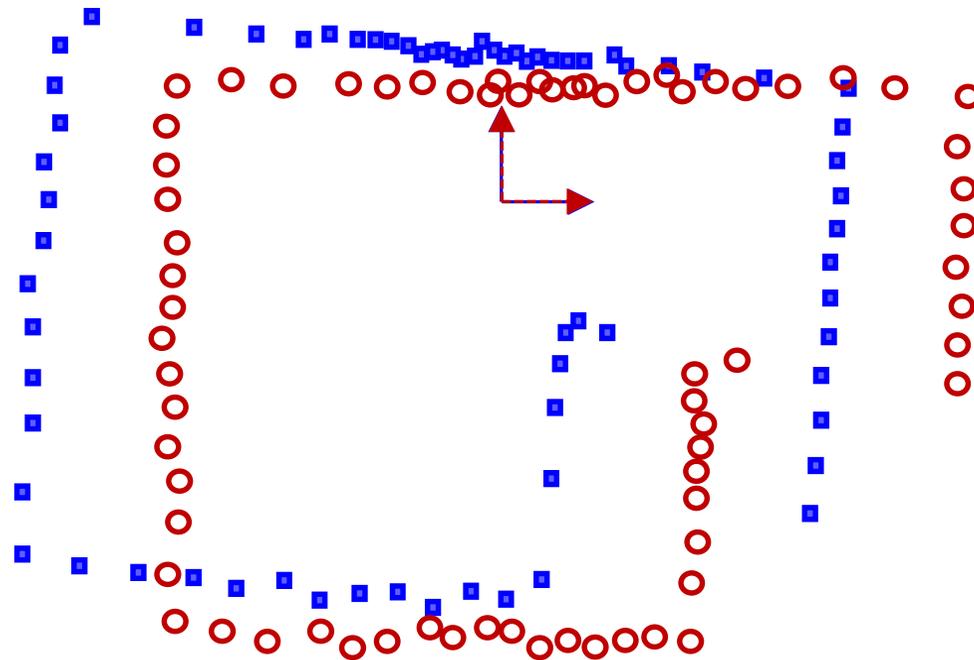
ICP : Iterative Closest Point

- Fonctionne par ajustement itératif avec T et R
- En gros, boucler jusqu'au critère de convergence :
 - Faire l'appariement $PPV(i)$
 - Estimer T et R qui minimisent* l'erreur :
$$A_i \quad B_{PPV(i)}$$
 - Applique la transformation : $B \leftarrow T + RB$
- **Attention!** ne fonctionne que pour des petites transformations T et R .
 - sorte de descente de gradient
 - sujet aux minimums locaux, si grands déplacements T et R

**solution approximative car dépend appariement, d'où le besoin de faire de multiples itérations*

Position de départ de recherche ICP

Utiliser l'odométrie + gyroscope pour trouver une position approximative

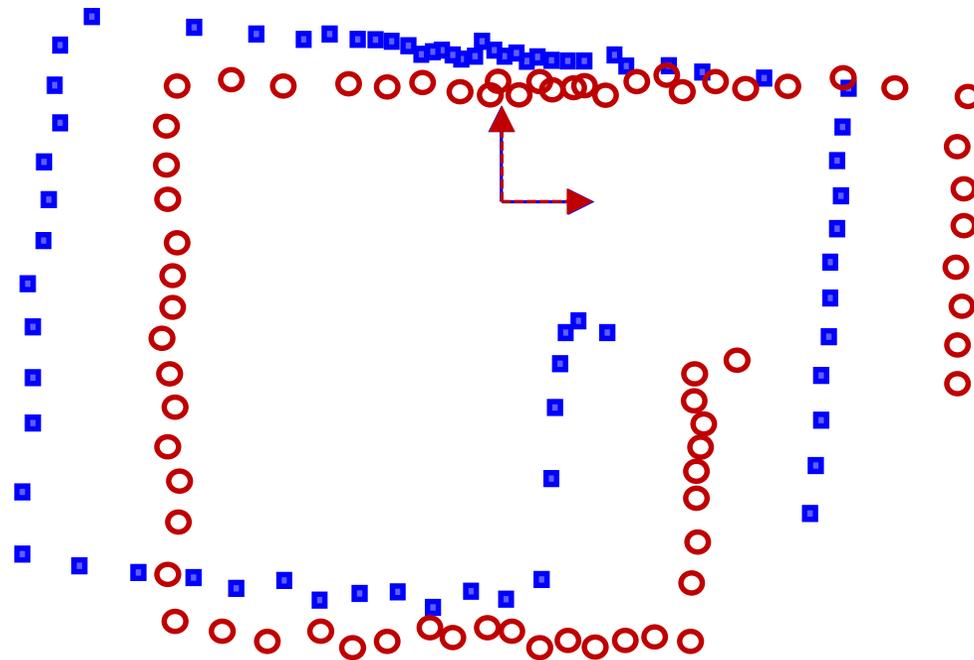


Proprioceptif

Avant ajustement odométrie

Position de départ de recherche ICP

Utiliser l'odométrie + gyroscope pour trouver une position approximative

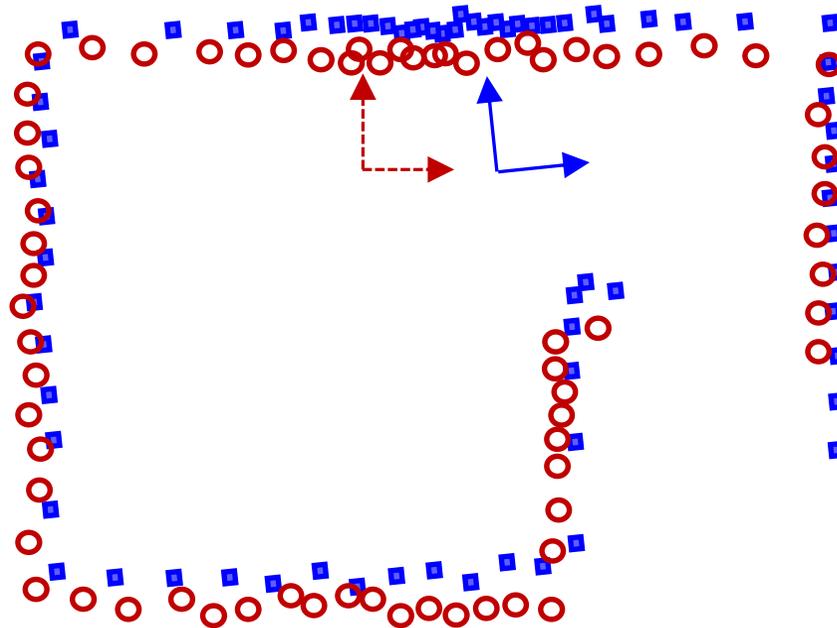


Proprioceptif

Avec ajustement odométrie

Position de départ de recherche ICP

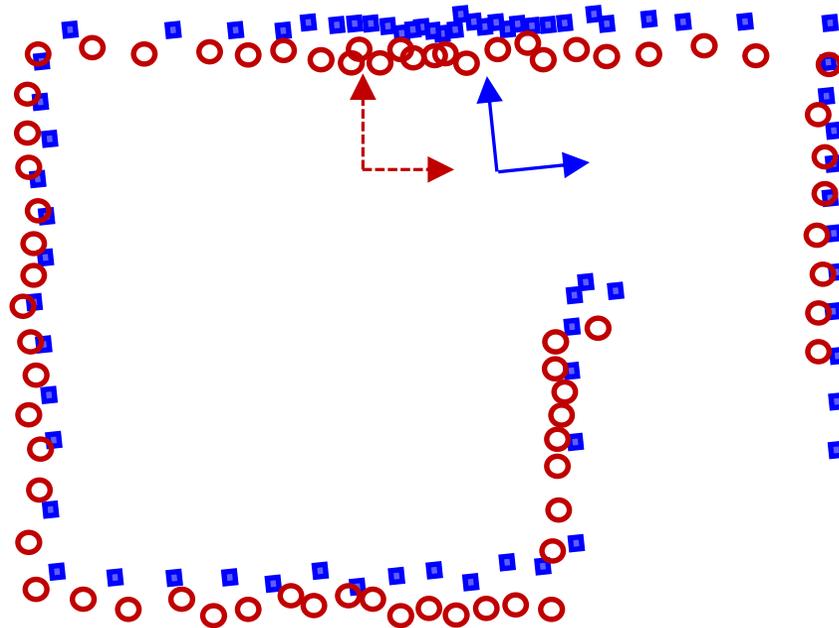
- Applique ensuite ICP, car on est proche du minimum local



Extéroceptif

Position de départ de recherche ICP

- Applique ensuite ICP, car on est proche du minimum global : raffine la position



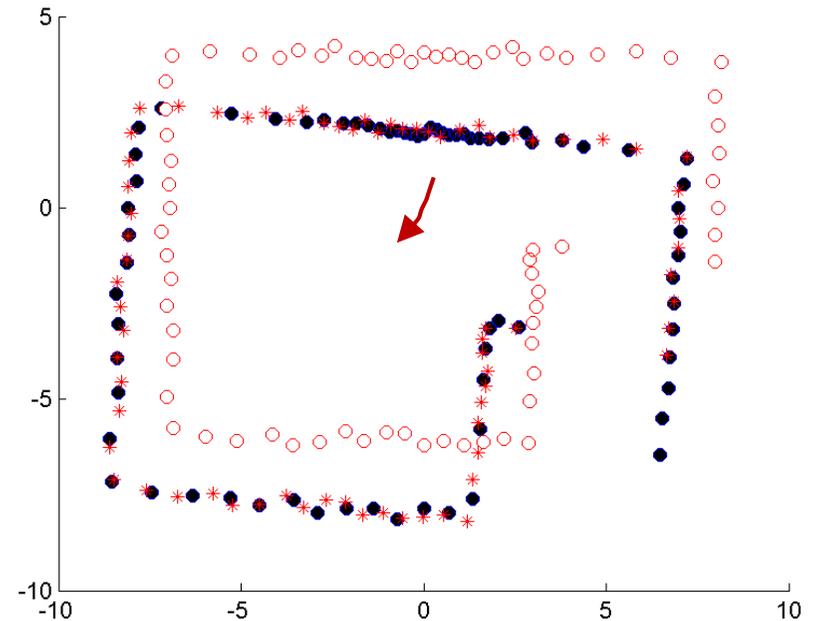
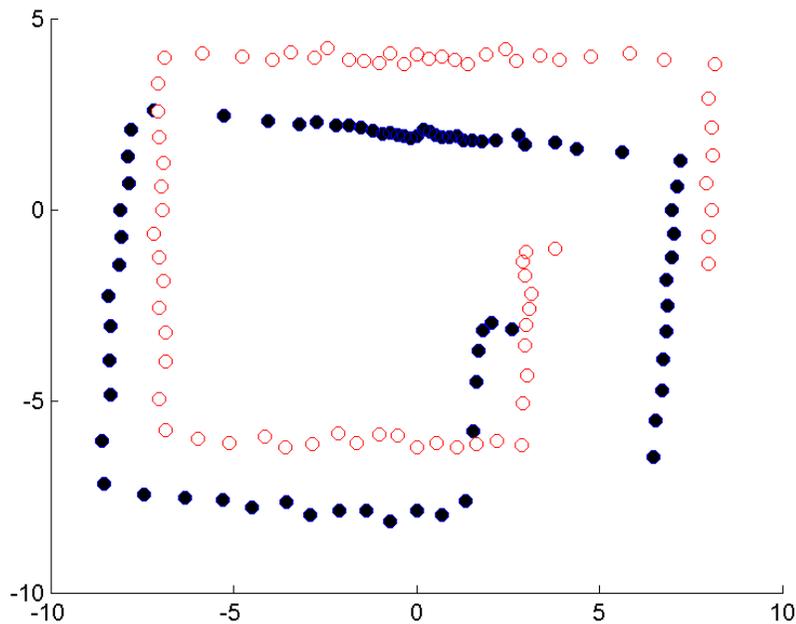
Extéroceptif

Exemple ICP

- Fonction `icp.m` de Per Bergström

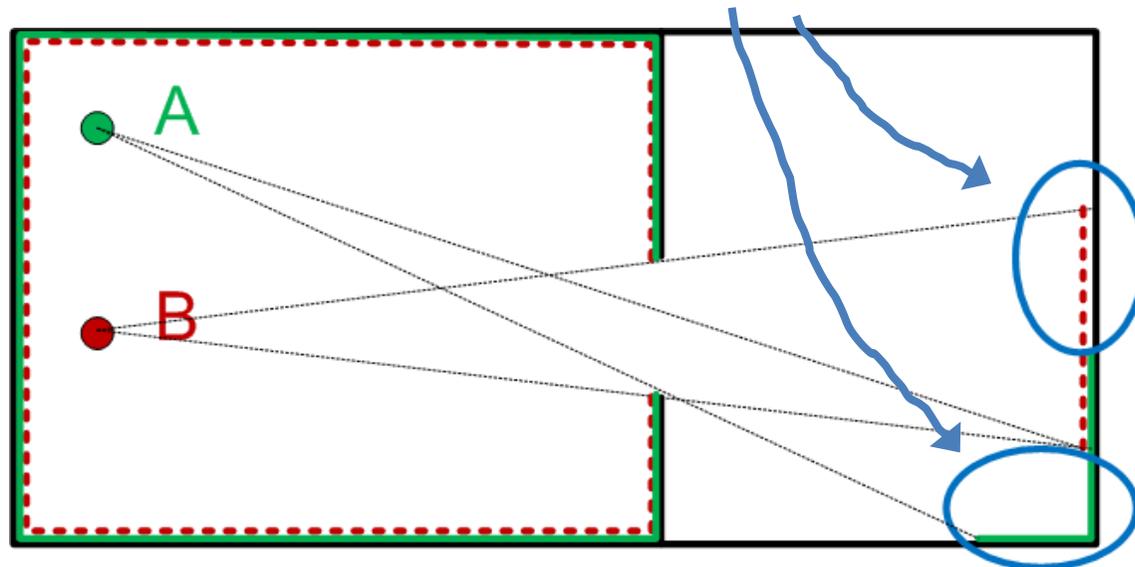
(<http://www.mathworks.com/matlabcentral/fileexchange/12627>)

Laboratoire de ce vendredi



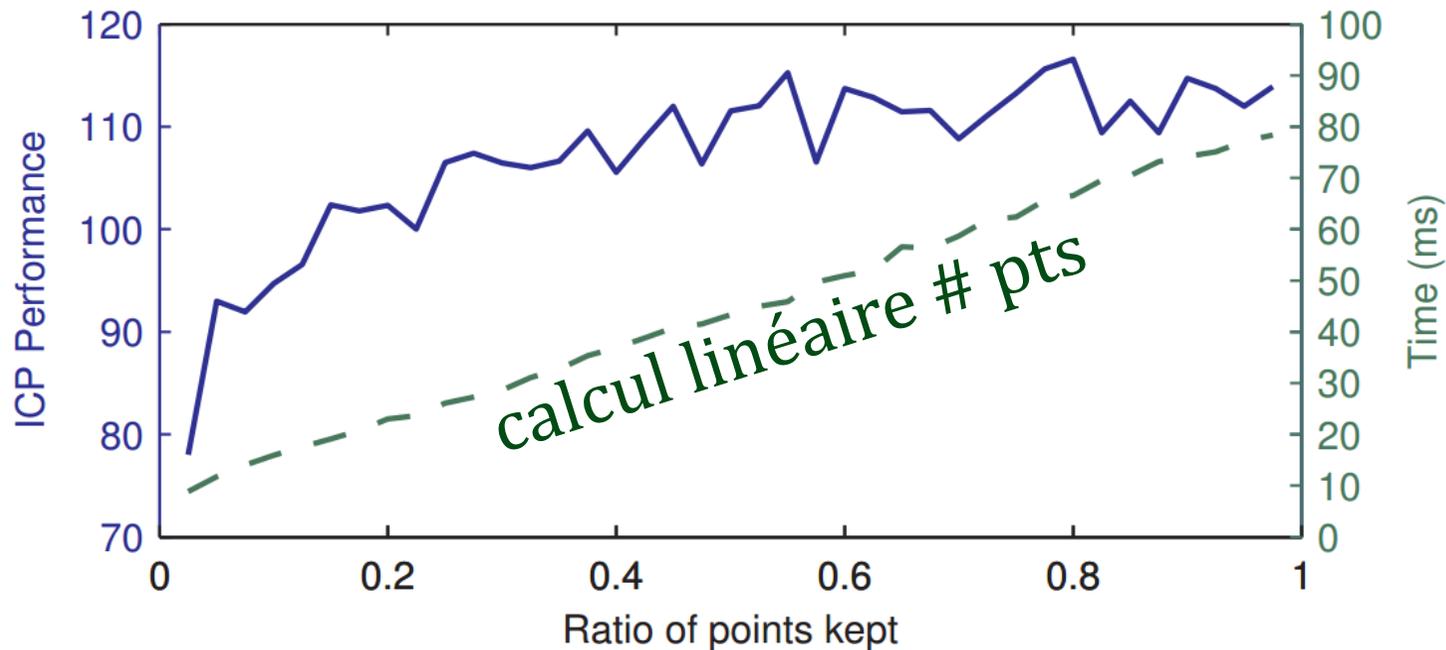
ICP : Problème des données aberrantes

- Quant il y a des occlusions dans la scène
 - pièce n'est pas convexe
- Certains points ne peuvent pas être apparié
- Présence de données aberrantes (*outliers*).



Considération : # points

- Possible d'accélérer les performances en décimant le nombre de points



Tracking a Depth Camera: Parameter Exploration for Fast ICP,
François Pomerleau et al., *IROS* 2011.

Librarie `libpointmatcher` F. Pomerleau

- LiDAR 2D+unité pan-tilt, centrale inertielle et odométrie



Olivier Falardeau, stagiaire d'été 2014.