Suivi de routes multi hypothèses.

Johan MATHE

SY27

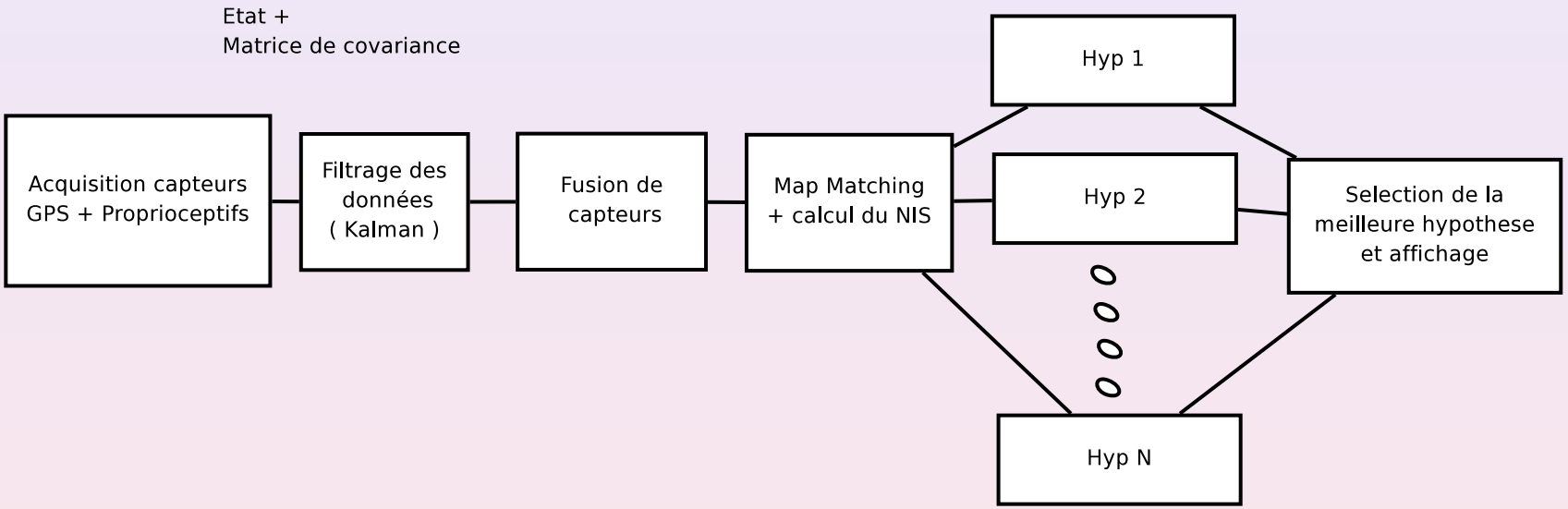
16 janvier 2008

Problématique - définition des objectifs

Objectifs

- Mettre en place un système de suivi de routes
- Par rapport à une carte
- Multi hypothèses

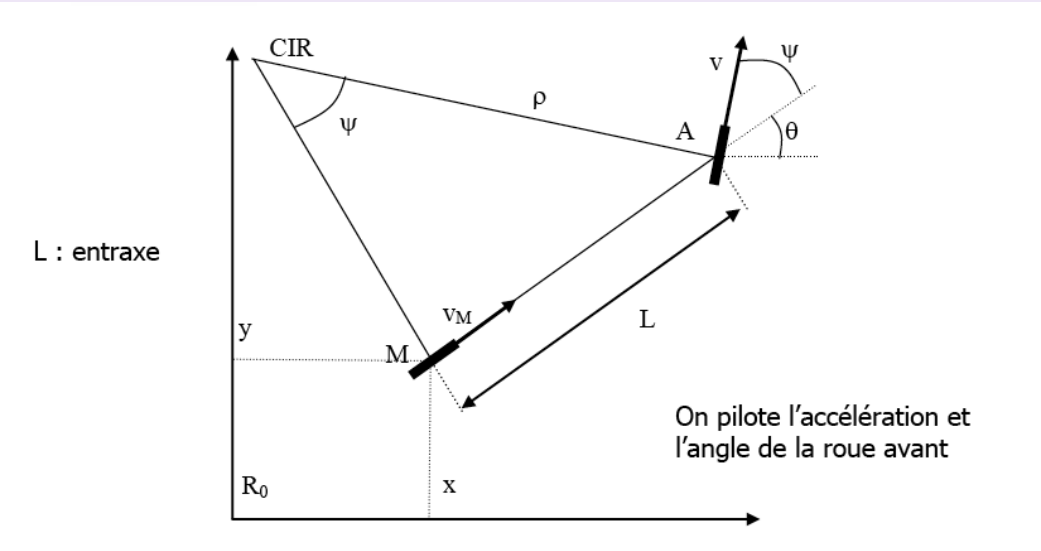
Synotpique de l'application



Modèle utilisé

Modélisation

- Modèle vélocipède
- Représentation d'états



Filtrage

- Nécessité d'implémenter un filtrage des données
- Filtrage de Kalman

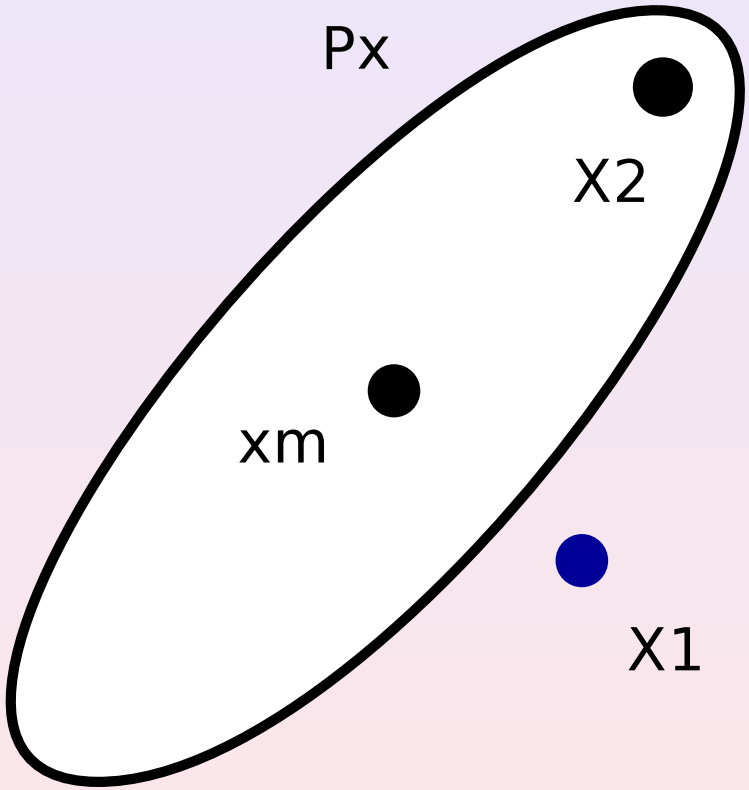
$$F_k = \left.\frac{\partial f}{\partial X}\right|_{X_k} = \begin{bmatrix} 1 & 0 & -v_k \sin(\theta_k) \\ 0 & 1 & v_k \cos(\theta_k) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_{k+1} + T_e v_k \cos(\theta) + \alpha_X \\ Y_{k+1} + T_e v_k \sin(\theta) + \alpha_Y \\ \theta_{k+1} + \alpha_\theta \end{bmatrix} = f(X_k) + Q_A$$

Fusion multicapteurs

- Besoin de fusionner les informations des capteurs
- Utilisation de la distance de Mahalanobis D
- Correspond à une normalisation par la variance
- Suit une loi χ^2

$$D^2 = (x - m_x)^T P^{-1} (x - m_x)$$



Simplification

- Simplification de la matrice de covariance

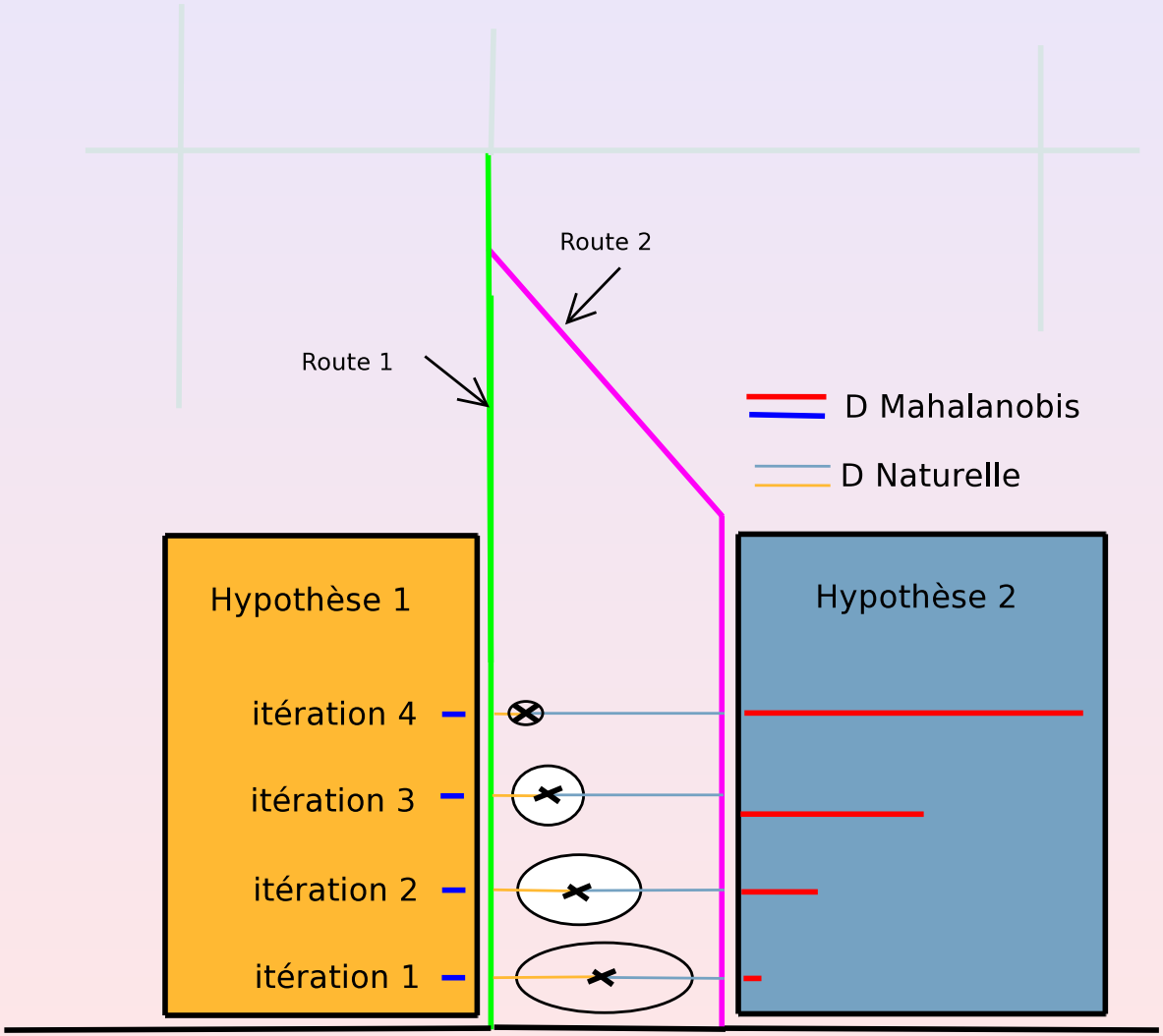
$$P = \begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y & 0 \\ \sigma_x\sigma_y & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \equiv P_s = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_r^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

- La distance de Mahalanobis devient

$$D^2 = X^T P_s^{-1} X = \frac{d_{abs}^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2} = NIS \sim \chi^2(2)$$

X_k VTA gaussien, donc NIS est un χ^2 d'espérance k et de variance $2k$.

Illustration



Multi hypothèses

Objet mathématique contenant :

1. Une route sur la carte
2. Une valeur de NIS
3. Une valeur de notre indicateur statistique W

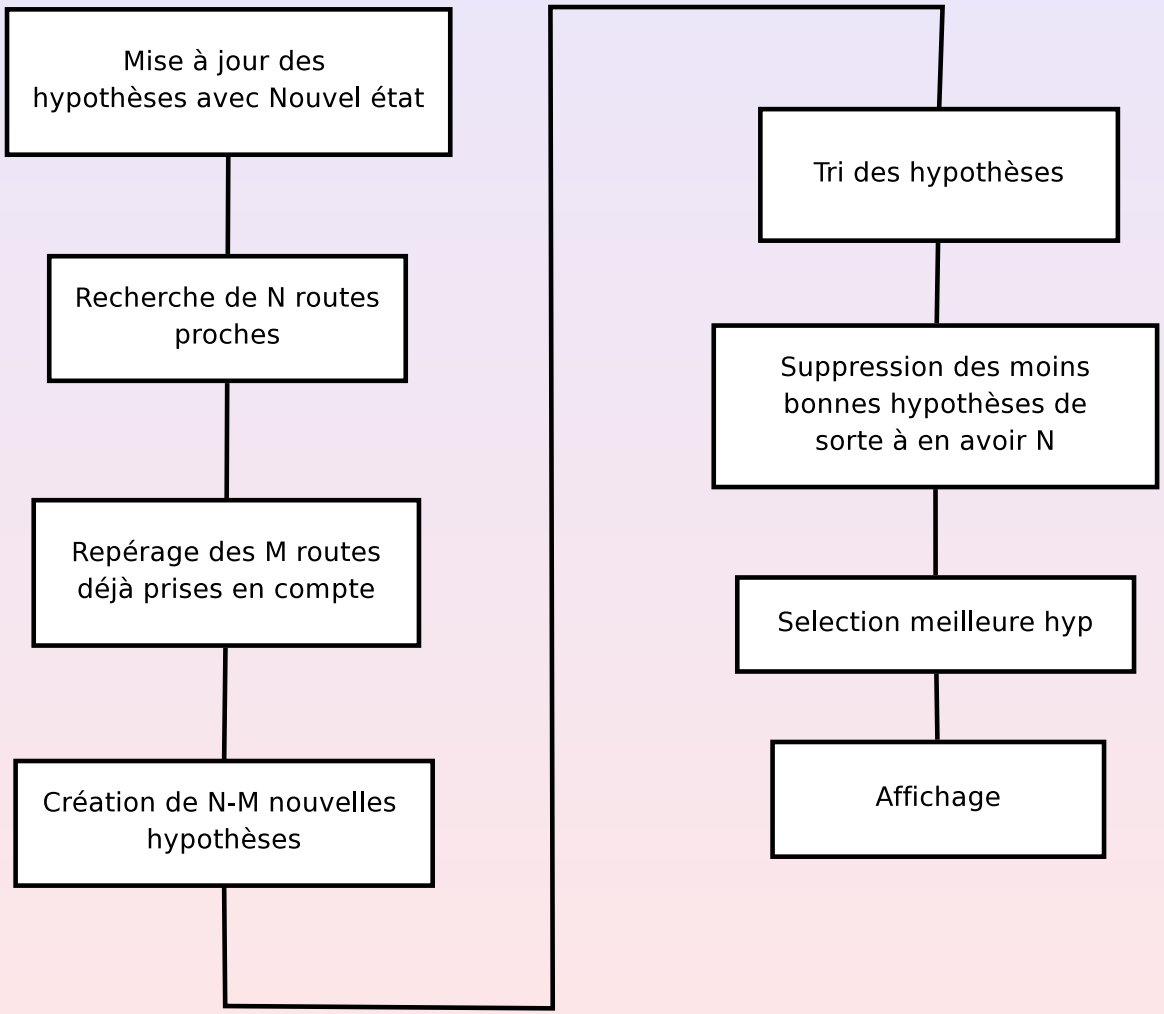
Indicateur statistique

Objectifs

- donne une "inertie" à une hyp
- α coefficient d'inertie
- $w_i[k]$ indicateur
- $W_i[k]$ indicateur normalisé

1. $w_i[k+1] = W_i[k]e^{-\alpha NIS_i[k]}$
2. $\log(w_i[k+1]) = \log(W_i[k]) - \alpha NIS_i[k]$
3. $W_i[k] = \frac{w_{i[k]}}{\sum_{i=1}^{N} w_i[k]}$

Algorithme général



Implémentation + Démo

Bibliothèques utilisées

- Langage C++
- QT (interface graphique)
- OpenGL + GLU (rendu 2D 3D)
- STL+Boost (formats + tri)