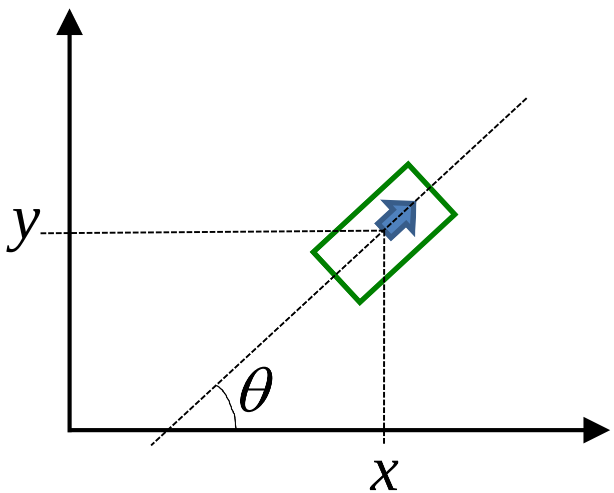
Modélisation des déplacements

Pose du robot

- Travailler dans un environnement plan



$$\text{pose}: x_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

(aucun lien entre x_t et x)

Modèle de déplacement par vitesses

- On peut généralement modéliser le déplacement d'un robot par une combinaison de vitesse
  - de translation V_t (unités de m/s)
  - angulaire ω_t (unités de rad/s, sens antihoraire)

$$\boxed{u_t = \begin{bmatrix} V_t \\ \omega_t \end{bmatrix}}$$

Commande u_t avec cond. différentielle

- Deux roues motorisées indépendantes.
- Roues avances avec vitesses V_g, V_d.

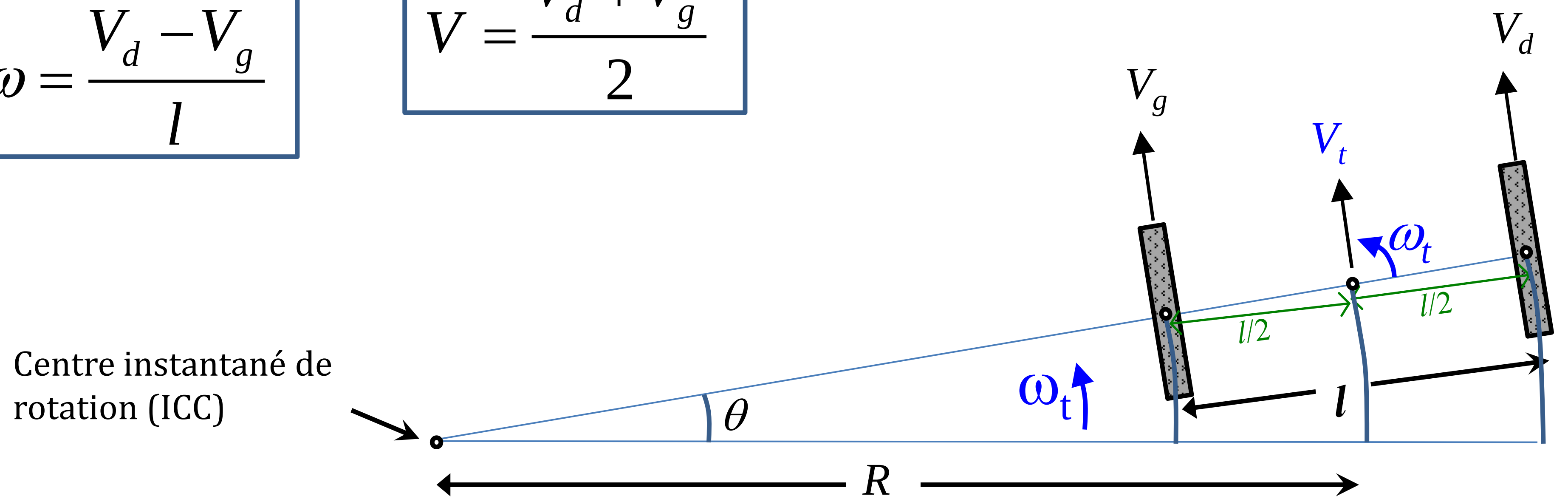
vitesse angulaire

$$\omega = \frac{V_d - V_g}{l}$$

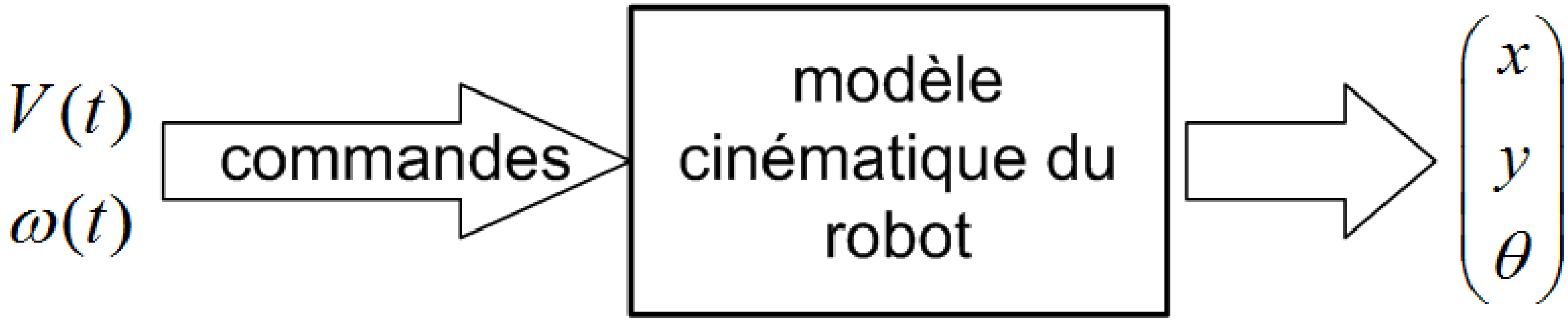
vitesse avant

$$V = \frac{V_d + V_g}{2}$$

$$u_t = \begin{bmatrix} V_t \\ \omega_t \end{bmatrix}$$



Déplacements, conduite différentielle



estimé de la position

avec matlab

```
dt = 0.01;
x(t)=x(t-1) + V(t)*cos(Theta(t-1))*dt;


y(t)=y(t-1) + V(t)*sin(Theta(t-1)*dt;


Theta(t)=Theta(t-1) + Omega(t)*dt;
```

$$x(t) = \int_0^t V(t)\cos(\theta(t))dt$$

$$y(t) = \int_0^t V(t)\sin(\theta(t))dt$$

$$\theta(t) = \int_0^t \omega(t)dt$$

Intégration d'ordre zéro

On voit bien que l'état suivant ne dépend pas des états t-2, t-3, etc…
(propriété de Markov, à venir)

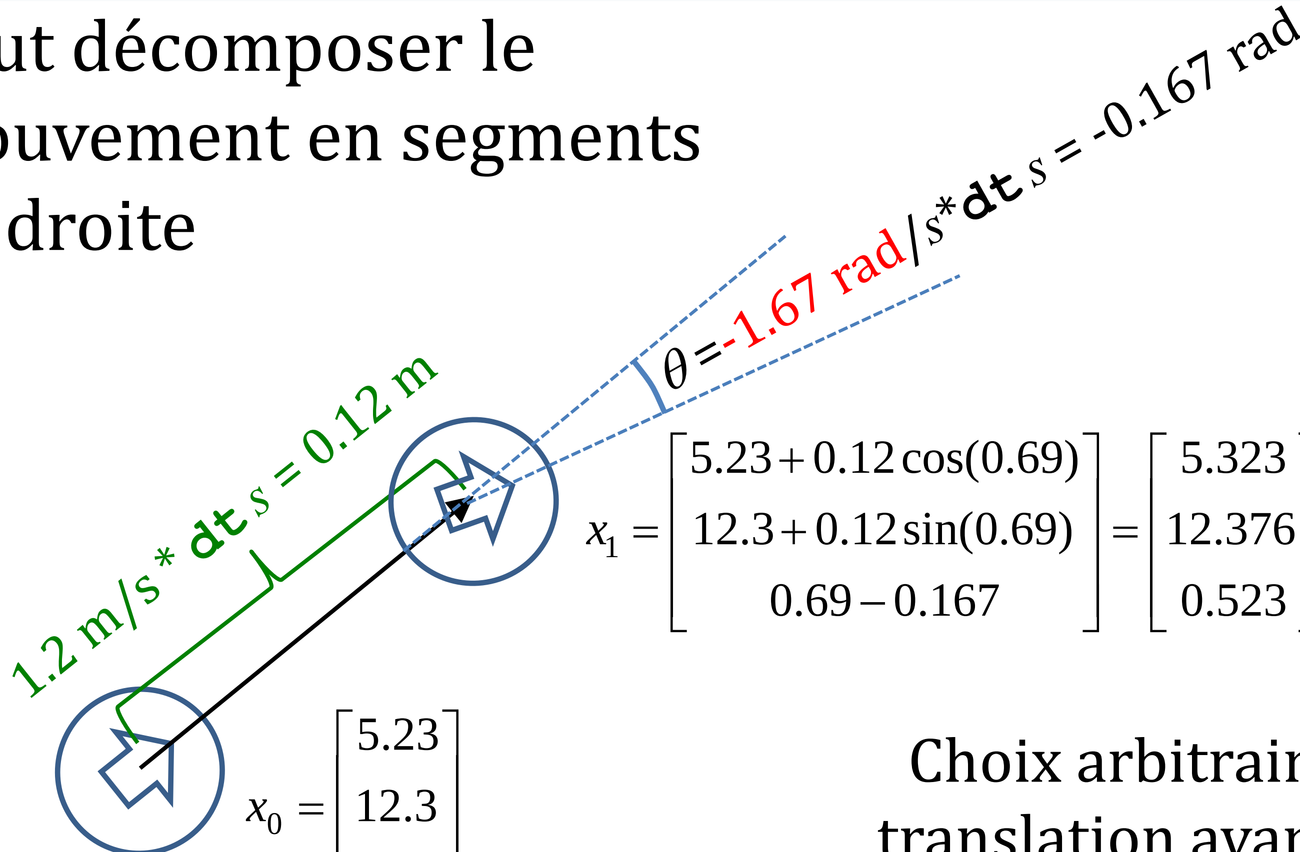
Une commande = un pas

- Peut décomposer le mouvement en segments de droite

pose $x = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$

`dt=0.1` (*secondes*)

$$u_t = \begin{bmatrix} V_t \\ \omega_t \end{bmatrix}$$



$$x_1 = \begin{bmatrix} 5.23 + 0.12\cos(0.69) \\ 12.3 + 0.12\sin(0.69) \\ 0.69 - 0.167 \end{bmatrix} = \begin{bmatrix} 5.323 \\ 12.376 \\ 0.523 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 5.23 \\ 12.3 \\ 0.69 \end{bmatrix}$$

$$u_1 = \begin{bmatrix} 1.2 \text{ (m/s)} \\ -1.67 \text{ (rad/s)} \end{bmatrix}$$

Choix arbitraire de faire la translation avant la rotation.

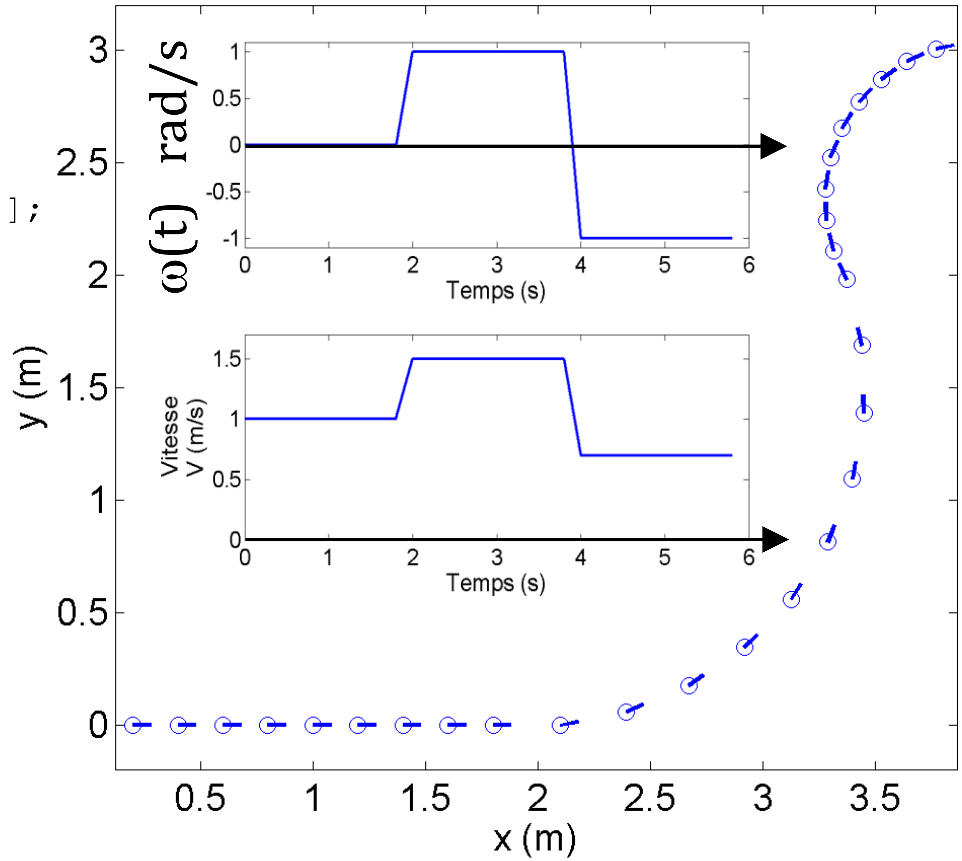
Pour $\Delta t \rightarrow 0$, cela ne fera plus de différence.

Exemple avec *matlab*

```
dt = 0.2; % increment en temps
Omega = [zeros(1,10) 1*ones(1,10) -1*ones(1,10)];
V     = [ones(1,10) 1.5*ones(1,10) 0.7*ones(1,10)];
Time  = (0:(size(V,2)-1))*dt;

% Initialisation
Theta(1) = 0; x(1) = 0; y(1) = 0;

for t = 2:size(V,2)
    x(t) = x(t-1) + V(t)*cos(Theta(t-1))*dt;
    y(t) = y(t-1) + V(t)*sin(Theta(t-1))*dt;
    Theta(t) = Theta(t-1) + Omega(t)*dt;
end
```



exemple complet sur site web: **`ExempleDiffDriveSansBruit.m`**

Solution plus précise

- En considérant un déplacement en arc de cercle plutôt qu'une droite.

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} -\frac{V}{\omega}\sin\theta_t + \frac{V}{\omega}\sin(\theta_t + \omega\Delta t) \\ \frac{V}{\omega}\cos\theta_t - \frac{V}{\omega}\cos(\theta_t + \omega\Delta t) \\ \omega\Delta t \end{pmatrix}$$

Tiré de Probabilistic Robotics, S. Thrun, W. Burgard, D. Fox, p. 127

Processus de Markov

- Pour simplifier les mathématiques, on prétendra que le robot a la propriété de Markov

Processus stochastique, dont la prédiction du futur à partir du présent n'est pas rendue plus précise par des éléments d'information concernant le passé

Comment cela vient modifier les probabilités :

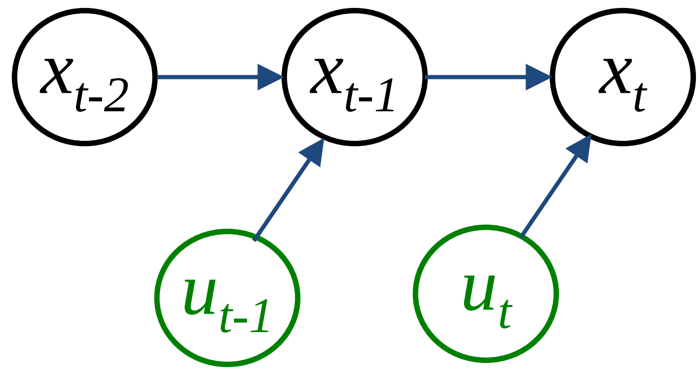
$$P(X_{t+1} = x \mid X_0, u_1, X_1, u_2, X_2, \ldots, u_t, X_t) = P(X_{t+1} = x \mid u_t, X_t)$$

Déplacements probabilistes

- Parce que le monde n'est pas parfait…
- Pour modéliser une commande u_t probabiliste
  - bruits dans les actionneurs
  - perturbations extérieures

$$p(\underbrace{x_t}_{\text{pose actuelle}} \mid \underbrace{x_{t-1}}_{\text{pose précédente}}, \underbrace{u_t}_{\text{commande de déplacement}})$$

Processus de Markov (modèle graphique)



- Modèle **plus réaliste**

Bien identifier le modèle du bruit

- En général, on préfère **surestimer** le bruit dans le modèle que de le **sous-estimer**
- Pensez marge de sécurité!

viaduc de la Concorde

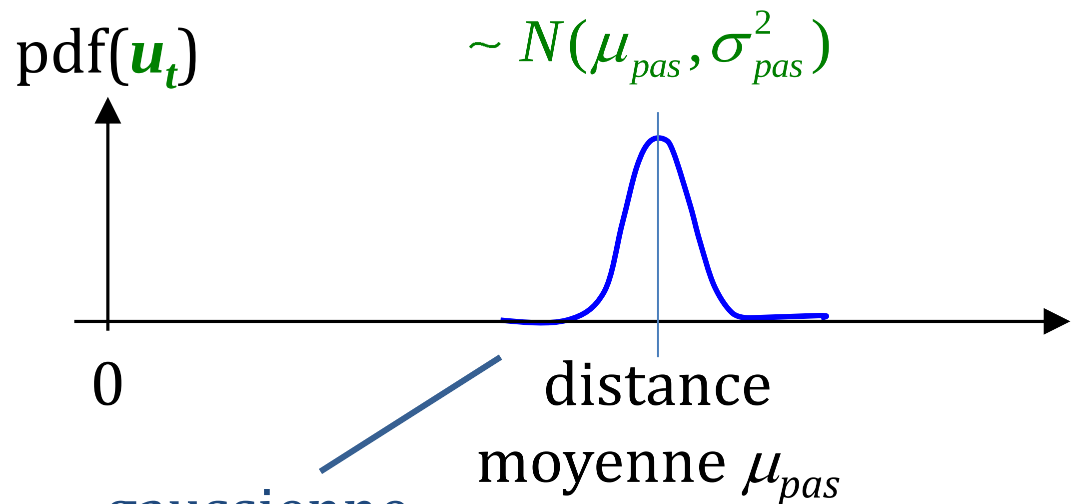
  - $\sigma_{\text{modèle}} > \sigma_{\text{réel}}$: sous-estimer la précision
  - $\sigma_{\text{modèle}} < \sigma_{\text{réel}}$: perte de robustesse des algos (système inconsistant)

(peut-on prévoir toutes les erreurs d'un système?)

UNIVERSITÉ LAVAL

Modèle du déplacement u_t

- Pour un robot 1 D



forme de la distribution = ?

gaussienne

Faire un pas : $$X_{t+1} = X_t + u_t$$

Somme de deux variables aléatoires

gaussienne

Distribution d'une somme de variables aléatoire

- La distribution de probabilité de la somme de deux variables aléatoires indépendantes X_1 et X_2 est la ***convolution*** de leur distribution individuelle

$$X_3 = X_1 + X_2$$

$$p_{X_3}(x) = p_{X_1}(x) * p_{X_2}(x)$$

symbole de la convolution

Convolution

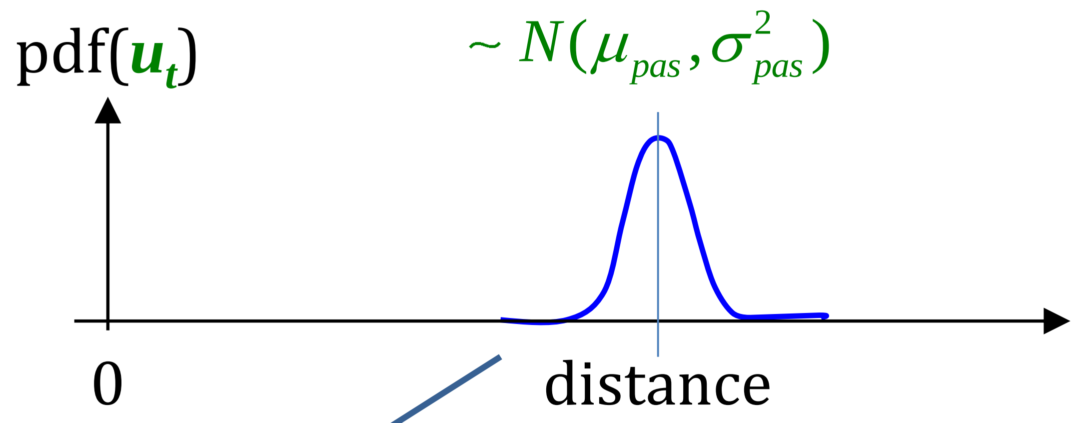
$$\underbrace{f(t) * g(t)}_{\text{2 fonctions en entrée}} = \underbrace{\int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau}_{\text{1 fonction en sortie}}$$

(à titre informatif)

UNIVERSITÉ LAVAL

Modèle du déplacement u_t

- Pour un robot 1 D



moyenne μ_{pas}

Gaussienne!

gaussienne

Faire un pas : $$X_{t+1} = X_t + u_t$$

Somme de deux variables aléatoires

gaussienne

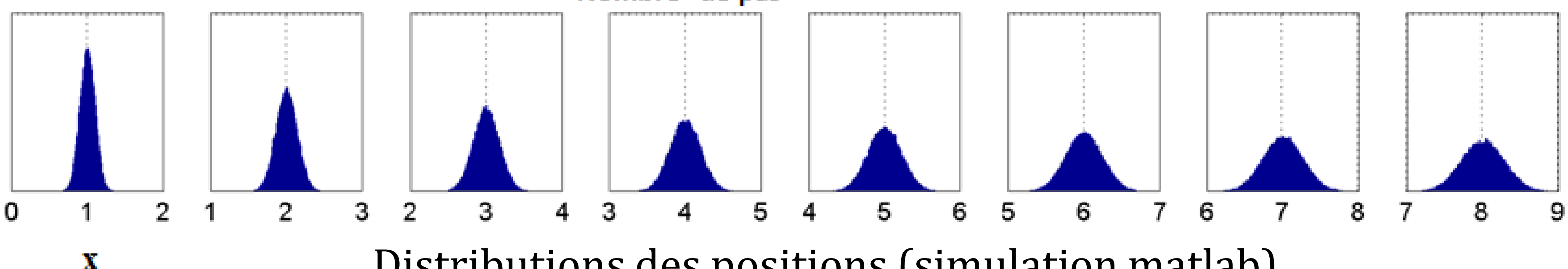
voir `Chapter7_SumOfProbabilities.pdf` sur site du cours

Augmentation de l'incertitude 1D

- 1 pas : $u_t \sim N(\mu_{pas},\sigma_{pas}{}^2)$ *ou* $u_t = \mu_{pas} + \varepsilon_{pas}$, $\underbrace{\varepsilon_{pas} \sim N(0,\sigma_{pas}{}^2)}_{\text{mettre en évidence bruit sans biais}}$
- **Variance** de la position véritable en fonction du nombre de pas *a*:

$$Var\{X+Y\} = Var\{X\} + Var\{Y\} + \cancel{Cov\{X,Y\}}$$

$$\sigma_x{}^2 = a\,\sigma_{pas}{}^2$$



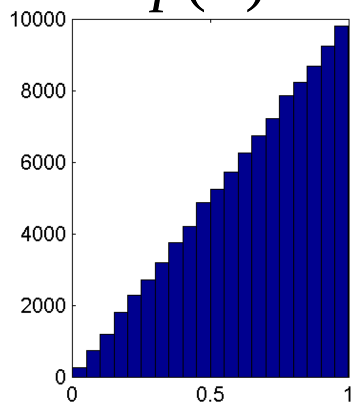
Distributions des positions (simulation matlab)

Erreur gaussienne? Vraiment?

- Mais si $p(u_t)$ n'est pas une distribution gaussienne?
- Théorème central limite :
  - *« toute somme de variables aléatoires indépendantes et identiquement distribuées tend vers une variable aléatoire gaussienne »* *source : wikipédia*

$X = \sqrt{rand}$, avec rand une uniforme entre 0 et 1

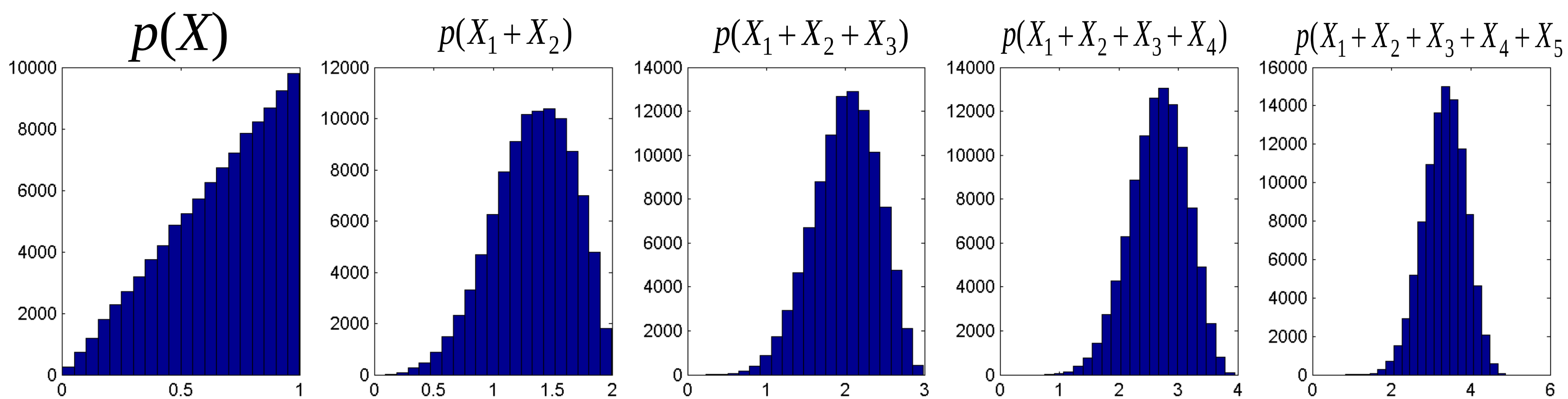
$p(X)$



Erreur gaussienne? Vraiment?

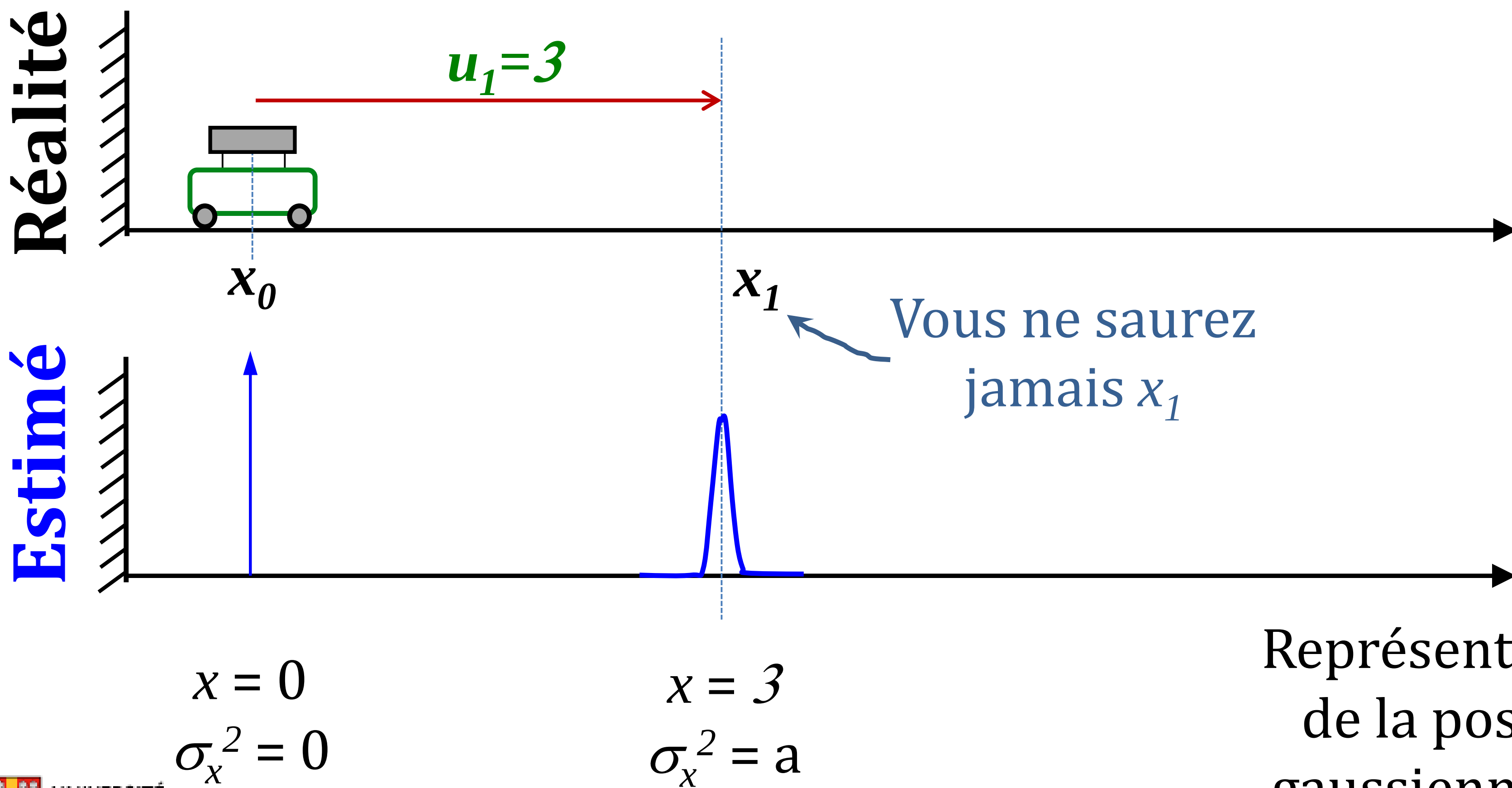
- Mais si $p(u_t)$ n'est pas une distribution gaussienne?
- Théorème central limite :
  - *« toute somme de variables aléatoires indépendantes et identiquement distribuées tend vers une variable aléatoire gaussienne »* *source : wikipédia*

    $X = \sqrt{rand}$, avec rand une uniforme entre 0 et 1



Exemple en 1 D

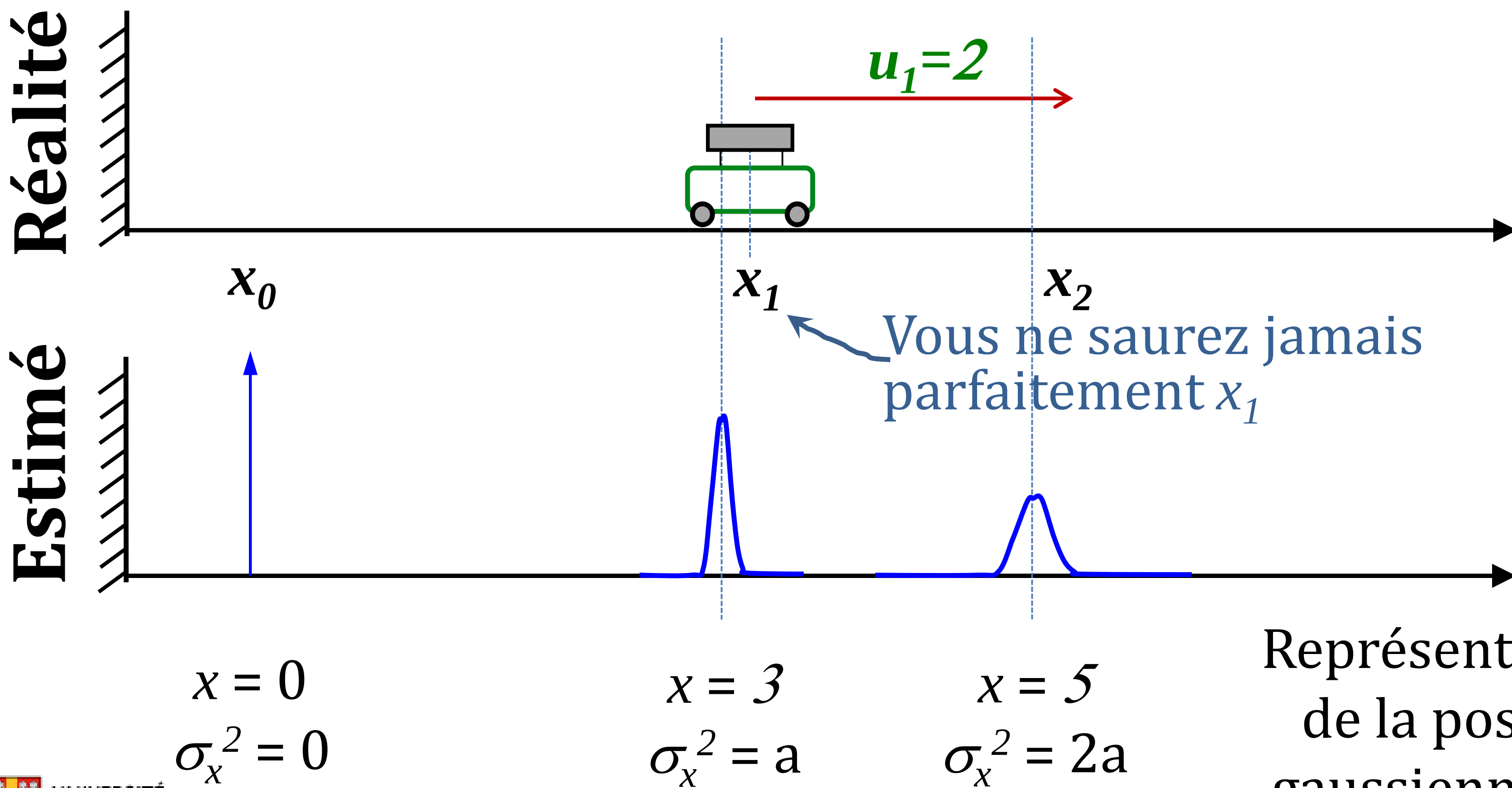
- Position de départ connue parfaitement : $\sigma_x = 0$
- Variance de déplacement est $\boldsymbol{a}$.



Représenter l'estimé de la position par gaussienne : $\boldsymbol{x}$ et σ_x^2

Exemple en 1 D

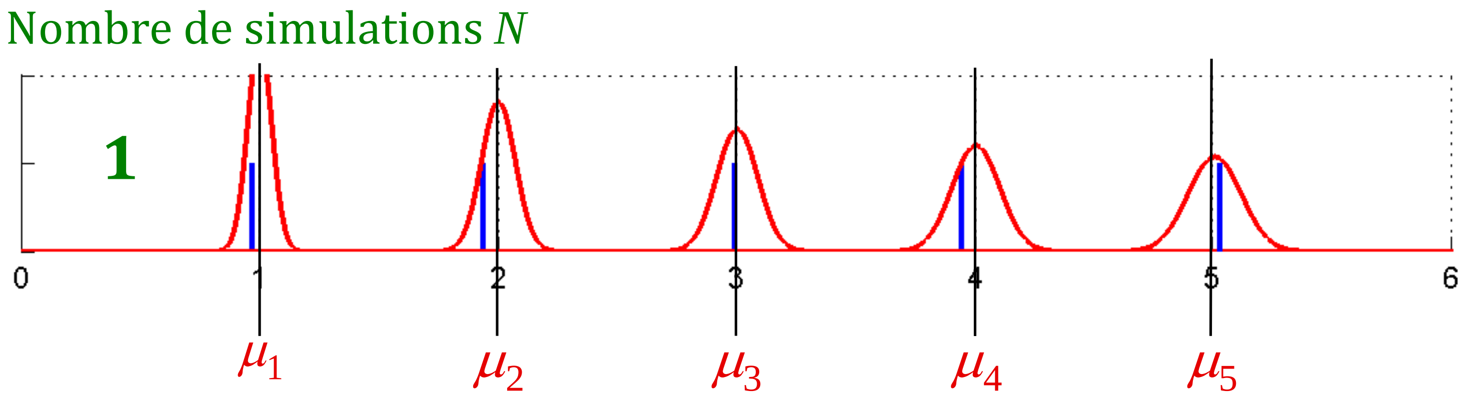
- Position de départ connue parfaitement : $\sigma_x = 0$
- Variance de déplacement est ***a***.



Représenter l'estimé de la position par gaussienne : x et σ_x^2

Exemples répétés de déplacements

En bleu : simulation

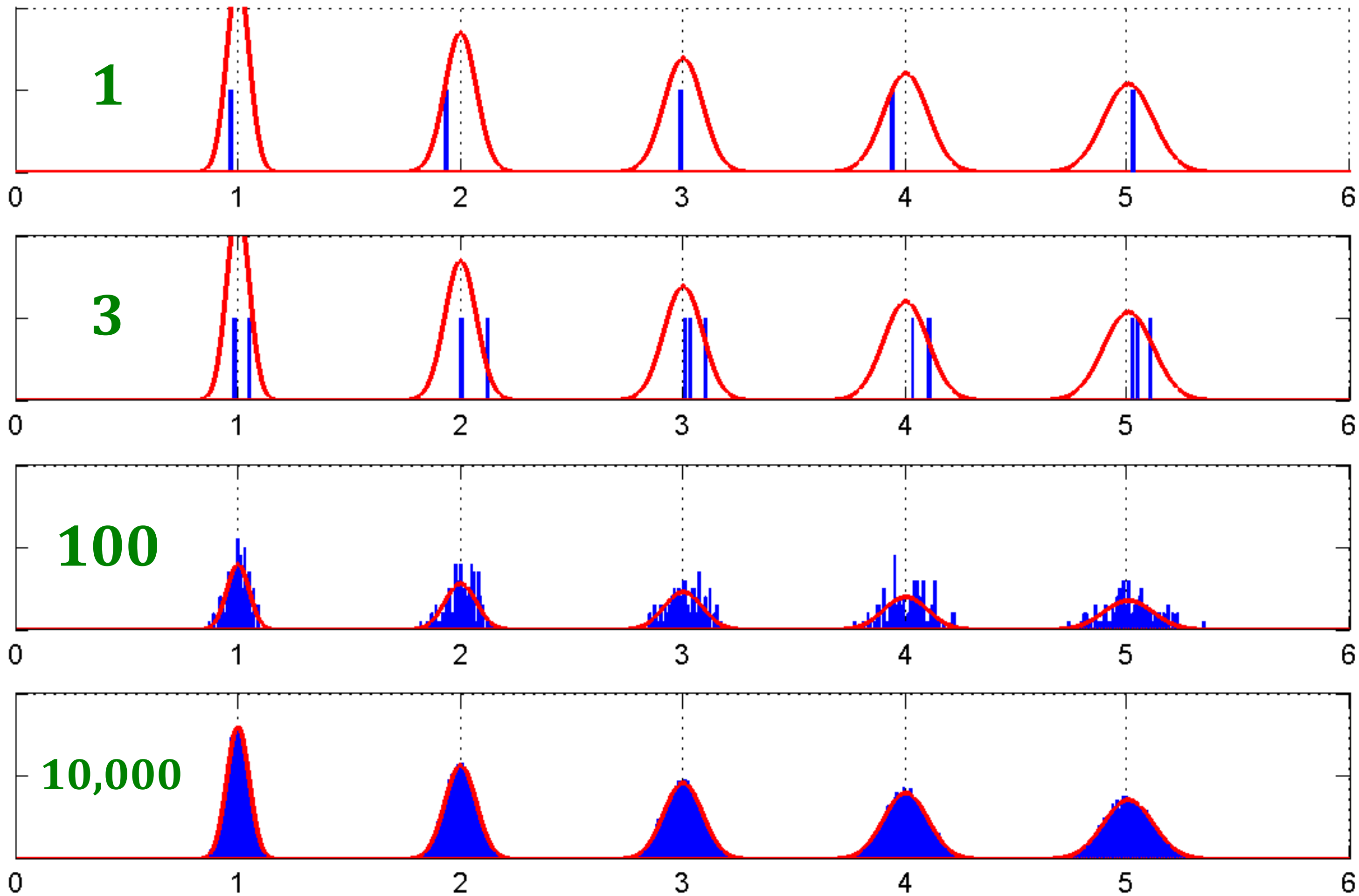


positions que vous estimez vous situer

Exemples répétés de déplacements

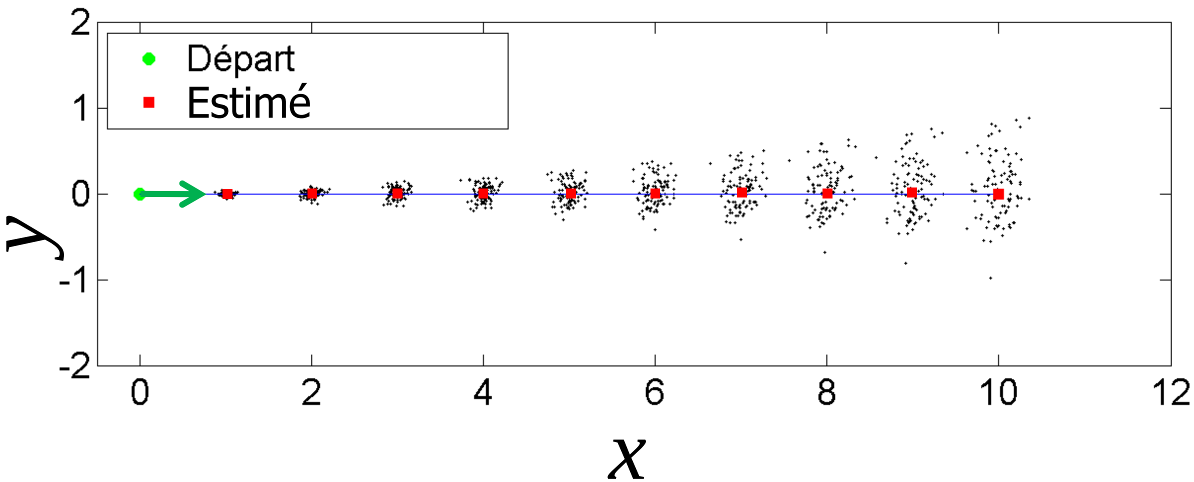
loi des grands nombres : *plus N est grand, on se rapproche de la distribution réelle*

Nombre de simulations *N*



Exemple en 2 D, conduite différentielle

- *Rayons* $r_g = r_d$ et terrain accidenté



Vitesse angulaire $\omega = 0 + \varepsilon_\omega, \quad \varepsilon_\omega \sim N(0, \mathbf{0.02}^2)$

Vitesse linéaire $V = 1 + \varepsilon_V, \quad \varepsilon_V \sim N(0, \mathbf{0.05}^2)$

erreur sur estimé s'accumule en fonction du temps/distance

ExempleClasseDiffDrive.m

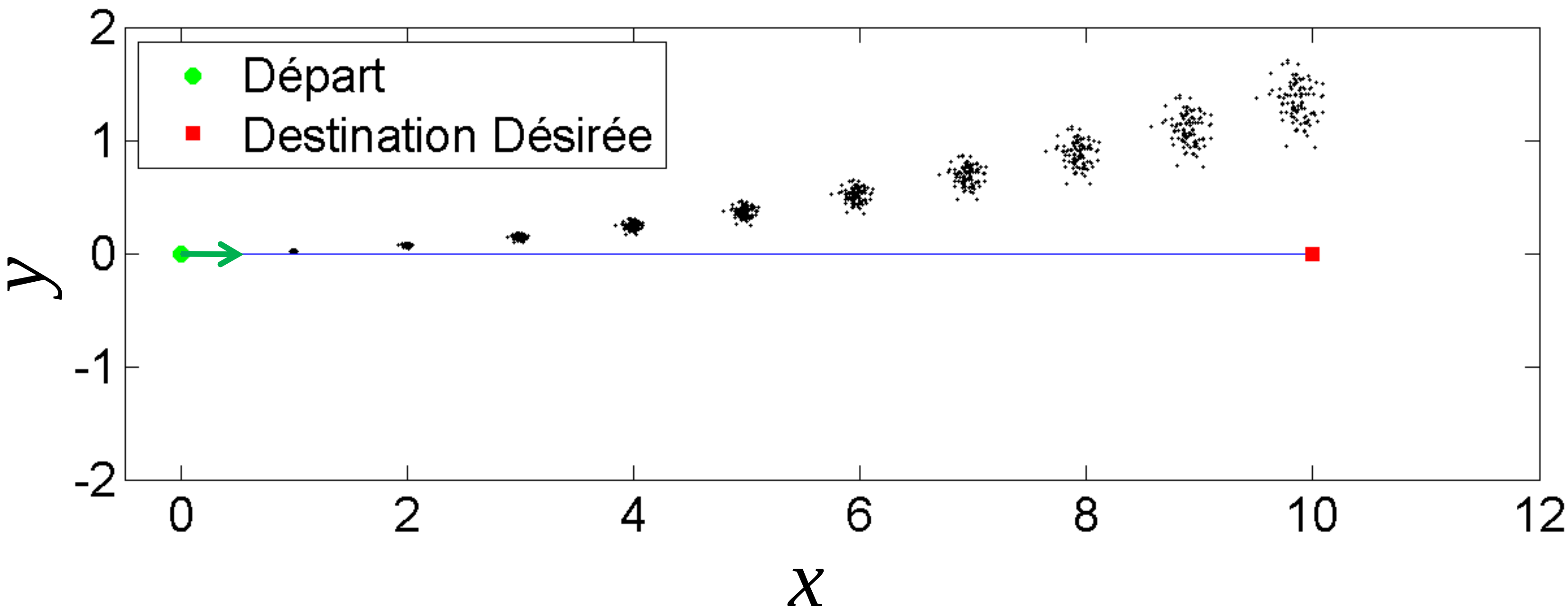
Navigation aveugle : conduite différentielle

- *Rayons* $r_d > r_g$

$$\boxed{\omega \sim N(\mu_\omega, \sigma_\omega^2)}$$

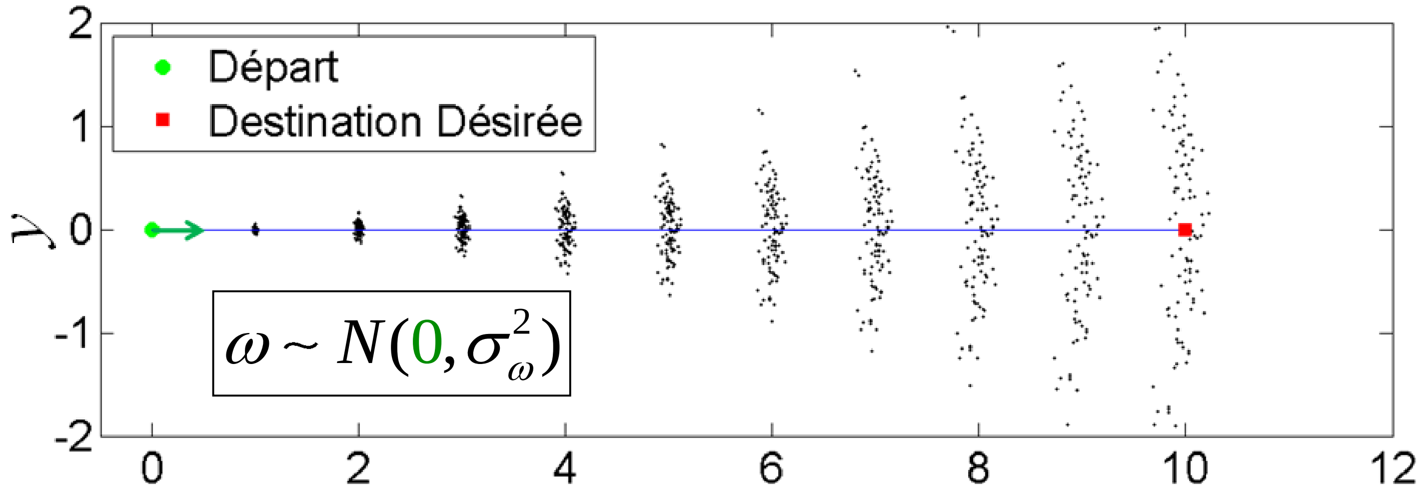
$$\boxed{V \sim N(\mu_V, \sigma_V^{\ 2})}$$

erreur biaisée

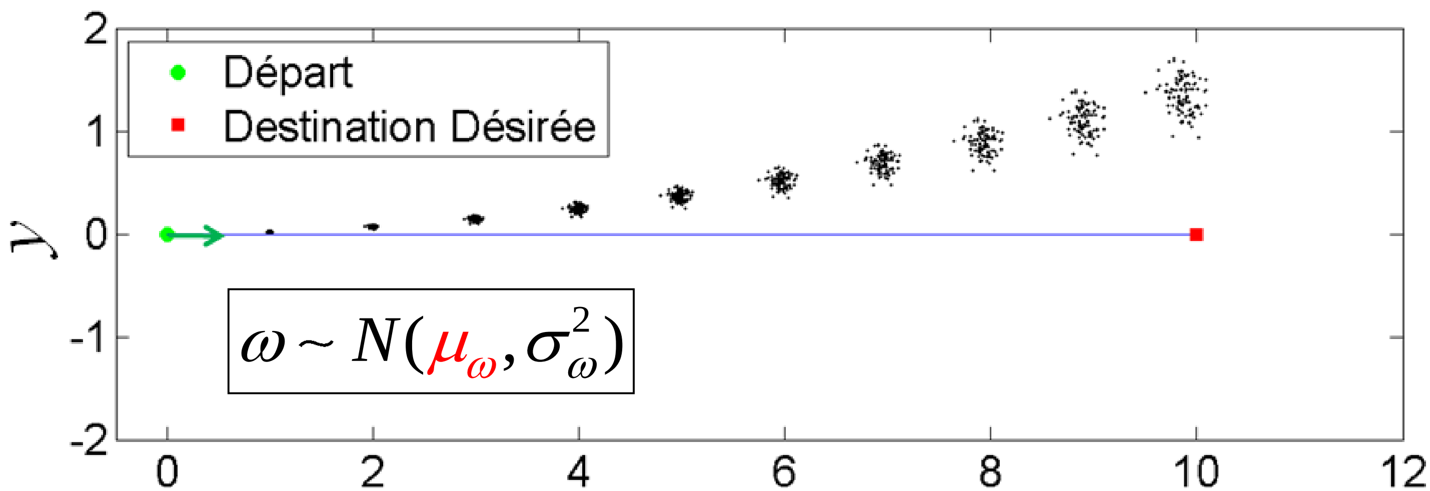


Trajectoire si biais dans l'angle

En « moyenne », on suit la trajectoire demandée



... mais pas si il y a un **biais**...



Bruit sur déplacement u_t toujours fixe?

- Dépendre du type de surface

tapis

céramique

pelouse

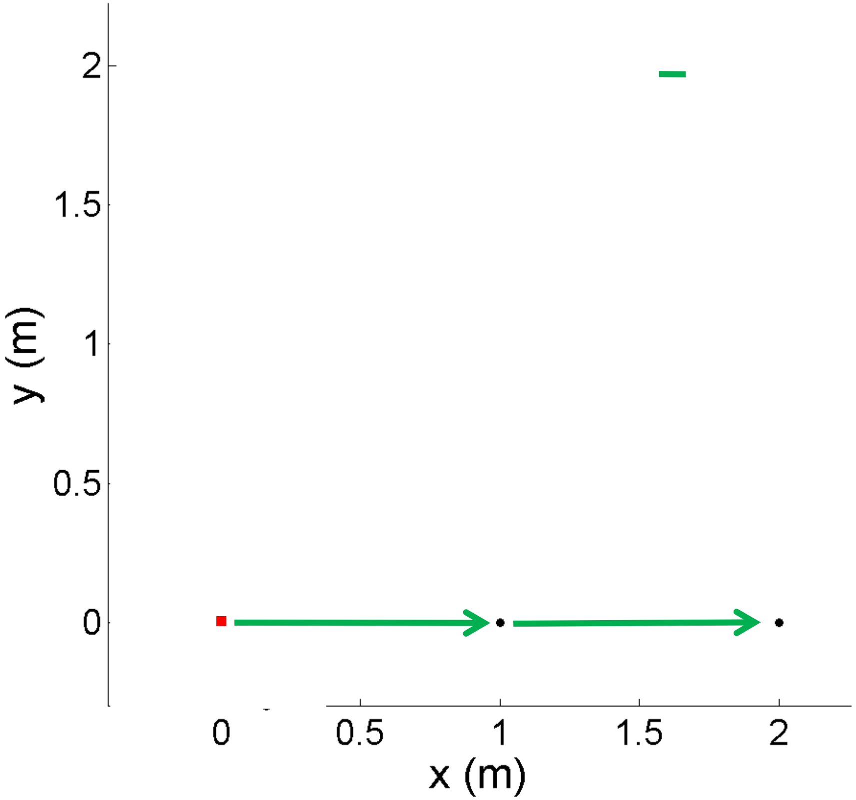
gravier

- État du robot
  - voltage batteries
  - pression des pneus

Exemple en 2 D, erreur commandes

- Erreur de 3% <u>seulement</u> sur :

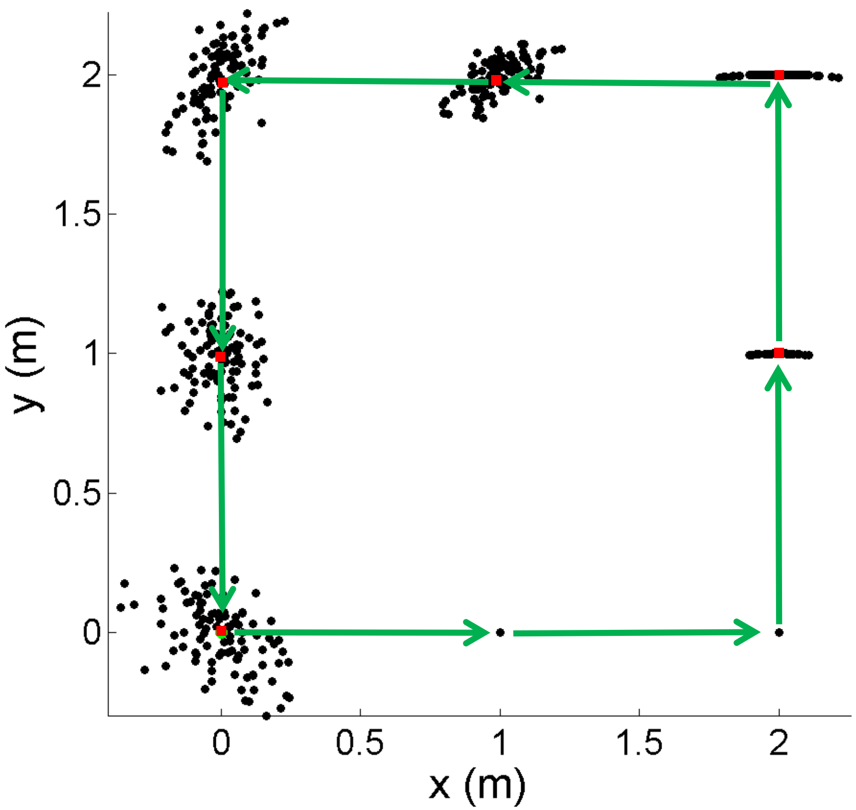
  commande de rotation



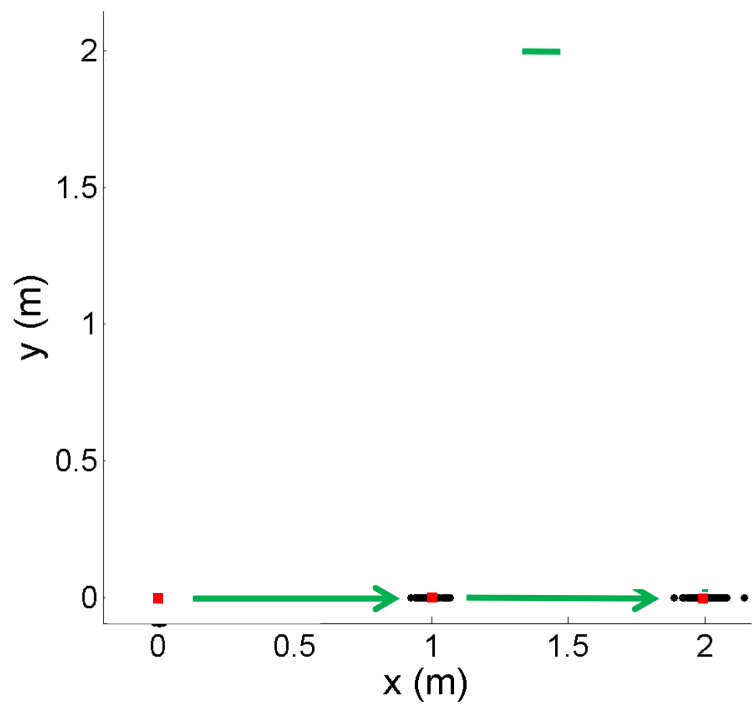
Exemple en 2 D, erreur commandes

- Erreur de 3% <u>seulement</u> sur :

commande de rotation



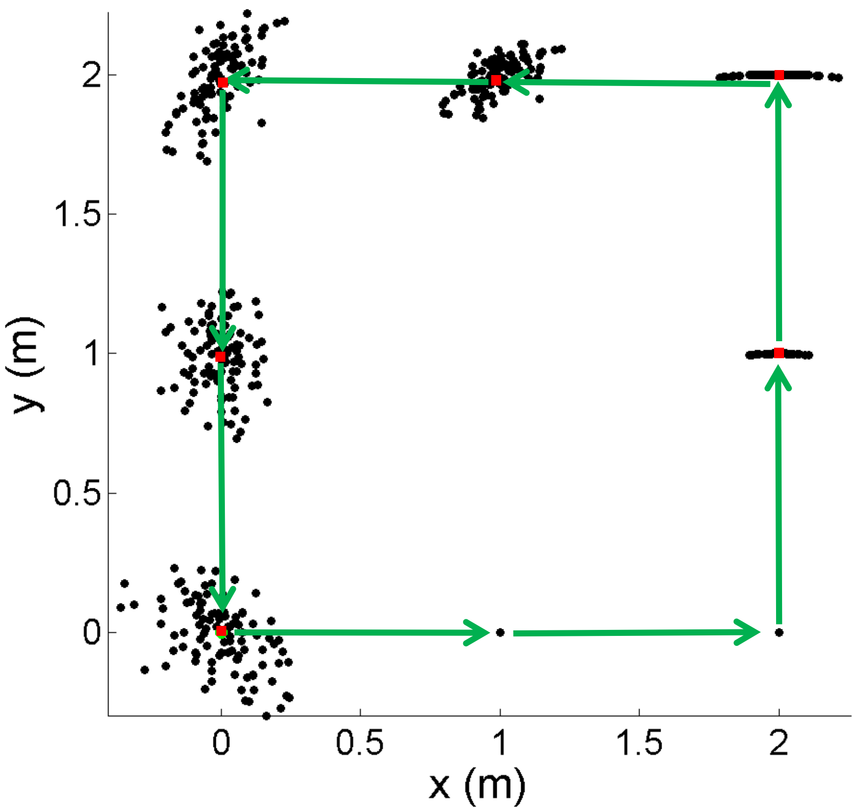
distance



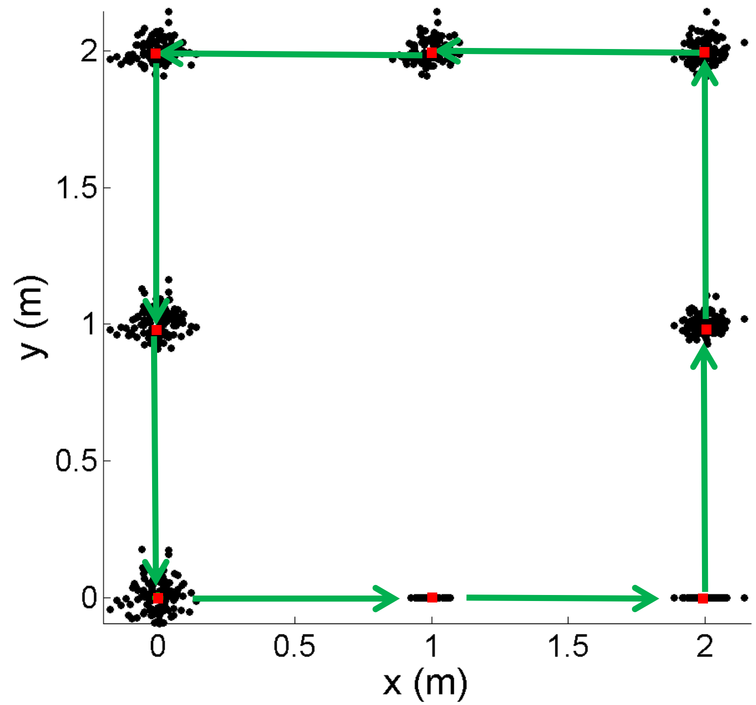
Exemple en 2 D, erreur commandes

- Erreur de 3% <u>seulement</u> sur :

commande de rotation



distance



Navigation à l’aveugle

Navigation aveugle : proprioception

- En anglais : *dead reckoning*
- Principe de base:
  - utilisation de capteurs proprioceptifs
  - pas de point de repères externes
  - angle roues (odométrie)
  - accéléromètre

$$\boxed{x = \int_0^t \int a_x(t)\,dt^2}$$

  - gyroscope

$$\boxed{\theta(t) = \int_0^t \dot{\theta}(t)\,dt}$$

UNIVERSITÉ LAVAL

Pourquoi la navigation à l'aveugle?

- Quand on a pas d'info des capteurs extéroceptifs :
  - temporairement
  - pas de repères externes
    - sous-marin
    - hors-portée des capteurs (gymnase + laser 4 m)
    - GPS dans tunnel/édifices
- Entre les mesures extéroceptives :
  - proprioceptif souvent à 30-300 *Hz*
  - extéroceptif <10 *Hz* (soit capteur, soit temps proces.)
  - on maintient ainsi un estimé de la pose du robot

DARPA Grand Challenge 2005

- Tunnel de 50 mètres : aucun signal GPS

Différence membre/robot mobile

- Pour un membre, l'erreur sur les angles est en général faible
  - *Barrett arm* : répétabilité de 200 μm pour espace de travail de 3.5 m^3.

- Pour robot mobile, l'erreur s'accumule avec les déplacements, lors de la navigation aveugle

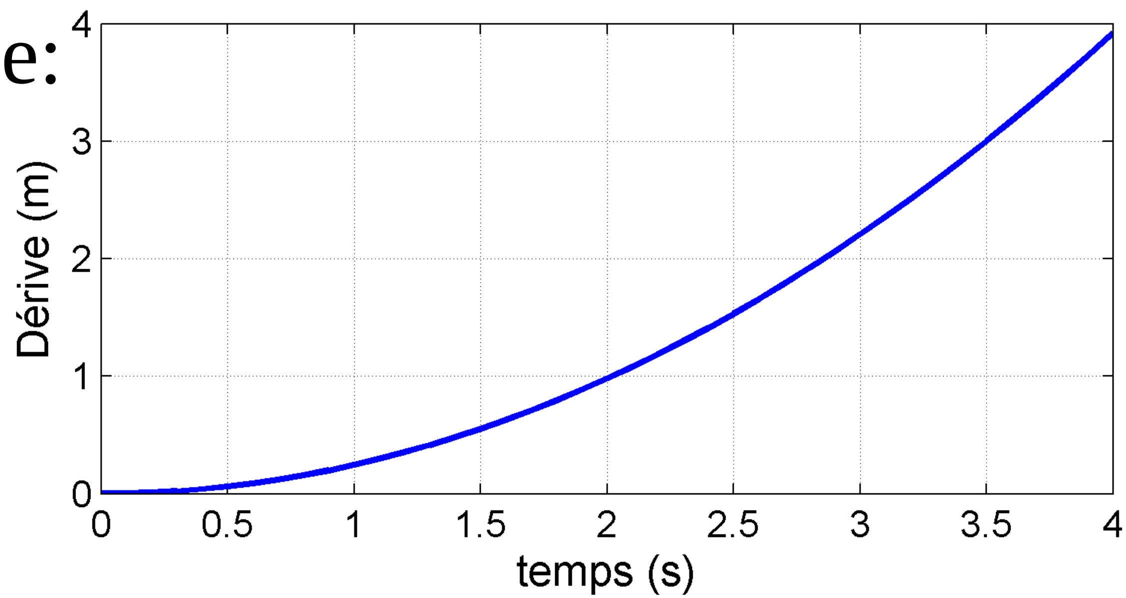
Navigation aveugle : dérive accéléromètre

- Accéléromètre avec erreur systématique 50 *mg*

$$g = 9.8\ m/s^2 \rightarrow 50\ mg = 9.8\ x\ 0.050 = 0.49\ m/s^2$$

$$x = \iint a\ dt^2 = \frac{at^2}{2}$$

- Dérive quadratique:



Source erreur accéléromètre

- Changement de température
- Mauvais alignement ou sol penché :
  - accéléromètre penche de 1° →

    $9.8\ m/s^2 \ \sin(1^o) = 0.17\ m/s^2$
- Différence fabrication
- Pour réduire l'erreur systématique:
  - phase calibration où véhicule est à l'arrêt sur surface plane
    - mesure le biais → erreur systématique
    - implique que le robot peut être immobile (avion? sous-marin?)
  - estime le biais → filtrage Bayésien (Kalman)

UNIVERSITÉ LAVAL

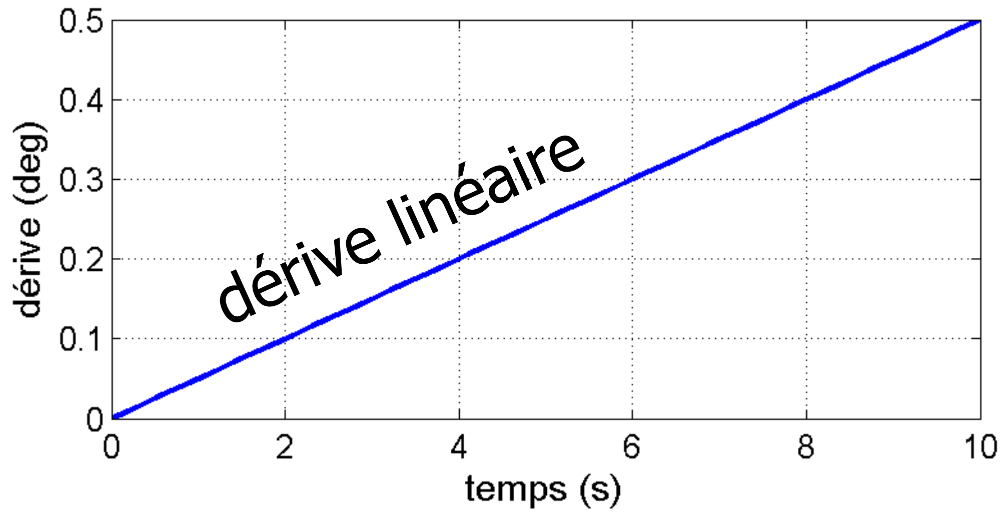
Navigation aveugle : dérive gyroscope

- Angle est l'intégration d'un *angular rate gyro*.

$$\theta_{gyro} = \int \dot{\theta}_{gyro} dt$$

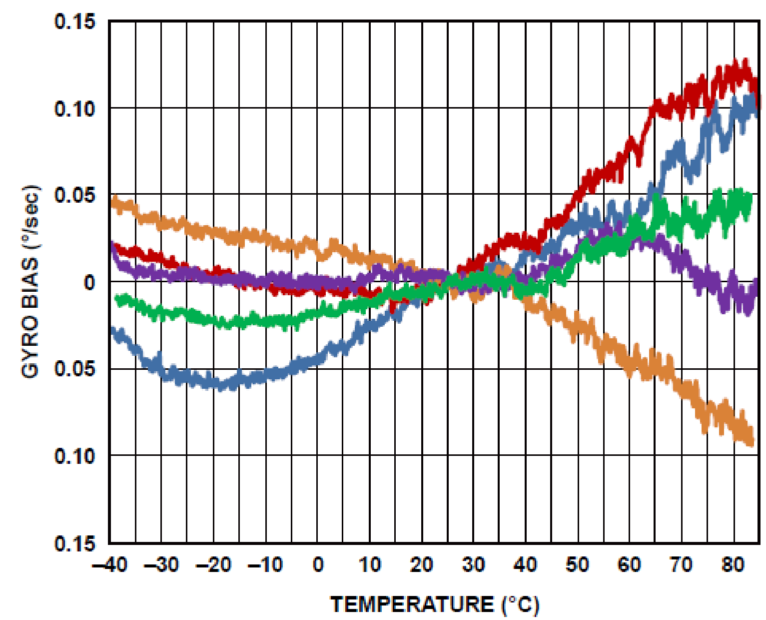
- Avec un biais de 0.05°/s:

erreur =0.05t



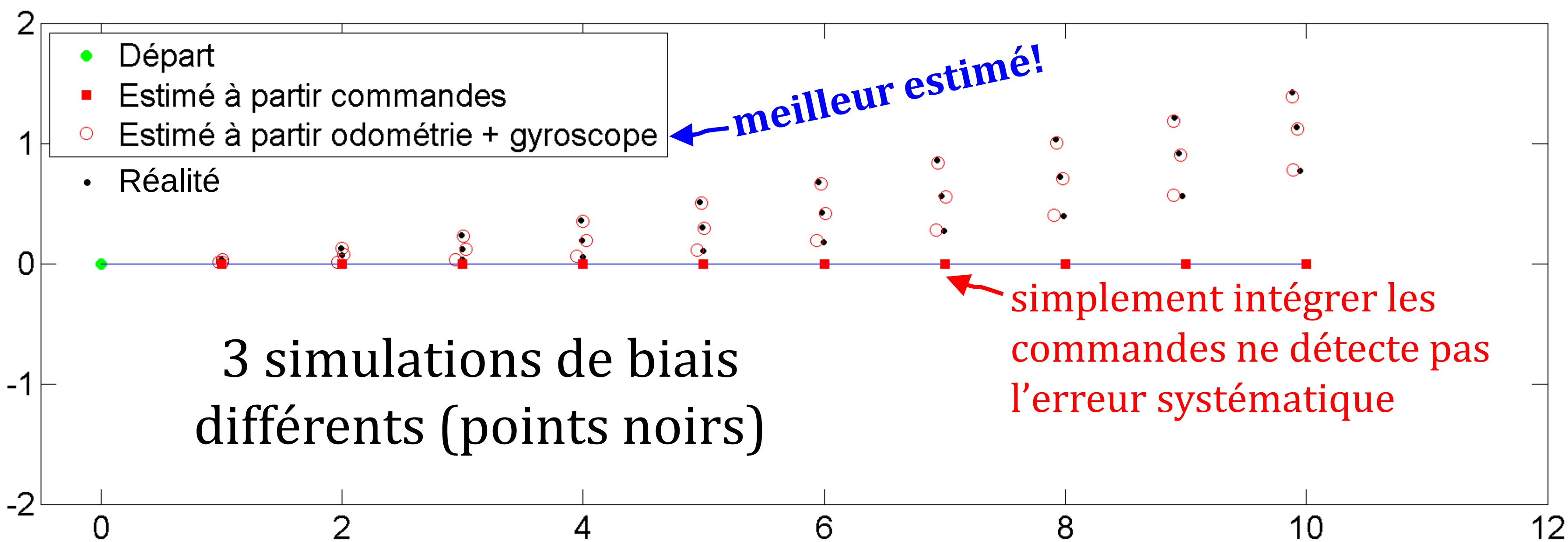
ANALOG DEVICES

±300°/Sec
Precision Angular Rate Sensor
ADIS16135



Odométrie compensée avec gyro

- Problème : roue plus grande → tourne toujours vers la gauche
  + odomètre a du bruit sur la distance (σ =1 %)
- Peut compenser avec un gyroscope (bruit σ = 0.05 °/s)



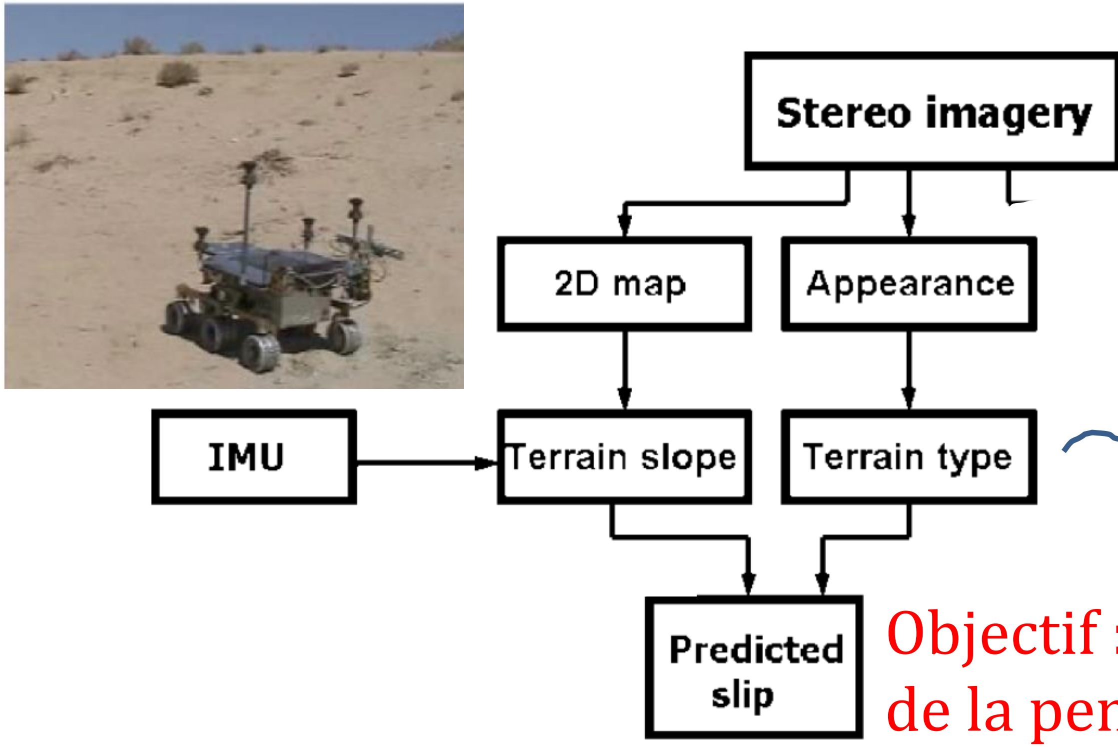
Le gyroscope nous permet de compenser (en partie) les erreurs d'orientation dues aux roues inégales

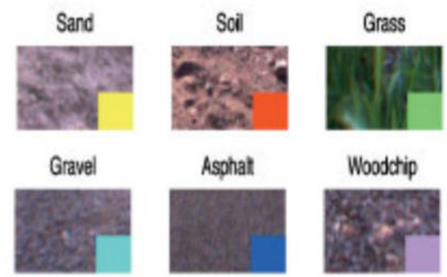
Détection de dérapage

- Dérapage : déplacement ne suit plus commande
- Intéressant de pouvoir le détecter
- Avec les capteurs proprioceptifs, il est possible de détecter si perte de traction
  - diminution du courant moteur pour vitesse angulaire roue constante
  - accélération angulaire des roues pour voltage moteur constant
  - décélération du véhicule
  - différence entre les vitesses angulaires des roues
- Utile pour véhicule dans le sable

Prédiction de la glisse automatique

Utilise extéroceptif





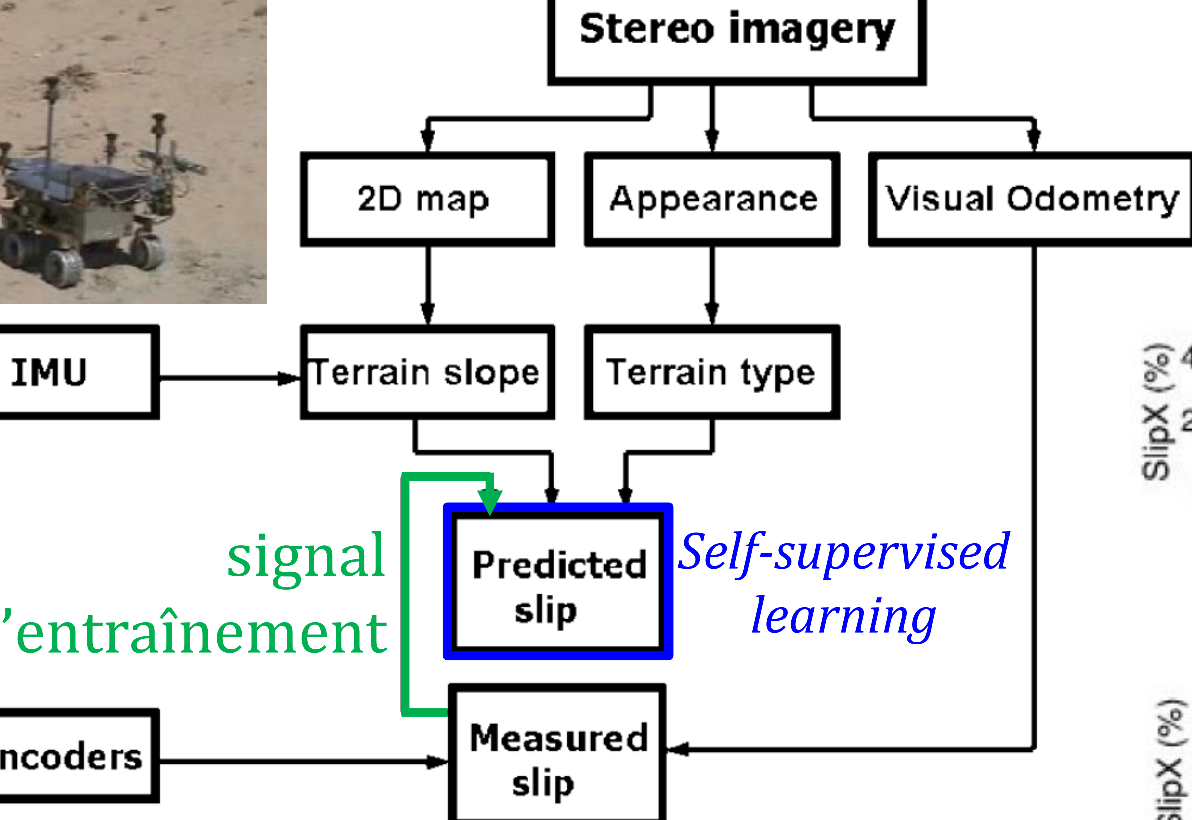
Objectif : prédire la glisse en fonction de la pente estimée et du type de sol

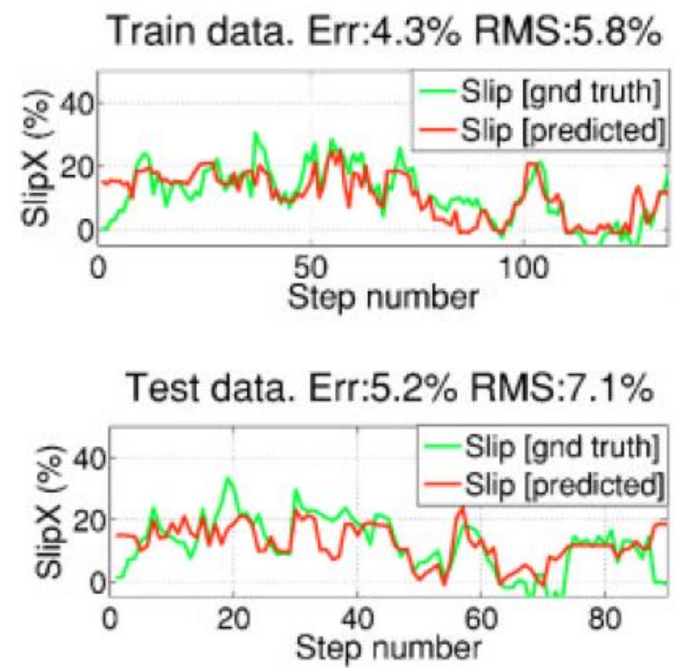
Pour mieux planifier les déplacements en milieu inconnu (mars), en évitant les pentes glissantes

A. Angelova, L. Matthies, D. Helmick, P. Perona. Learning and Prediction of Slip from Visual Information. *Journal of Field Robotics*, 2007.

Prédiction de la glisse automatique

Utilise extéroceptif

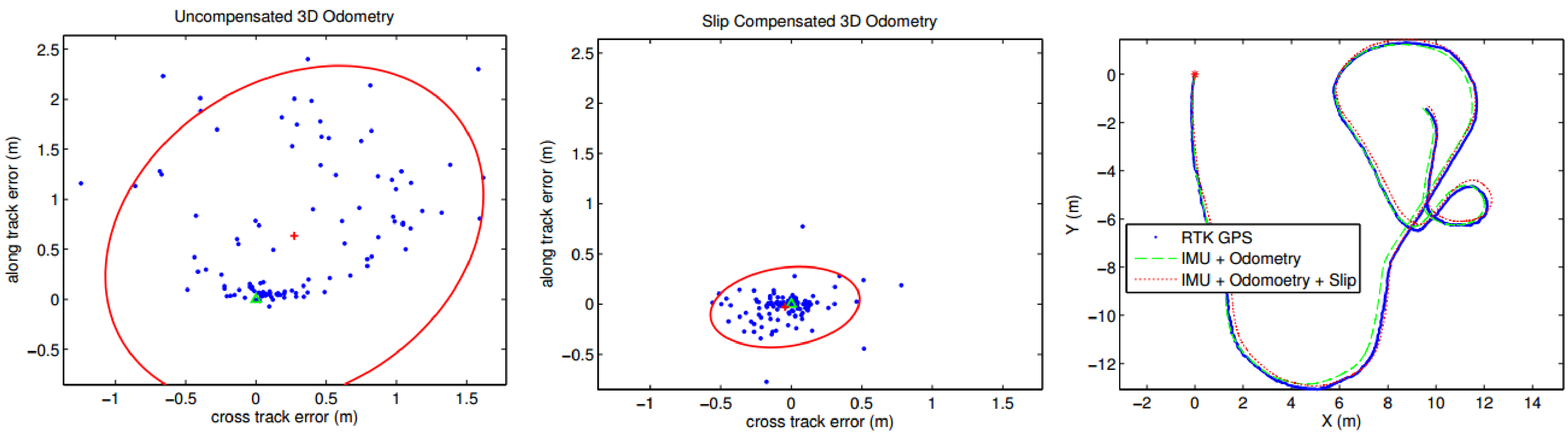




A. Angelova, L. Matthies, D. Helmick, P. Perona. Learning and Prediction of Slip from Visual Information. *Journal of Field Robotics*, 2007.

Estimation en ligne de la glisse

- Basé sur un filtre de Kalman qui estime le taux de glisse α



Aiding Off-Road Inertial Navigation with High Performance Models of Wheel Slip
Forrest Rogers-Marcovitz, Michael George, Neal Seegmiller, and Alonzo Kelly, *IROS* 2012

Projet AUDI TTS automatisé

- Pour faire le trajet de la course *Pikes Peak*

Parking avec du panache!

- ICRA 2010

Pour réduire erreur estimé position

- Utiliser des points de repère externes <u>fixes</u>
  - environnement statique vs. dynamique
- Repère visuel naturel
  - couleur/forme distincte
  - SIFT, SURF, BRIEF, BRISK
- Repère visuel artificiel
- Repère scan laser

CAPTEURS EXTÉROCEPTIFS

Examen

Examen : contenu

Concepts généraux

- Connaître théorème de Thalès
- Maîtriser le théorème de l'angle inscrit et de l'angle au centre. En particulier, savoir calculer rayon et position du cercle de la contrainte à partir des points de repères et des angles alpha
- Comment passer d'un système de coordonnées cartésien ←→ polaire
- Savoir faire les opérations de base avec les matrices et vecteurs
- Être capable de manipuler les fonctions trigonométriques et travailler avec les triangles
- Savoir approximer les fonctions trigonométriques sin et cos
- Connaître le concept du problème mal posé, et pouvoir en les identifier ou donner des exemples
- Pouvoir dériver des polynômes (exposants positifs et négatifs) et la règle du produit pour les dérivations de fonction : $(f \cdot g)' = f' \cdot g + f \cdot g'$.
- Difficulté du problème de *data association*

Examen : contenu

Capteurs

- Connaître des différentes technologies de capteurs, leurs limitations et leurs applications
- Comprendre la différence entre les capteurs proprioceptifs et extéroceptifs, et pouvoir classer des capteurs dans ces catégories
- Problématique des biais pour certains capteurs inertiels : dérive linéaire (angle) et quadratique lors de la navigation à l'aveugle
- Être capable de faire des problèmes de linéarisation
- Connaître la différence entre une fonction bijective et non-bijective.
- Comprendre la notion de sensibilité pour un capteur

Examen : contenu

Capteurs visuels et coordonnées homogènes

- Connaître les différents types de projections;
- Savoir estimer la focale *f* d'une caméra (calibration simpliste);
- Pouvoir calculer la position sur le plan image d'un point dans l'espace 3D;
- Savoir calculer les angles entre des objets dans une image;
- Savoir se localiser en utilisant des repères visuels dans une image et une carte; en particulier être capable résoudre le problème de triangulation (localisation) de repères visuels avec compas et règle. Voir le vidéo suivant, au besoin : http://youtu.be/Md6rryUmtBE
- Comprendre comment fonctionne les caméras stéréo, pouvoir calculer les distances en z des objets dans un environnement, à partir d'une paire d'image stéréo; comprendre le problème de la correspondance.
- Comprendre en surface les caméra stéréo actives (Kinect 1, Kinect 2)
- Pouvoir utiliser les transformations géométriques (matrices *T* et *R*) et les coordonnées homogènes. Ces matrices seront données à l'examen, vous n'avez donc pas besoin de les apprendre par cœur.
- Être capable de transférer les coordonnées d'un point d'un repère à un autre;
- Fonctionnement des détecteurs de coin Harris, FAST
- La notion de points de repère visuel (naturel ou fiduciaire).
- Fonctionnement des descripteurs BRIEF, ORB
- Appariement des descripteurs/features d'une image à l'autre
- Odométrie visuelle
- Pouvoir appliquer l'algorithme RANSAC sur un problème simple

Examen : contenu

Probabilité

- Être suffisamment familier avec les probabilités (distributions normales, écart-types, variances, indépendance, variable corrélé/non-corrélé, somme de variables aléatoires)
- Appliquer le théorème de Bayes
- Comprendre la notion de prior $p(x)$
- Comprendre la fonction de vraisemblance (measurement likelihood) $p(z|x)$, et pouvoir en créer une suivant une description en mot d'un système

- Mettre à jour des distributions discrètes (exactement) ou continue (tracé à main levée) suite à des mesures
- Produit de gaussienne = gaussienne

Examen : contenu

Locomotion

- Pouvoir travailler avec n'importe quelle configuration de robots à roues, et calculer les vitesses linéaires de chaque roue, vitesse angulaire ω du robot, et position du centre instantané de rotation ICC
- Équation d'intégration d'ordre zéro
- Se rappeler que la position d'un gyroscope n'influence pas le taux de rotation mesuré (son orientation peut influencer si l'axe de sensibilité n'est plus dans la même orientation que la rotation mesurée).

Autres

- Bien entendu, vous devez bien comprendre les problèmes des TP.
- Pouvoir refaire certaines démonstrations des équations de base en robotique