



Travail Pratique 2

Travail présenté à Philippe Giguère dans le cadre du cours *GLO-7021 : Introduction à la robotique mobile*

Travail réalisé par NAOUSSI SIJOU, Wilfried Armand
NI : 536 773 538

Hiver 2022

1 Filtre de Kalman linéaire : estimation de la température d'un échantillon

1.1 a) Valeurs de Φ , Γ , Λ

Variable d'état : $X = [x]$, où x est la température de l'échantillon

Variable de commande : $u = [-1.6]$

$X(k+1) = X(k) - 1.6 \cdot dt$, dt = variation de temps

ie $\Phi = [1]$, $\Gamma = [dt]$

La fonction de la sonde est : $h_S(X) = 3X + \sigma_S^2$

ie $\Lambda = [3]$

1.2 b) Exécution du filtre

Calcul	Iteration	X_t	P_t	X_{pred}	P_{pred}	Mesure z	Gain K	X_{t+1}	P_{t+1}
Manuel	1	295	4	293.4	4.05	903	0.316	300.60	0.21
Manuel	2	300.60	0.21	299.0	0.25	885	0.1765	296.882	0.12
Programme	3	296.882	0.12	295.24	0.17	880	0.14	294.42	0.096

TABLE 1 – Tableau à compléter pour l'exercice de Kalman.

1.3 c) Convergence du gain K à l'infini

K converge vers 0.125 pour un grand nombre d'itérations

2 Solution par filtre à particules

a) $X = [d_{init} \ 0]^T$

C	4	10	40
Précision	0.1360	0.2214	0.1700

TABLE 2 – Comparaison cas a

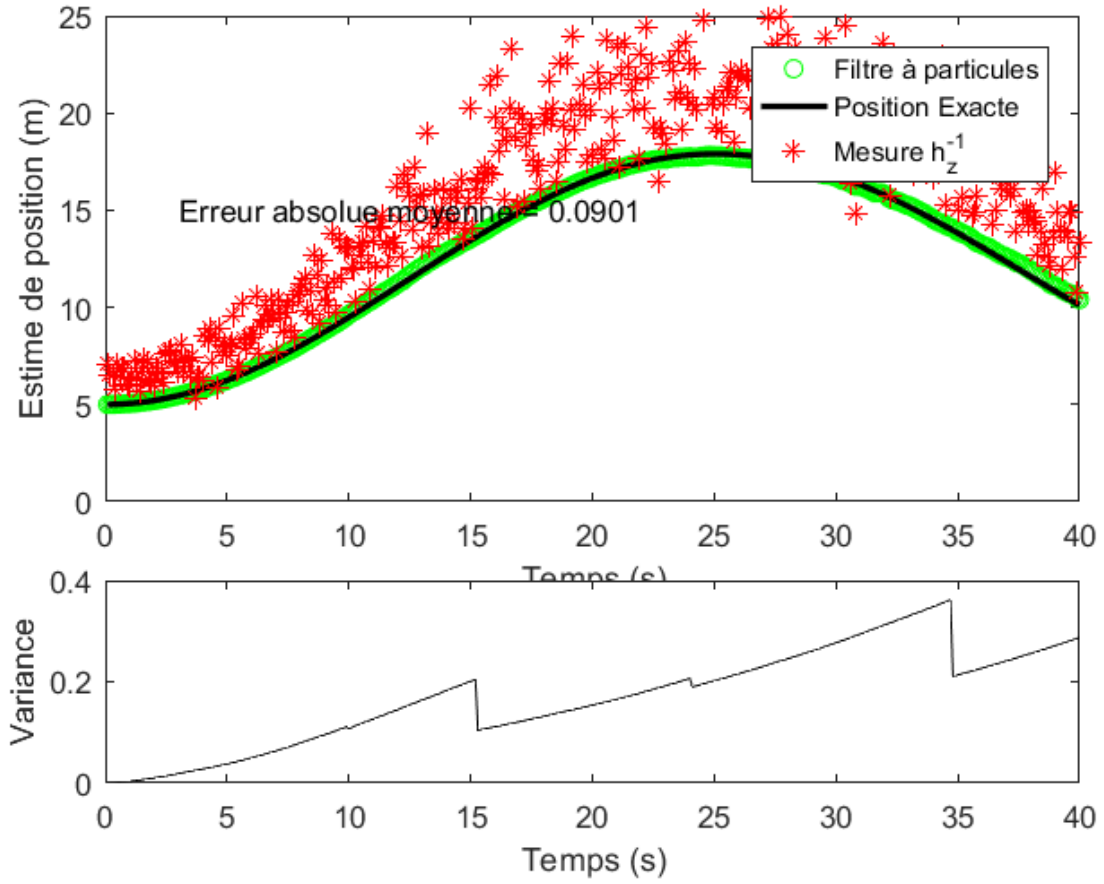


FIGURE 1 – Filtre à particules cas a, $C = 40$

b) $X \in [3, 12]$

C	4	10	40
Précision	0.0641	0.0580	0.2273

TABLE 3 – Comparaison cas b

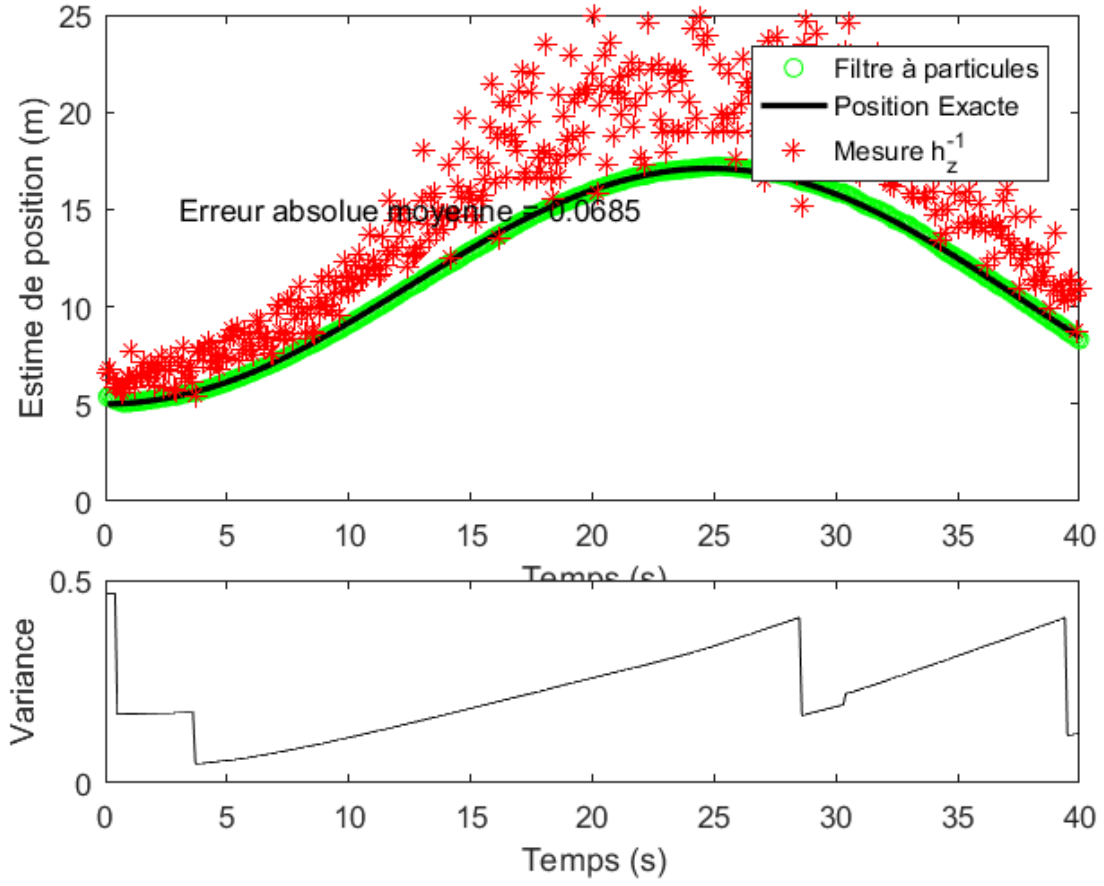


FIGURE 2 – Filtre à particules cas b, $C = 40$

3 Solution par filtre Kalman étendu (EKF)

3.1 Matrices Φ , Γ , et Λ

$$\Phi = \begin{bmatrix} 1 & dT \\ 0 & 1 \end{bmatrix}$$

$$\Gamma = [dT^2/2 \quad dT]^T$$

$$\Lambda = [-(1/8.0) * Pwifi * \exp(-X(1,1)/8.0) \quad 0]$$

3.2 Implémentation du filtre EKF

$$a) X = [d_{init} \ 0]^T, P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

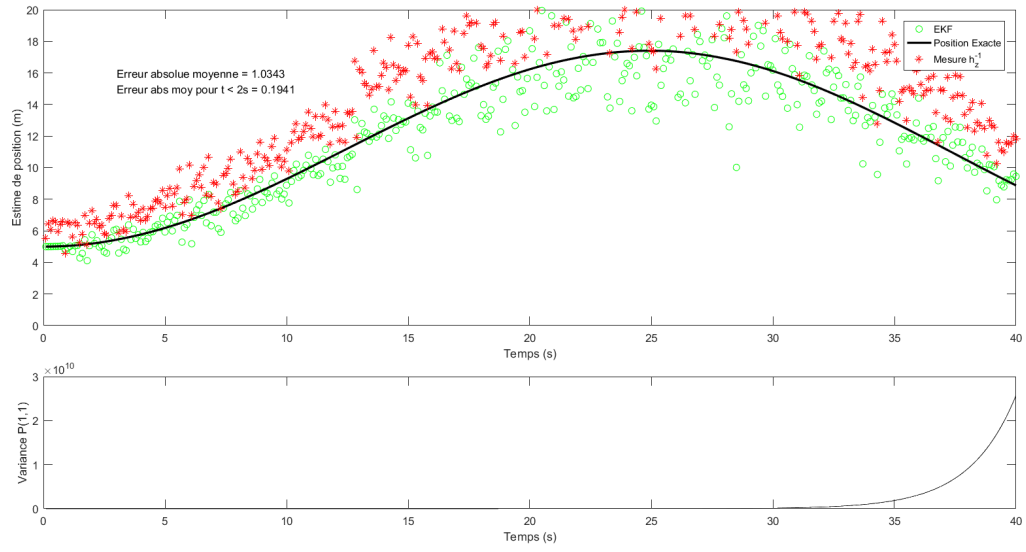


FIGURE 3 – EKF 1

b) $X = [d_{init} \ 0]^T$ et $P = \begin{bmatrix} 25 & 0 \\ 0 & 0 \end{bmatrix}$

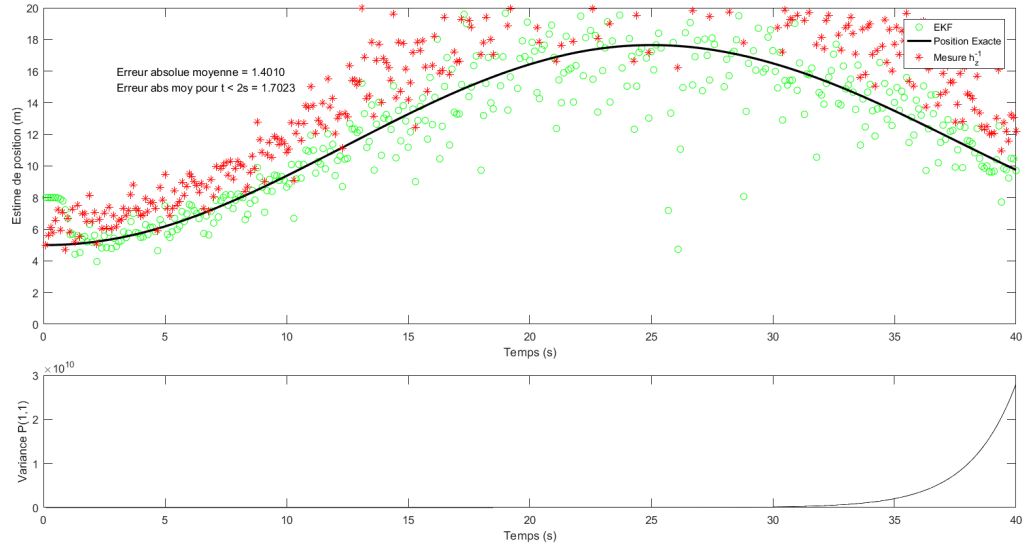


FIGURE 4 – EKF 2

c) $X = [8 \ 0]^T$ et $P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

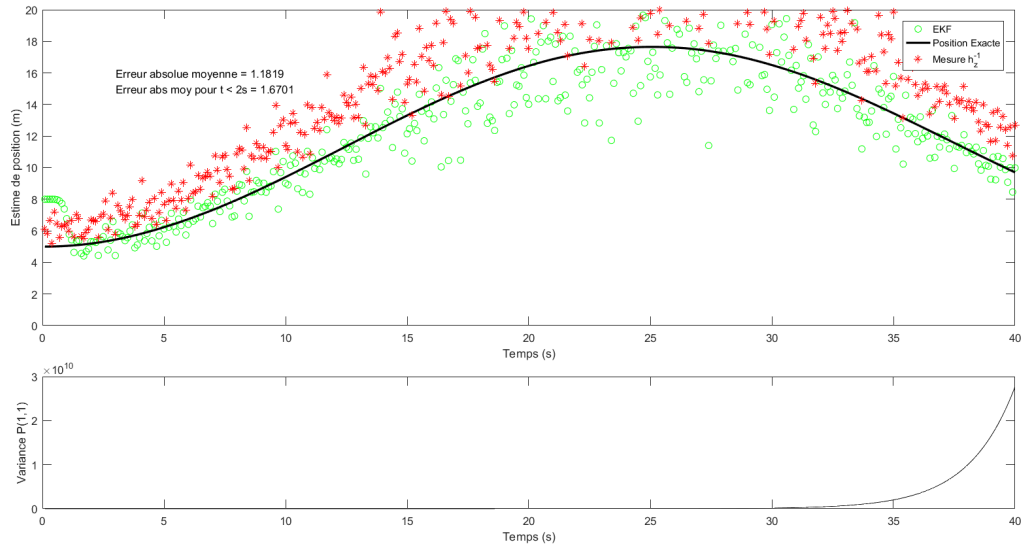


FIGURE 5 – EKF3

d) $X = [8 \ 0]^T$ et $P = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$

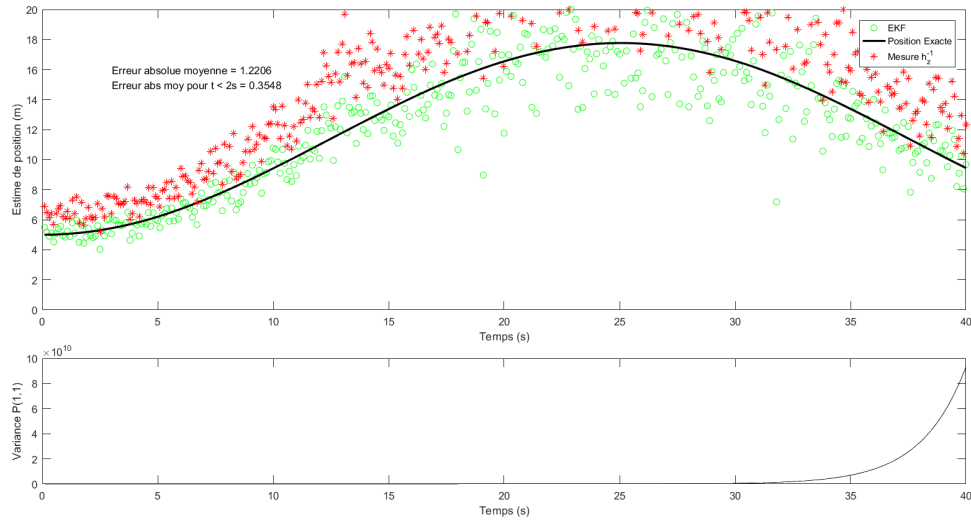


FIGURE 6 – EKF4

Dans tous les cas a, b, c et d ci-dessus, notre algorithme parvient à converger rapidement et retrouver la position réelle du chariot.

Dans la situation a, on part de la position initiale réelle avec une totale confiance sur l'estimé de notre position. Notre EKF donne donc plus de crédit à notre estimé qu'au capteur et tout va bien car on avait bon.

Dans la situation b, on part avec une valeur initiale correcte sur la position du chariot mais, on n'est pas totalement certain de notre estimé. Cela nous prend quelques itérations, mais on finit par converger et retrouver la pose réelle du chariot.

Dans la situation c, on part avec une valeur initiale biaisée sur la position du chariot et une confiance totale en notre estimé. Le biais sur la position étant faible, on parvient quand même après quelques itérations à retrouver une bonne approximation de notre position réelle.

Dans la situation d, on part avec une valeur initiale biaisée sur la position du chariot mais on fait plus confiance en notre capteur. Donc l'algorithme parvient à converger dès la première itération.

3.3 Divergence du filtre EKF

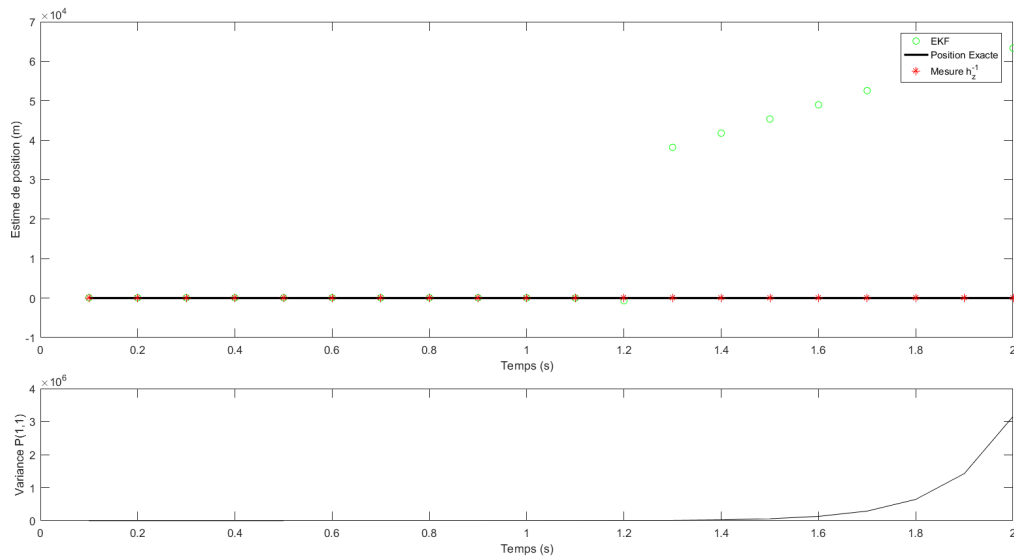


FIGURE 7 – EKF5

La distance initiale déterminée est $X = [80 \ 0]^T$ et la matrice de covariance est $P = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}$

4 Localisation globale par filtre à particules

4.1 Cas 1 : l'angle est connu

Discussion :

Etant donné qu'on ne connaît pas la position du robot il nous faut avoir assez de particules pour l'approximer correctement. Ayant une grande surface, on a déterminé que 1000 particules est le plus petit nombre nécessaire pour converger le plus de fois.

Pour S_V on décidé de surestimer sa valeur par rapport à la valeur réelle. On a donc fait des tests en incrémentant cette valeur de 0.01 à chaque fois. Et on a déterminé que celle qui donne les meilleurs résultats est 0.1.

On choisit σ_{angle} pas trop grand pour empêcher de dévier de la position réelle de notre robot si on converge et également pas trop petit pour pouvoir brasser suffisamment l'angle du robot et changer la tendance en cas de mauvais positionnement.

Pour cela, on a fait plusieurs tests en incrémentant à chaque fois la valeur de σ_{angle} de 0.01 à chaque fois on mesure l'erreur moyenne obtenue. Les résultats obtenus sont contenus dans le graphe suivant.

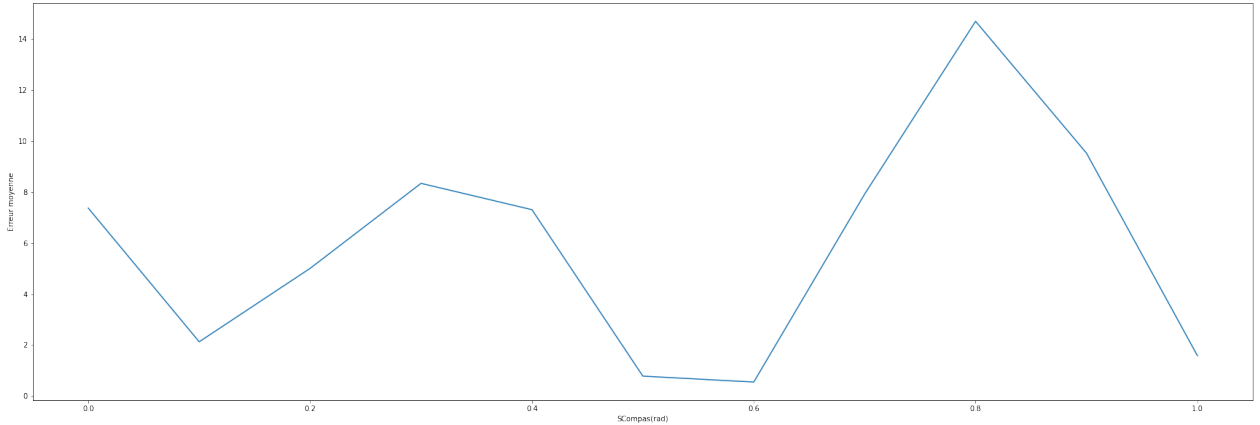


FIGURE 8 – Erreur moyenne en fonction de σ_{angle}

On peut donc déterminer que la valeur de σ_{angle} pour laquelle on obtient un score minimal est 0.6.

Les particules disposées uniformément sur la carte se concentrent rapidement autour de certains points. Et en quelques itérations il parvient à se rassembler autour d'un seul centre.

Le filtre peut échouer à cause de plusieurs raisons :

- Une mauvaise configuration dès le départ des particules qui ne couvrent pas toutes la surface du polygone ;
- De plus, plusieurs zones de la carte se ressemblent et induisent donc facilement en erreur notre algorithme.

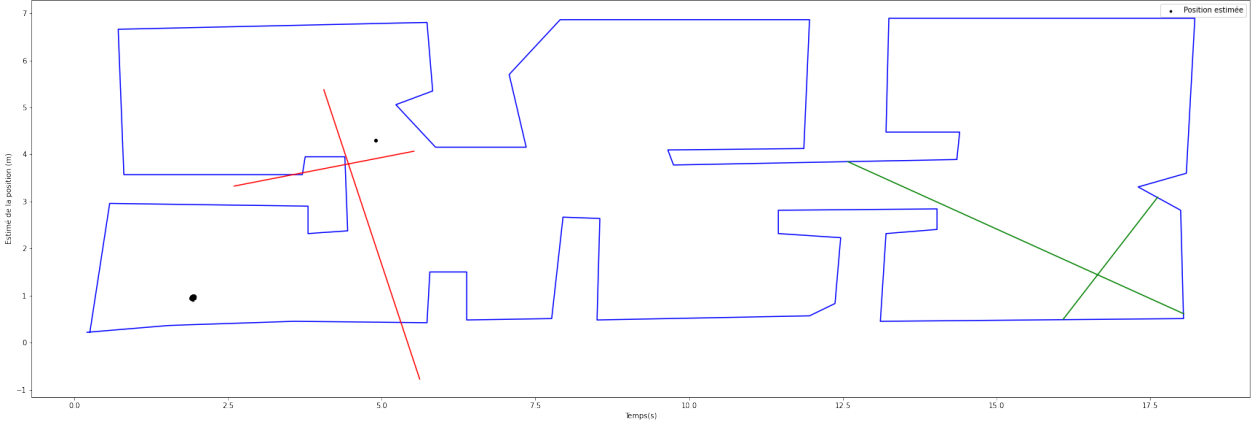


FIGURE 9 – Filtre à particules diverge



FIGURE 10 – Filtre à particules converge

4.2 Impact du bruit utilisé dans la vraisemblance LiDAR

On a fait plusieurs tests pour observer la comment se comporte l'erreur en fonction du choix de σ_{LiDAR} . On a constaté que plus σ_{LiDAR} est petit, plus il faut de particules pour pouvoir converger. A contrario, plus σ_{LiDAR} est grand et moins vite il converge.

Le graphe ci-dessous montre l'erreur moyenne en fonction du choix de σ_{LiDAR} . On peut remarquer grâce à celui-ci que plus σ_{LiDAR} est grand plus l'erreur est grande. Cela montre qu'on tolère plus d'estimations grossières. On pourrait cependant réussir à obtenir de bon résultat par chance lorsque les particules sont bien placées dès le départ. C'est ce qui explique le pic inversé pour $\sigma_{LiDAR} = 0.7$

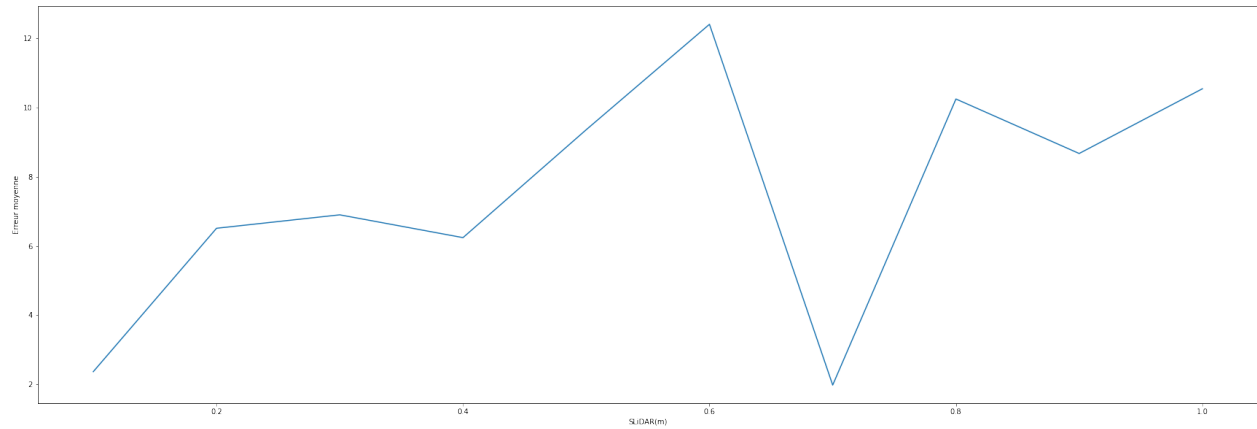


FIGURE 11 – Erreur moyenne en fonction de σ_{LiDAR}

Il serait donc intéressant d'augmenter la valeur de σ_{LiDAR} au fil du temps pour éviter à notre algorithme de sauter aux conclusions rapidement lorsqu'on tombe sur des coins similaires à notre position réelle.