

Compte rendu TP Localisation

Yichen ZHU

November 18, 2017

1 Localisation avec ancrs

On veut connaître la position d'un capteur ($\theta = (x, y)$) à partir des M ancrs autour du capteur [$z_1 = (\alpha_1, \beta_1) \cdots z_M = (\alpha_M, \beta_M)$]. On connaît toutes les distances entre les ancrs et le capteurs ($d_1 \cdots d_M$) On a donc :

$$\begin{aligned} d_k^2 &= \|\theta - z_k\|^2 \\ &= \|\theta\|^2 - 2 \langle z_k, \theta \rangle + \|z_k\|^2 \end{aligned}$$

Pour k allant de 2 à M

$$\begin{aligned} d_k^2 - d_1^2 &= -2 \langle z_k - z_1, \theta \rangle + \|z_k\|^2 - \|z_1\|^2 \\ &= -2 \langle z_k - z_1, \theta \rangle + (\alpha_k^2 - \alpha_1^2 + \beta_k^2 - \beta_1) \end{aligned}$$

On peut résumer cette relation par :

$$\Phi \theta = b - \delta$$

$$\text{avec } \Phi = 2 \begin{pmatrix} \alpha_2 - \alpha_1 & \beta_2 - \beta_1 \\ \vdots & \vdots \\ \alpha_M - \alpha_1 & \beta_M - \beta_1 \end{pmatrix} \quad b = \begin{pmatrix} \alpha_2^2 - \alpha_1^2 + \beta_2^2 - \beta_1 \\ \vdots \\ \alpha_M^2 - \alpha_1^2 + \beta_M^2 - \beta_1 \end{pmatrix} \quad \delta = \begin{pmatrix} d_2^2 - d_1^2 \\ \vdots \\ d_M^2 - d_1^2 \end{pmatrix}$$

En observation, on mesure les puissances autour des ancrs avec un bruit pour estimer les distances. Une fois on a les distances estimées ($\hat{d}_1^2 \cdots \hat{d}_M^2$) on a notre $\hat{\delta}$ d'où on établit la nouvelle équation :

$$\Phi \theta = b - \hat{\delta}$$

On cherche le θ qui satisfait toutes équations pour k allant de 2 à M mais on ne peut pas le résoudre parce que avec M équations, deux inconnus, la solution n'est pas unique. On veut donc estimer θ au sens moindres-carrés c'est à dire on cherche un θ qui minimise : $\|\Phi \theta - (b - \hat{\delta})\|^2$. Donc :

$$\hat{\theta} = \arg \min_{\theta} \|\Phi \theta - (b - \hat{\delta})\|^2$$

On calcule le gradient

$$\begin{aligned}
0 &= \nabla_{\theta} (\|\Phi\theta - (b - \hat{\delta})\|^2) \\
&= \Phi^T (\Phi\theta - (b - \hat{\delta})) \\
\Rightarrow \Phi^T \Phi \hat{\theta} &= \Phi^T (b - \hat{\delta}) \\
\hat{\theta} &= (\Phi^T \Phi)^{-1} \Phi^T (b - \hat{\delta})
\end{aligned}$$

On implémente cette méthode en Matlab et on joue avec les paramètres M (nombre d'ancres) et T (nombre de mesures) :

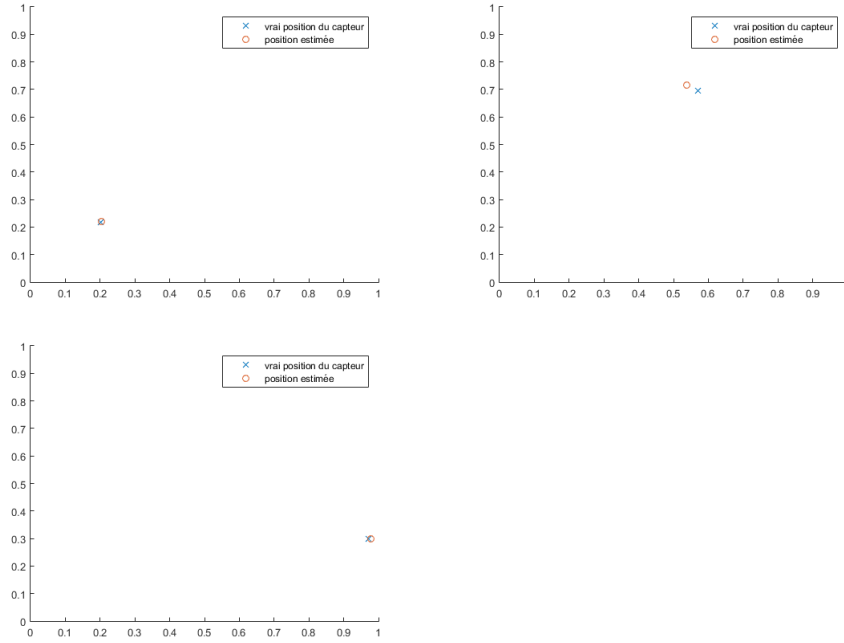


Figure 1: $M = 5, T = 100$, $M = 5, T = 10$ et $M = 10, T = 10$

On remarque quand on a plus d'ancres ou plus de mesures par ancres, l'estimation est plus précise.

Maintenant on s'intéresse à un autre estimateur au sens du maximum de vraisemblance. À partir des observations des puissances, on veut chercher un $\hat{\theta}$ qui minimise les erreurs $E = \frac{(P_0 - \bar{P})}{10(\eta - \log_{10}(\hat{\theta} - z/d_0))}$. On peut effectuer une approximation avec l'algorithme du gradient stochastique :

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma E * (\hat{\theta}_n - z) / \left| \hat{\theta}_n - z \right|^2$$

On implémente cet algorithme et on compare les résultats pour les différentes valeurs du pas γ .

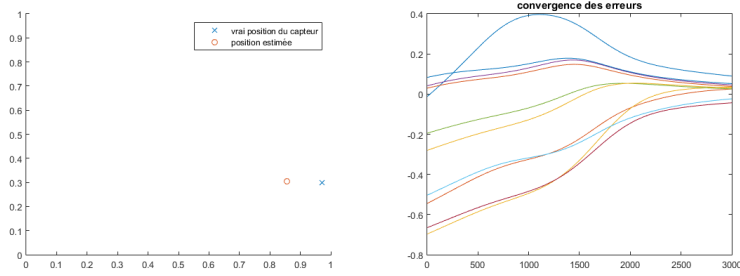


Figure 2: $\gamma = 0.002$

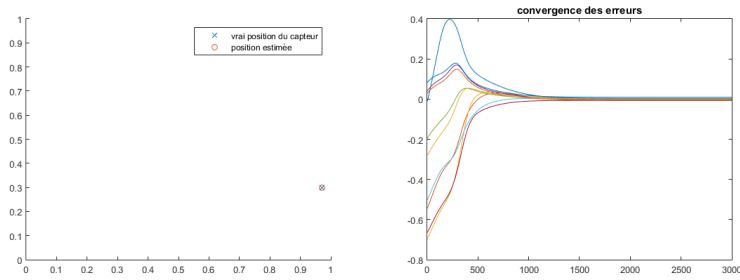


Figure 3: $\gamma = 0.01$

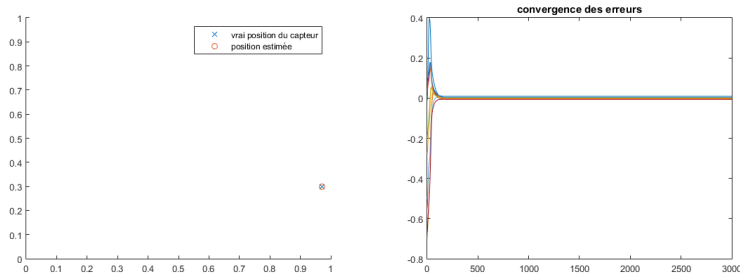


Figure 4: $\gamma = 0.1$

On peut voir quand on augmente le pas, l'algorithme converge plus vite. On arrive toujours à bien localiser le capteur si le nombre d'itération est suffisant pour que l'algorithme converge. Par rapport à la méthode de moindre carrés, on arrive tous à localiser le capteur et la performance dépend de la quantité de

données. Si on a peu d'ancres et peu de nombre de mesures par ancre, on utilisera la méthode du gradient stochastique et en revanche quand chaque ancre mesure plusieurs fois, il vaut mieux de prendre la méthode de moindre carrés parce que la complexité de la méthode du gradient est plus grande.

Maintenant on fait déplacer le capteur et on veut estimer sa position en ligne avec une méthode adaptative. On cherche à minimiser $\mathbb{E}(\|\Phi\theta - (b - \hat{\delta}_n)\|^2)$. On montre d'abord que : $\mathbb{E}[\|\Phi\theta - (b - \hat{\delta}_n)\|^2] = \|\Phi\theta - (b - \delta)\|^2$

$$\begin{aligned}\mathbb{E}[\|\Phi\theta - (b - \hat{\delta}_n)\|^2] &= \mathbb{E}[\|\Phi\theta\|^2 - 2\langle b - \hat{\delta}_n, \Phi\theta \rangle] + Cst \\ &= \|\Phi\theta\|^2 - 2\langle b - \mathbb{E}(\hat{\delta}_n), \Phi\theta \rangle + Cst\end{aligned}$$

On a $\mathbb{E}(\hat{\delta}_n) = \mathbb{E}(\delta)$

$$\begin{aligned}&= \|\Phi\theta\|^2 - 2\langle b - \delta, \Phi\theta \rangle + Cst \\ &= \|\Phi\theta - (b - \delta)\|^2\end{aligned}$$

Le problème de minimisation devient donc : $\arg \min_{\theta} \mathbb{E}[\|\Phi\theta - (b - \hat{\delta}_n)\|^2]$ On peut donc conclure l'algorithme de LMS :

$$\begin{aligned}\hat{\theta}_{n+1} &= \hat{\theta}_n - \gamma \nabla_{\theta}(\|\Phi\theta - (b - \hat{\delta})\|^2) \\ &= \hat{\theta}_n - \gamma \Phi^T(\Phi\hat{\theta}_n - (b - \hat{\delta}_{n+1}))\end{aligned}$$

En pratique, on implémente l'algorithme LMS et on modifie les paramètres pas et bruit pour voir l'influence. On fait d'abord vérifier le pas :

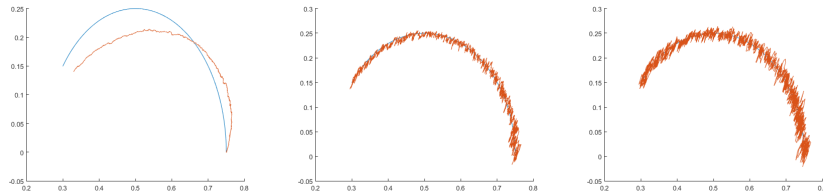


Figure 5: $\gamma = 0.005$, $\gamma = 0.05$ et $\gamma = 0.1$

On peut conclure que le choix du pas est un jeu de compromis entre la transitoire et la fluctuation. Quand γ est petit, la fluctuation est petite mais la transitoire est trop lente pour suivre le capteur. Quand on fait augmenter le γ , l'observation arrive à suivre la trajectoire mais le niveau du bruit est plus présent.

Ensuite, on fait varier le bruit σ et on voit que cela n'influe que sur la partie fluctuation.

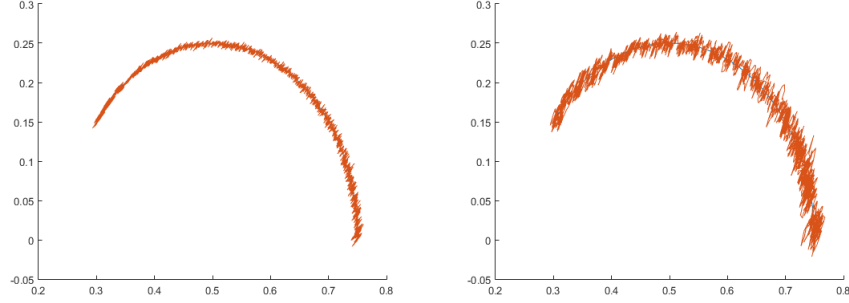
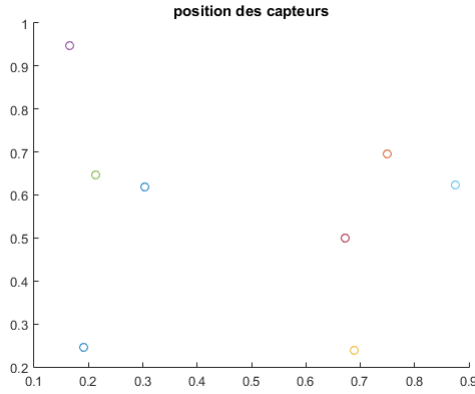


Figure 6: $\gamma = 0.1$ $\sigma = 0.1$ et $\gamma = 0.1$ $\sigma = 0.3$

2 Localisation sans ancrs

On va localiser plusieurs capteurs sans ancrs. On mesure les distances inter-capteurs. On ne peut récupérer les positions qu'à une isométrie près parce qu'on n'a aucune donnée absolue dans le plan.

Ensuite, on génère les capteurs dans le plan :

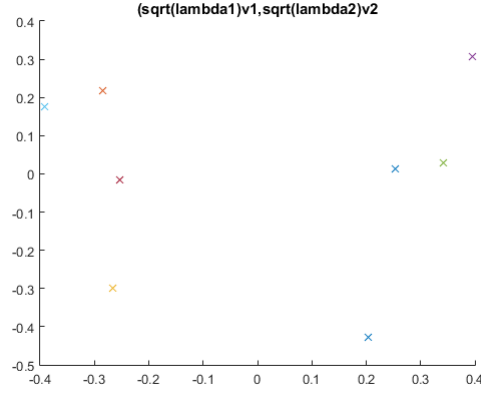


On génère la matrice qui contient toutes les distances inter-capteurs :

$$D = \begin{pmatrix} 0 & d_{1,2}^2 & d_{1,3}^2 & \cdots & d_{1,N}^2 \\ d_{2,1}^2 & 0 & d_{2,3}^2 & \cdots & d_{2,N}^2 \\ d_{3,1}^2 & d_{3,2}^2 & 0 & \cdots & d_{3,N}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N,1}^2 & \cdots & -\frac{1}{N} & \cdots & 0 \end{pmatrix}$$

et on a $P = \begin{pmatrix} 1 - \frac{1}{N} & -\frac{1}{N} & \dots & -\frac{1}{N} \\ -\frac{1}{N} & 1 - \frac{1}{N} & \ddots & \vdots \\ \vdots & \ddots & 1 - \frac{1}{N} & -\frac{1}{N} \\ -\frac{1}{N} & \dots & -\frac{1}{N} & 1 - \frac{1}{N} \end{pmatrix}$ et $M = -\frac{1}{2}PDP$

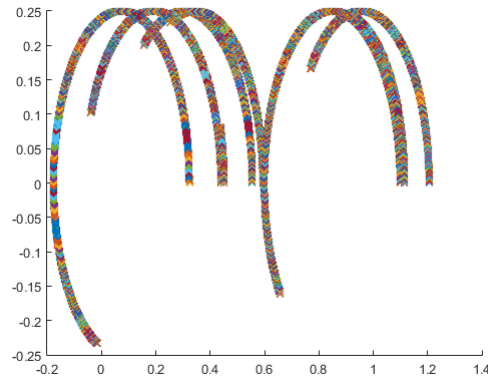
En prenant les deux valeurs propres qui sont non-nulles et les vecteurs associés de la matrice M , on trace dans le plan les coordonnées $(\sqrt{\lambda_1}\nu_1, \sqrt{\lambda_2}\nu_2)$:



On remarque que les distances inter-capteurs sont les mêmes. On pourrait faire l'estimation en retirant les valeurs et vecteurs propres de la matrice M .

On veut montrer théoriquement : $M = ZZ^T$. Le principe est de développer PDP et voir si c'est égal à $-2ZZ^T$.

On fait maintenant la localisation en ligne en appliquant l'algorithme d'Oja. On fait déplacer les capteurs comme précédemment :



Ici on ne peut pas avoir des points estimés qui suivent la trajectoire parce qu'on ne peut pas récupérer des positions absolues. Le seul moyen de vérifier

le fonctionnement de l'algorithme est de voir à tel instant, les distances inter-capteurs sont à peu près pareils. On a ci-dessous $n = 4000$ et $n = 5000$:

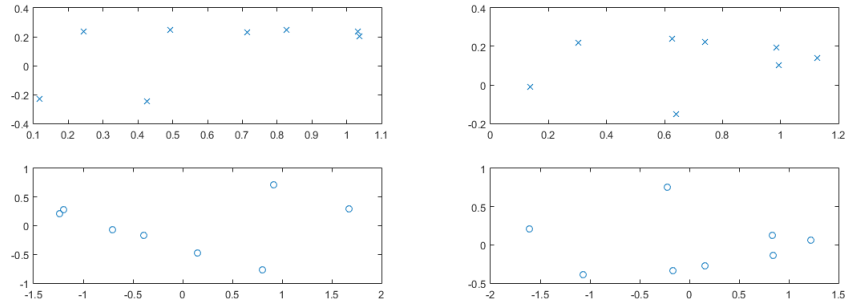


Figure 7: $n = 4000$, $n = 5000$

On remarque les vraies capteurs et ceux estimés ont la même forme. L'un peut être obtenu par l'autre avec soit la rotation soit le miroir. Cela vérifie le bon fonctionnement de l'algorithme. Comme cela est aussi un algorithme adaptatif, le choix du pas est comme l'algorithme de LMS qu'on a vu précédemment, on veut avoir un compromis entre la transitoire et la fluctuation.