

ESTIMATION OPTIMALE DE TRAJECTOIRES D'ANIMAUX MARINS

Julie Van Der Ham,
Maître de Stage: Philippe Gaspar

Septembre 2006-Mars 2007



Table des matières

1	Introduction	2
2	Description de CLS et du travail à réaliser	3
2.1	CLS	3
2.2	Le sujet du stage	3
3	Le problème de l'estimation de trajectoires	5
3.1	La modélisation du mouvement	5
3.2	Le traitement des mesures	6
4	Traitements et filtrage des données	8
4.1	Traitements préalables	8
4.2	Le filtre de Kalman	10
4.3	Le choix des paramètres d'erreur	11
5	Tunage du filtre	12
5.1	Les simulations	12
5.2	Principe des tests de χ^2	13
5.2.1	Test de moyenne et covariance sur l'erreur d'estimation (a)	14
5.2.2	Test de moyenne et variance sur l'innovation normalisée (b)	15
5.2.3	Test de blancheur sur l'innovation normalisée (c) . . .	16
5.3	Apprentissage du filtre	17
6	Resultats obtenus et commentaires	19
6.1	Résultats observés pour quelques trajectoires	19
6.2	Comparaison avec d'autres méthodes d'estimation	21
7	Annexes	23
7.1	Principe de fonctionnement des balises Argos	23
7.2	Description des principales routines écrites sous Matlab	25
8	Conclusions	26

1 Introduction

Ce stage concernant l'estimation de trajectoire a été réalisé au sein de la DOS de CLS pendant 6 mois. Il concernait la mise en oeuvre d'un filtre de Kalman. Les méthodes déjà en place ont été étudiées, théoriquement puis par observation des résultats, avant de se pencher sur la réalisation de la méthode demandée. La décision d'un filtre classique, sans ajout des actions de courant a été prise. Ce rapport présentera brièvement l'entreprise CLS et le sujet du stage, puis le problème général d'estimation d'etrajectoire. La théorie sur le filtre de Kalman et les tests qui ont été réalisés seront traités ensuite, avant d'aborder finalement les resultats observés.

Je remercie Philippe Gaspar qui m'a permis de réaliser un stage intéressant et instructif ainsi que Patrick Danes, pour l'aide et les corrections apporté à mon travail.

2 Description de CLS et du travail à réaliser

2.1 CLS

CLS est une filiale du CNES (Centre National d'Etudes Spatiales), de l'IFREMER (Institut Français de Recherche pour l'Exploitation de la MER) et de banques françaises, créée en 1986. Le siège se situe à Ramonville (Toulouse) et plusieurs filiales se trouvent à travers le monde. Les principales activités de CLS sont :

- La collecte de données satellites et l'exploitation opérationnelle des systèmes fournissant ces données.
- La commercialisation des produits et services provenant de ces systèmes.
- L'étude et le développement.

Ainsi, CLS exerce ses activités dans différents domaines liés à l'environnement :

- L'océanographie, avec l'étude des changements climatiques, la prévention de crises
- La pêche, en surveillant le respect des règles et les efforts des bateaux
- La sécurité maritime, avec la surveillance de flottes (comme par exemple lors de courses)
- La protection de la faune (avec le balisage Argos d'animaux en danger)

Ce stage a été réalisé au sein de la Direction Océanographie Spatiale, ou DOS, dont les missions principales sont :

- le maintien et développement des compétences en océanographie spatiale et opérationnelle
- le développement et la mise sur le marché des applications commerciales

2.2 Le sujet du stage

Classée espèce en danger depuis de nombreuses années, la Tortue Luth est pourtant encore souvent tuée accidentellement dans les filets de pêche. Le recoupement des zones de pêche actuellement connues avec les routes migratoires des tortues luths pourrait permettre à terme de participer à la conservation de cette espèce. En effet, la mise en place d'une réglementation précise des pêches dans des zones sensibles pourrait éviter les comportements les plus dangereux pour cette espèce.

Des tortues sont ainsi équipées de balises Argos, dans le but de simuler leur déplacement à partir des mesures récoltées et des modèles de dynamique développés.

Cependant, les techniques de suivi individuel nous livrent des séries temporelles de positions le plus souvent entachées d'erreurs importantes. Les animaux localisés par satellite (ARGOS) n'emportent que des balises de faible puissance qui ne permettent qu'une localisation de faible qualité avec des erreurs atteignant souvent plusieurs kilomètres.

L'objectif du stage proposé est de développer des techniques de reconstruction de trajectoires s'appuyant sur ses positions estimées combinées avec des modèles de déplacement simple.

3 Le problème de l'estimation de trajectoires

- Il y a une double incertitude lors de la reconstruction de trajectoire :
- Il faut établir une équation de déplacement de l'animal, que l'on ne connaît pas dans l'absolu, et qui sera notre connaissance à priori.
 - Les données reçues par satellites sont elles même erronées et il faudra en tenir compte.

Les seules connaissances à priori que nous avons sont la distribution de l'état initial, ainsi que les modèles de dynamique et de mesure.

3.1 La modélisation du mouvement

L'hypothèse qui sera émise est que la tortue se déplace à vitesse constante, vitesse que l'on ne connaît pas et qui doit donc apparaître dans l'état à estimer. On considère ainsi l'état X comme étant de dimension 4 :

$$X = \begin{pmatrix} x \\ y \\ u \\ v \end{pmatrix}$$

avec :

- x : abscisse de la position.
- y : ordonnées de la position.
- u : vitesse selon x .
- v : vitesse selon y .

Ne supposant pas pour le moment qu'il y ait de bruit lié à l'erreur de ce modèle, l'équation s'écrit formellement, à temps continu :

$$\dot{X} = A.X$$

avec la matrice A qui diffère selon le type de déplacement. Dans notre modèle, en supposant la vitesse constante, nous considérerons la matrice A égale à :

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Considérons maintenant le bruit de dynamique du modèle, il est aussi de dimension 4 :

$$W = \begin{pmatrix} w_y \\ w_u \\ w_v \end{pmatrix}$$

- w_x et w_y traduisent le fait que les vitesses u et v ne s'intègrent pas parfaitement aux positions.
- w_x et w_y existent car les vitesses ne sont pas exactement constantes.

C'est un bruit supposé centré et blanc qui suit une loi normale de moyenne 0 et de matrice de covariance

$$Q_w = \begin{pmatrix} \text{var}(w_x) & 0 & 0 & 0 \\ 0 & \text{var}(w_y) & 0 & 0 \\ 0 & 0 & \text{var}(w_u) & 0 \\ 0 & 0 & 0 & \text{var}(w_v) \end{pmatrix}$$

L'équation de mouvement s'écrit alors :

$$\dot{X} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} X + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} W$$

Il faut ensuite transformer cette équation en un modèle à temps discret entre deux instants successifs séparés de δT . Par intégration puis discrétisation approchée, on obtient :

$$X_{k+1} = AX_k + G_k W_k$$

$$\text{avec } A_k = \begin{pmatrix} 1 & 0 & \delta T & 0 \\ 0 & 1 & 0 & \delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{et } G_k = \begin{pmatrix} \delta T & 0 & (\delta T)^2/2 & 0 \\ 0 & \delta T & 0 & (\delta T)^2/2 \\ 0 & 0 & \delta T & 0 \\ 0 & 0 & 0 & \delta T \end{pmatrix}$$

Cette équation sera celle utilisée pour le filtrage.

3.2 Le traitement des mesures

Les balises Argos émettent des messages qui sont reçus par des satellites en orbite terrestre basse, qui transfèrent ensuite les messages vers des stations terrestres. Celles-ci transmettent automatiquement les messages vers les centres de traitement Argos, qui déterminent la position des émetteurs, et traitent les données fournies par les capteurs. Plus de détails sont disponibles dans l'annexe "principe de fonctionnement des balises Argos".

L'évaluation de la précision de la localisation est basée sur :

- Les conditions géométriques du passage satellite traité,
- Le nombre de messages reçus pendant le passage,
- La stabilité en fréquence de l'émetteur.

A chaque localisation correspond donc une classe de précision. On en distingue 6, dont la précision correspondante varie entre moins de 150m et plus de 1000m. Il faudra tenir compte de ces classes dans l'estimation de l'erreur de mesure.

Une équation de mesure s'écrit de manière générale : $Z_k = h(X_k) + V_k$, où h est donc fonction de X_k . Cette équation est linéaire pour une fonction h est indépendante de X . En choisissant de travailler en coordonnées métriques, nous aurons positions et estimation d'erreur dans les même unités, et l'équation de mesure, en discret ou en continu, s'écrit :

$$Z_k = HX_k + V_k$$

avec

- Z_k la mesure à l'instant t_k .
- V_k le bruit de mesure.
- H la matrice d'observation.

avec la matrice H qui est :

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Le modèle est parfaitement linéaire, le filtre de Kalman linéaire peut donc s'appliquer.

4 Traitements et filtrage des données

4.1 Traitements préalables

Les données Argos à traiter sont sous forme de fichiers Diag. Ceux ci se présentent de la manière suivante :

Les données fournies sont :

- Le numéro de balise
- La date
- La classe de la loc LC
- Les coordonnées de la position (longitude 1, latitude 1)
- Les coordonnées de la position image (longitude 2, latitude 2)

Les autres informations concernent le nombres de message recus pour cette position par le satellite, la fréquence de ces messages. Ces informations ne sont pas utilisées ici. Voici un extrait de fichier Diag pour deux données reçues :

```
04261 Date : 06.12.02 07:19:45 LC : Z IQ : 00
      Lat1 : 3.7026S Lon1 : 10.9506E Lat2 : 3.6186N Lon2 : 21.551W
      Nb mes : 004 Nb mes>-120dB : 000 Best level : -122 dB
      Pass duration : 120s NOPC : 0
      Calcul freq : 401 652455.0 Hz Altitude : 0 m
                   00          92          36

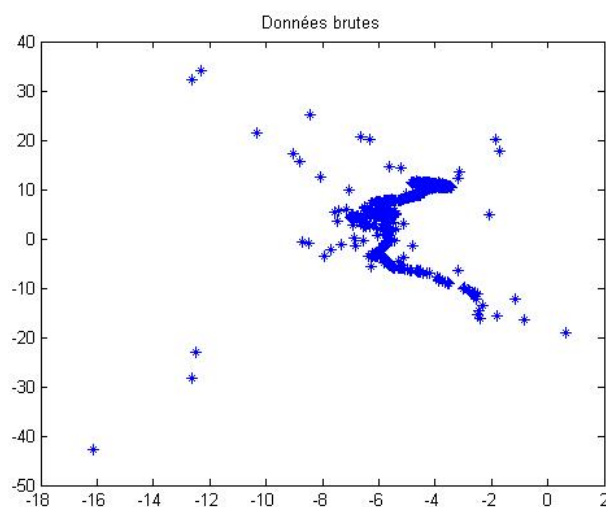
04261 Date : 06.12.02 12:37:07 LC : Z IQ : 00
      Lat1 : 3.7019S Lon1 : 10.9561E Lat2 : 0.6113N Lon2 : 29.5955E
      Nb mes : 004 Nb mes>-120dB : 000 Best level : -122 dB
      Pass duration : 120s NOPC : 0
      Calcul freq : 401 652455.0 Hz Altitude : 0 m
                   00          92 |          36
```

Il faut dans un premier temps récupérer ces données, et comme nous travaillons sous Matlab, nous allons stocker ce qui nous est utile sous forme de matrice.

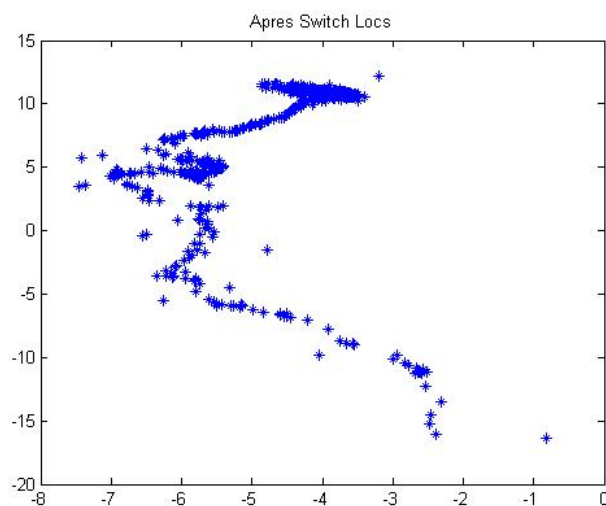
Il existe des données inutilisables, car la classe de localisation, ou bien les coordonnées elles même n'ont pas de valeurs, et se trouvent quand même dans le fichier de données. Un algorithme se charge de les éliminer.

Le principe de calcul par les centres de traitements implique que deux positions sont fournies pour chaque mesure, comme il est expliqué dans l'annexe. Dans les deux positions retenues, une est censée être la vraie position, alors que la deuxième est la localisation image, qui se situe souvent loin de la première. Parfois, une erreur survient, et la localisation image est fournie

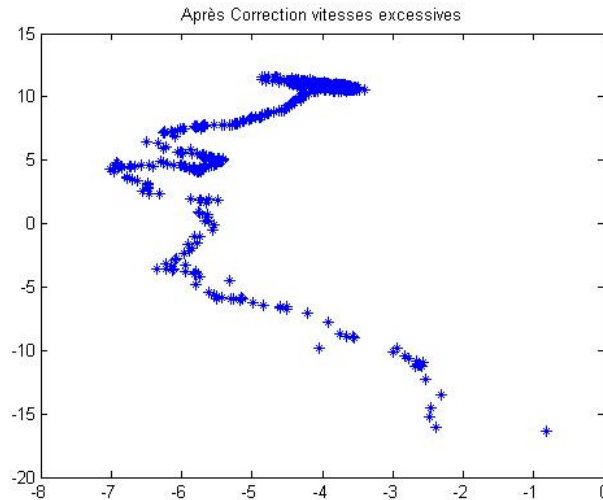
comme étant la position correcte. Avant tout traitement par filtrage sera appliqué un algorithme d'échange des loc/loc images dans les cas erronés. Après les avoir repérées, il échange les deux localisations pour que lors du filtrage, la première localisation correcte soit systématiquement choisie. L'image ci-dessous montre l'exemple d'un ensemble de données, avant et après échange des localisations :



Après échange, certaines localisations ont été replacées dans la trajectoire :



Il subsiste enfin, comme on peut le voir sur la figure, des données non réalistes quant à la vitesse de la tortue. Elles proviennent de la mauvaise qualité de certaines mesures. Un algorithme élimine les positions qui impliquent une vitesse supérieure à un certain seuil, car il est très peu probable que la tortue ait pu se déplacer à cette vitesse ! Voici la même trajectoire après suppression de ces positions :



Les algorithmes de pré-traitements ont été écrits par Beatriz Calmettes.

- readDiag : récupère les données et les stocke dans une matrice
- cleanData : enlève les lignes contenant des valeurs non valides pour la localisation dans la matrice de données brutes
- correction-mat-choix-loc : corrige les localisations images
- elimination-mat-vitesse-excessive : élimine les positions impliquant une vitesse trop importante

Les données sont désormais prêtes à être filtrées.

4.2 Le filtre de Kalman

Le principe du filtre de Kalman est une estimation bayésienne de la loi a posteriori de l'état conditionné aux mesures disponibles. Il détermine les moments (d'ordre un et deux) des états de la loi à priori et à posteriori. Le calcul se réalise comme suit, en deux étapes pour chaque pas temporel :
Etape de prédiction :

$$\hat{x}_k^- = A\hat{x}_{k-1}$$

$$P_k^- = AP_{k-1}A' + Q_k$$

Etape de correction :

$$\text{Matrice de gain : } K_k = P_k^- H' (H P_k^- H' + R_k)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

avec :

- \hat{x}_k^- état prédit $E[X_k | Z_k = z_{1:k-1}]$
- \hat{x}_k état corrigé $E[X_k | Z_k = z_{1:k}]$
- z_k la mesure de l'instant k
- P_k^- erreur de covariance estimée à priori
- P_k erreur de covariance estimée a posteriori
- Q_k matrice de covariance du bruit de modelisation
- R_k matrice de covariance du bruit de mesure à l'instant k

La phase de prédiction utilise l'état estimé à l'instant précédent pour produire une estimation de l'état courant. Dans l'étape de mise à jour, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

La matrice Q est la matrice de covariance des produits $G_k W_k$ de l'équation de mouvement de référence, ce n'est pas Q_w , matrice de covariance des W_k . Elle vaut donc : $Q = G_k Q_w G_k'$

Pour initialiser le filtre, il faut une connaissance à priori. Une manière d'opérer est de déterminer une zone de départ possible pour la tortue et une vitesse initiale, puis de fusionner cette connaissance avec la première mesure.

4.3 Le choix des paramètres d'erreur

Les paramètres qu'il nous faut fixer pour le filtre sont :

- La matrice de covariance d'erreur de dynamique Q .
- Les matrices de bruits de mesures R_k qui diffèrent selon la classe de la localisation k .

Pour les R_k , nous avons à notre disposition les indications d'erreur fournies par Argos, qui définiront donc les valeurs de variance d'erreur. Pour la matrice Q_w , nous avons évalué dans un premier temps les valeurs possibles de W_k défini plus haut. Nous avons retenu les valeurs d'erreur de $5.10^{-3} m.s^{-1}$ et $1.10^{-9} m.s^{-2}$. Ces valeurs n'étant qu'un premier essai, que nous modifierons plus tard avec les tests réalisés.

5 Tunage du filtre

Le principe du tunage est de rendre le filtre consistant. Il s'agit de vérifier si les erreurs fournies par le filtre suivent leur propriétés statistiques théoriques. Il y a l'erreur d'estimation d'une part, définie par $e_i = X_k - \hat{X}_k$ (où X_k est le vrai état qu'on cherche à estimer, et l'innovation $\eta_i = Z_i - HX_i^-$). La consistance d'un filtre est cruciale pour une estimation optimale.

Il y a trois points à tester pour vérifier la consistance :

- (a) L'erreur d'estimation doit avoir une moyenne nulle et une variance proche de celle calculée par le filtre.
- (b) L'innovation doit vérifier les mêmes critères
- (c) La séquence d'innovation doit être blanche

Le premier critère se vérifie seulement sur des simulations, car nécessite de connaître les vrais états. Avec de vraies données, on ne peut effectuer que les tests sur les deux derniers critères.

5.1 Les simulations

Le principe est de générer des états et les mesures qui auraient pu en découler puis d'examiner les états restitués par le filtre qui ne connaît que les mesures. Lors de simulations, il est important de noter qu'il faut utiliser :

- Le même modèle de déplacement pour simulation et filtrage
- Les mêmes paramètres de variance d'erreur lorsqu'on génère les bruit pour créer le jeu de données en simulation, et lorsqu'on filtre.

Pour simuler une trajectoire, il faut générer au préalable tous les bruits, de mesures et de modélisation, afin de générer les états et les mesures.

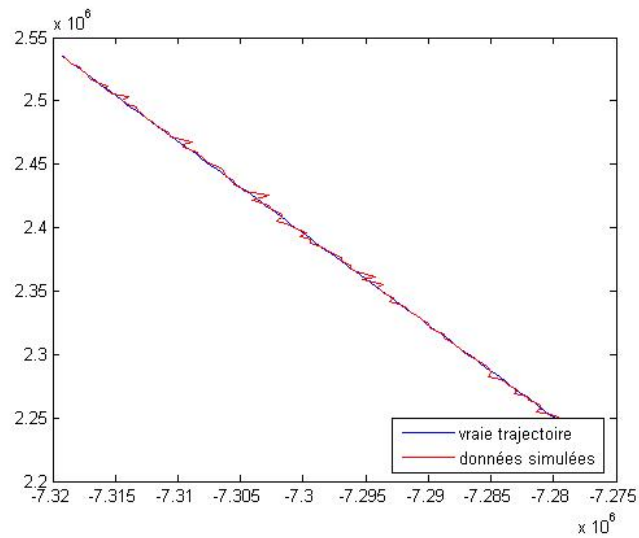
On génère les bruits W_k qui sont les bruits d'erreur du modèle, en utilisant la matrice Q_w du filtre, afin de générer les positions théoriques de la tortues :

$$X_k = AX_{k-1} + G_k W_k$$

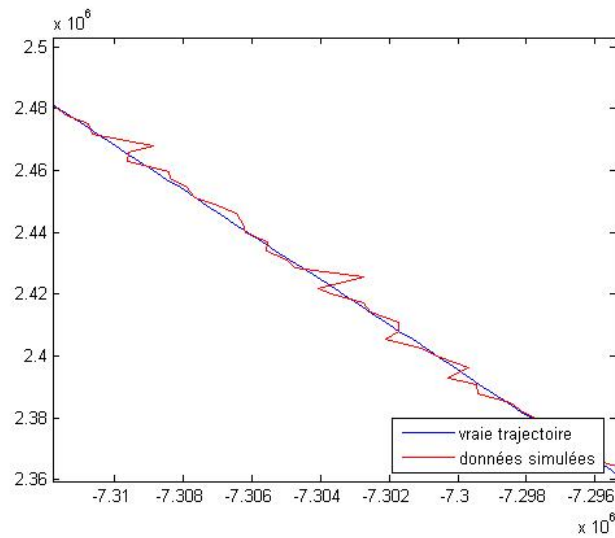
On génère aussi les bruits V_k qui sont les bruits de mesures, en utilisant les valeurs d'erreur de chaque localisation (la classe de localisation est tirée au hasard), que l'on utilise pour simuler des mesures :

$$Z_k = HX_k + V_k$$

On génère ainsi une trajectoire en possédant les positions censées être les bonnes, et les mesures qu'on aurait récoltées. L'exemple suivant est obtenu pour 100 pas de temps, soit 100 heures (on suppose une donnée recue toute les heures).



En regardant de plus près, on distingue la vraie trajectoire en bleue, qui est rectiligne, et la trajectoire simulée, en rouge qui suit la précédente, avec les erreurs que l'on a indiquées pour la simulation.



5.2 Principe des tests de χ^2

Lorsqu'une n variables x_k suivent une loi Gaussienne centrée et réduite, alors la somme $\sum_{k=1}^n x_k$ suit une loi du χ^2 à n degré de liberté. Les tests

de χ^2 vont permettre de vérifier la consistance, avec des sommes d'erreur d'estimation lorsque les états réels sont connus (ce qui est le cas lorsqu'on simule une trajectoire, on connaît alors états et mesures) ou bien d'innovation (lorsqu'on ne connaît que les mesures dans le cas de mesures seules).

Il s'agit ici de vérifier si les erreurs d'estimation et les innovations se distribuent bien selon une loi de χ^2 . On tracera pour chaque test les limites à 95% de la loi, et on observera si ces quantités se situent entre les deux bornes pour au moins 95% des points.

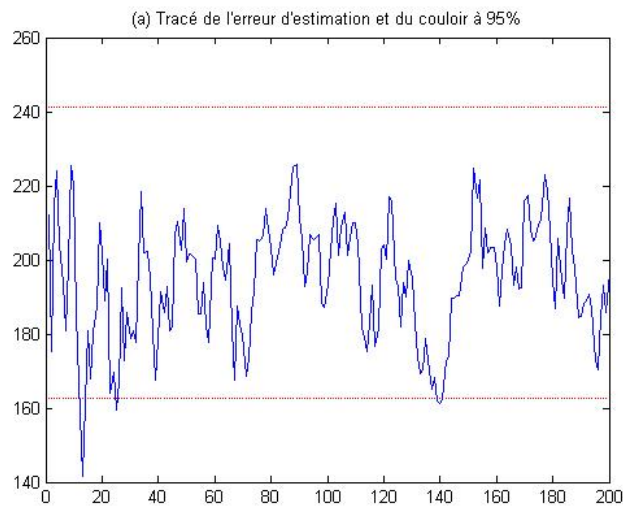
5.2.1 Test de moyenne et covariance sur l'erreur d'estimation (a)

Ce test est le plus important. après plusieurs simulations d'une même trajectoire (rapelons que celles ci diffèrent du fait de la génération aléatoire des bruits), on possède pour chaque point de la trajectoire plusieurs valeurs générées. C'est le principe des simulations de Monte Carlo. Nous notons $e(k)$ l'erreur d'estimation normalisée, qui vaut :

$$e(k) = [(X_k - \hat{X}_k)'(P_k)^{-1}(X_k - \hat{X}_k)] \text{ pour le point } k$$

Sur N simulations, on possède donc une série de $e_i(k)$ pour chacune des positions k, i variant de 1 à N. Le principe de ce premier test est de vérifier que pour chacun des point k, la moyenne de sa série : $\bar{e}_k = \frac{1}{N} \sum_{i=1}^N e_i(k)$ est telle que $N\bar{e}_k$ suit une loi de χ^2 de degré de liberté Nn_x ou n_x est la dimension du vecteur X. Le test est validé si la valeur de \bar{e}_k se situe dans les 95% de la surface de représentation de la loi de χ^2 .

Voici ce que nous avons obtenu avec une simulation de 100 trajectoires de 200 points chacune :



On reste bien dans la zone à 95% de confiance pour une tres grande proportion de points.

5.2.2 Test de moyenne et variance sur l'innovation normalisée (b)

A la différence du premier test, celui ci peut être effectué sur de vraies données car il nécessite seulement de connaitre les mesures que l'on a fournies au filtre. L'innovation normalisée, pour un point donnée est définie par :

$$e_{\eta}(k) = \eta'_k S_k^{-1} \eta_k$$

où η_k est l'innovation et S_k sa covariance. Cette innovation normalisée doit suivre une loi du χ^2 , de degré de liberté n_z , dimension des mesures. Dans le cas où l'on réalise ce test sur un jeu de plusieurs simulations (N), on calcule la moyenne des innovations normalisées, pour chaque point k :

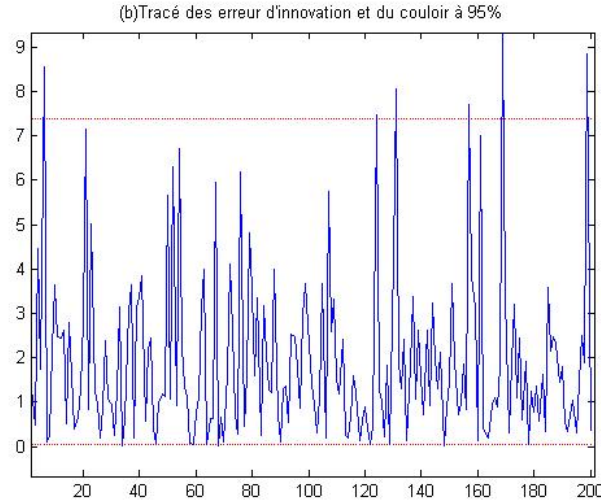
$$\bar{e}_{\eta}(k) = \frac{1}{N} \sum_{i=1}^N e_{\eta i}(k)$$

$N\bar{e}_{\eta}(k)$ doit suivre une loi de χ^2 de degré de liberté Nn_z . Dans le cas d'un seul jeu de K données réelles, on met l'hypothèse d'ergodicité, et sur un seul jeu de donnée, qui devrait être aussi représentatif qu'un ensemble de N simulation, on vérifie que :

$$\bar{e}_{\eta} = \frac{1}{K} \sum_{i=1}^K e_{\eta}(i)$$

suit une loi de χ^2 de degré de liberté Kn_z .

Voici un exemple de ce que l'on obtient, pour une trajectoire de 200 points, dans le cas d'une seule simulation.



Les points restent bien dans la zone à 95% ici encore.

5.2.3 Test de blancheur sur l'innovation normalisée (c)

On utilise ici la statistique d'une série d'autocorrélation :

$$\bar{\rho}(k, j) = \sum_{i=1}^N \eta_i(k)' \eta_i(j) [\sum_{i=1}^N \eta_i(k)' \eta_i(k) \sum_{i=1}^N \eta_i(j)' \eta_i(j)]^{-1/2}$$

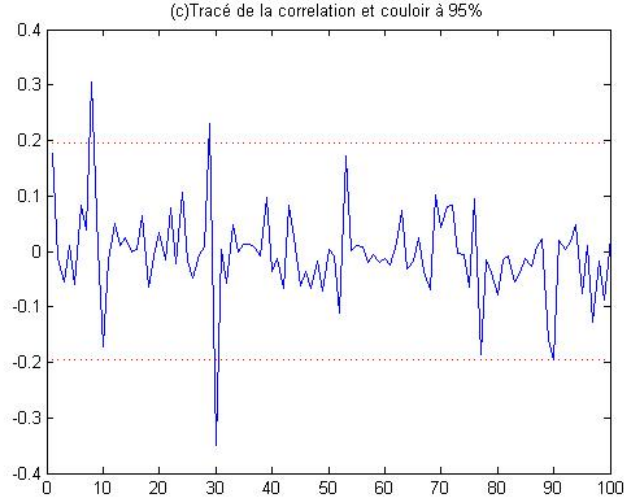
Pour un N assez grand, une approximation normale pour $\bar{\rho}(k, j)$ est acceptable. Si les innovations sont centrées et blanches, alors les paramètres de la loi normale en question sont moyenne nulle et variance égale à $\frac{1}{N}$. Il faut donc vérifier que $\bar{\rho}(k, j)$ appartient à $[-\frac{1.96}{\sqrt{N}}; \frac{1.96}{\sqrt{N}}]$.

Dans le cas d'un seul jeu de K données réelles, on regarde si :

$$\bar{\rho}(l) = \sum_{k=1}^K \eta(k)' \eta(k+l) [\sum_{k=1}^K \eta(k)' \eta(k) \sum_{k=1}^K \eta(k+l)' \eta(k+l)]^{-1/2}$$

appartient à $[-\frac{1.96}{\sqrt{K}}; \frac{1.96}{\sqrt{K}}]$. En général, on prend la moitié des points de la trajectoire et on regarde si ils sont corrélés avec les autres points. Ainsi, le résultat du test s'observe sur $\frac{K}{2}$ points, dans le cas d'une trajectoire de 200 points, on vérifie si moins de 5 points sur les 100 représentés se situent dans le couloir.

Voici un exemple, pour des trajectoires de 200 points, sur une simulation.

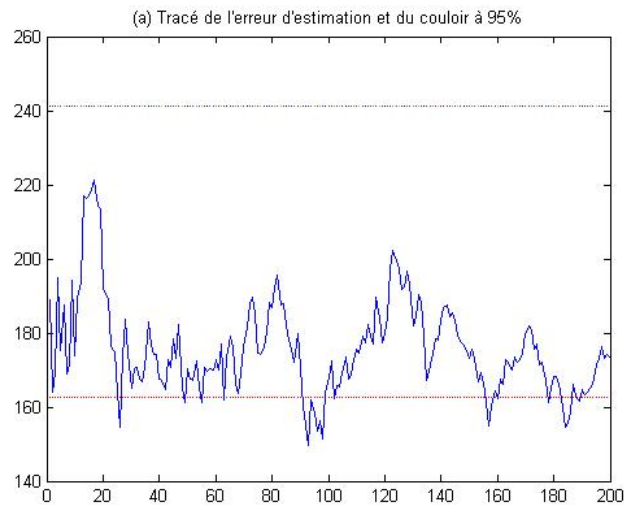


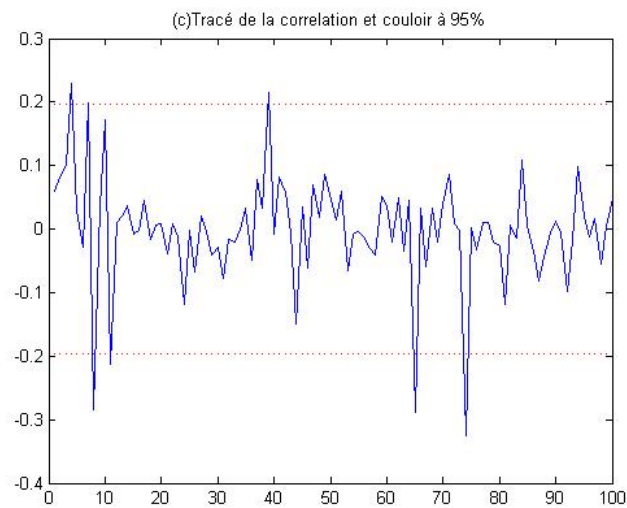
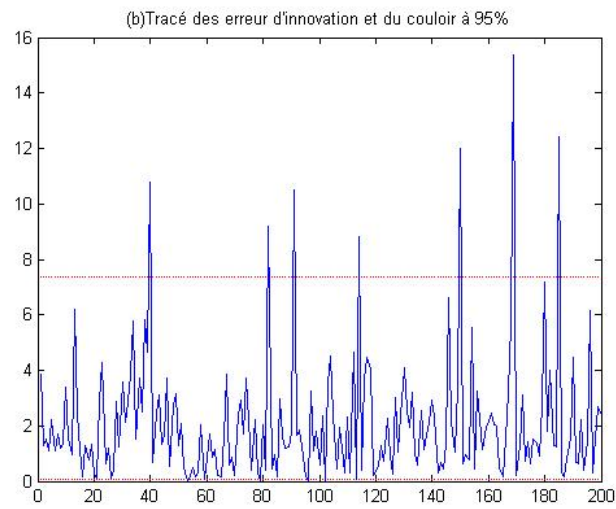
Le résultat est aussi satisfaisant.

5.3 Apprentissage du filtre

Les résultats trouvés ci-dessus, ont prouvé un bon fonctionnement du filtre, lorsqu'on connaît les bonnes hypothèse de dynamique, avec les simulations. Ce qui se produit en pratique est tout autre, puisqu'on ne connaît pas les bruits de dynamiques. Il va donc falloir apprendre à déterminer sans les connaître, quelles sont les erreurs de dynamiques qui permettent de restituer au mieux la trajectoire, c'est à dire en donnant de bons résultats aux tests de χ^2 . Les tests ont été refaits d'abord sur des simulations, en prenant pour les valeurs de Q des valeurs très élevées, soit un filtre très pessimiste puis en essayant par les tests de χ^2 de se rapprocher des valeurs réelles de Q , il s'agit alors de recueillir un peu d'expertise.

Au départ, les résultats des tests montrent clairement que l'on est trop pessimistes, puisqu'on dépasse souvent la borne inférieure.





La deuxième étape est d'essayer de refaire la même chose qu'avec les simulations sur des données réelles. Enfin, il s'agira d'essayer à l'épreuve le filtre ainsi expertisé sur des trajectoires comportant par exemple des positions très étonnantes, pour étudier son comportement et sa capacité à deviner les bons bruits de dynamique.

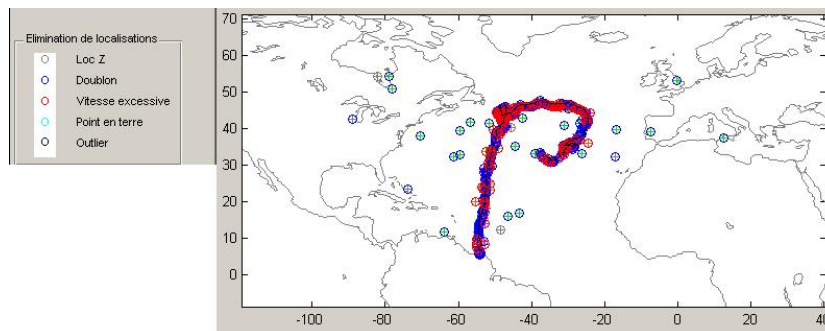
6 Résultats obtenus et commentaires

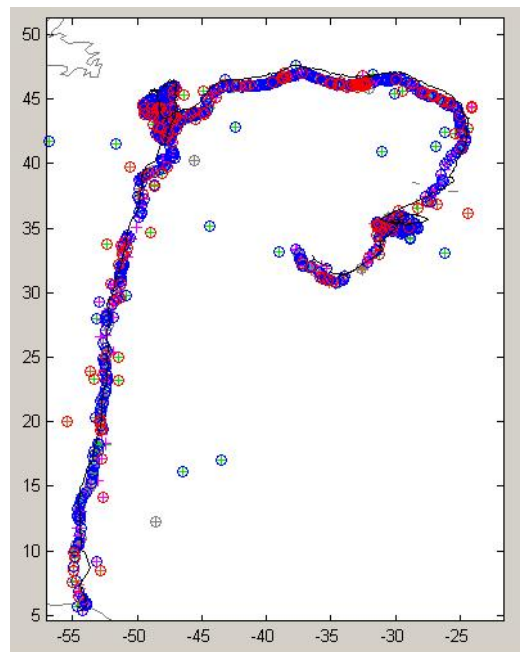
Il n'a pas pu être encore possible à ce jour de réaliser l'étape d'expertise. C'est ainsi qu'avec les valeurs de bruits d'edynamique définies au debut, le filtre a été testé sur des trajectoires rééles, même si celui n'est pas encore optimal. Le tracé des figures sera effectué avec l'interface plotres, réalisé par Béatriz Calmettes et qui permet d'observer plusieurs paramètres comme la vitesse de nage de la tortue par exemple. Il permet aussi de comparer la trajectoire obtenue avec celle obtenue par d'autres méthodes d'estimation, comme la régression linéaire.

6.1 Résultats observés pour quelques trajectoires

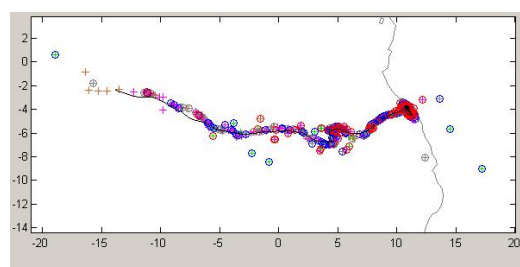
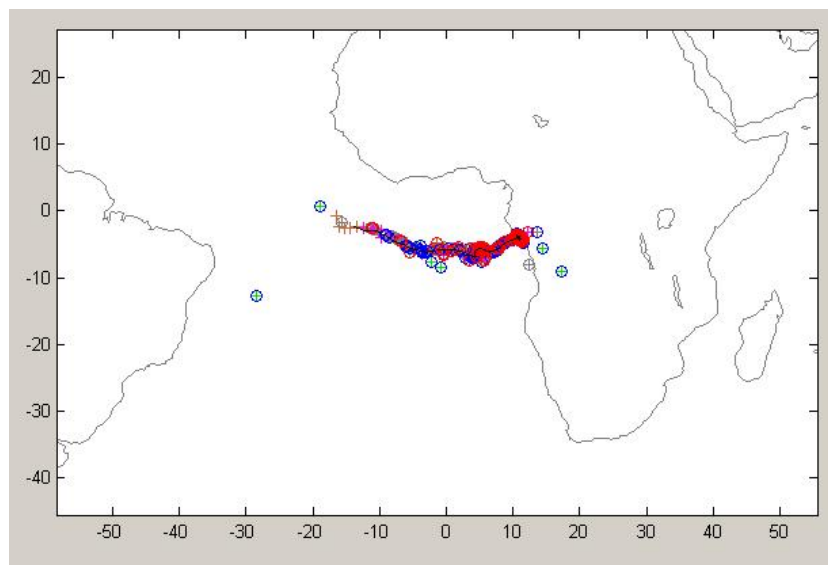
Voici des trajectoires obtenues pour quelques tortues. Les points éliminés lors du traitement sont ceux entourés. La légende donne les raisons d'élimination du point sur le premier graphique et les positions initiales sont encore présentent sur les figures suivantes :

Résultats pour tortue 8 :

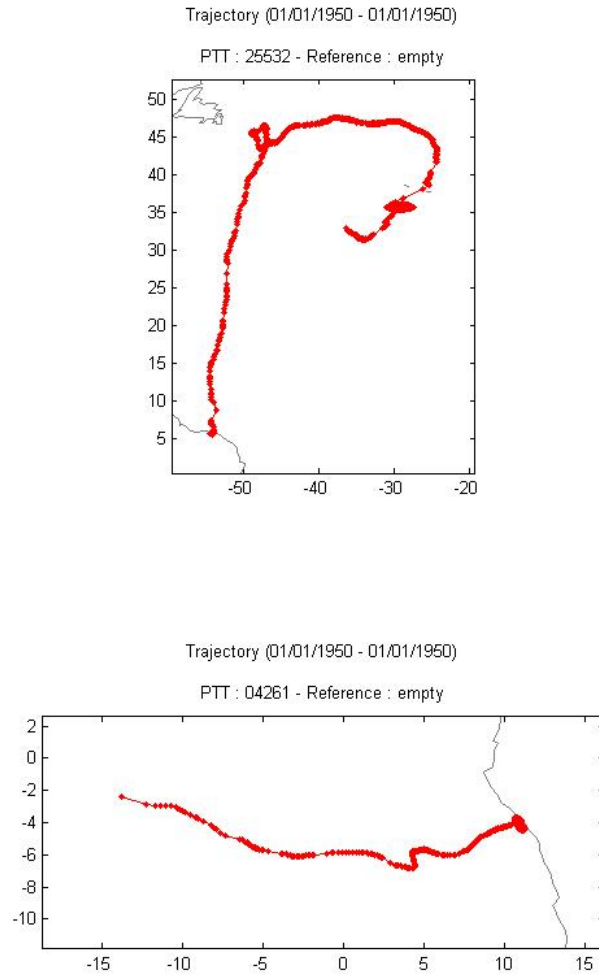




Résultats pour cepe 2 :



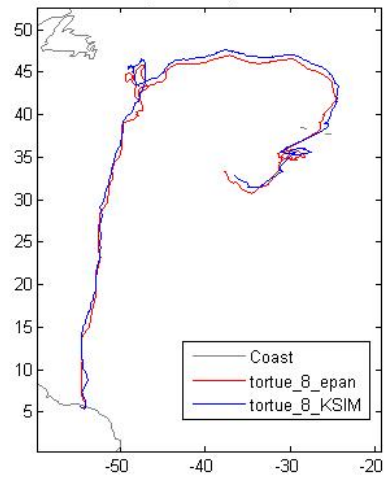
On ne visualise sur ces graphes que les trajectoires restituées par le filtre :



6.2 Comparaison avec d'autres méthodes d'estimation

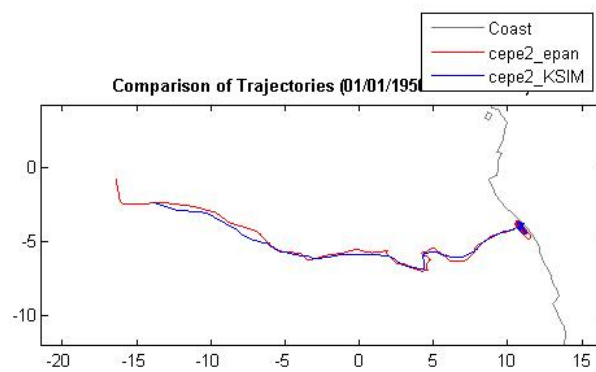
Nous comparons ici les trajectoires obtenues avec le filtre de Kalman et celles obtenues par la méthode de lissage d'Epanechnikov, méthode mise au point par Julien Vandrstraten, précédant stagiaire. Les trajectoires estimées avec les deux méthodes sont représentées sur la figure suivante : Pour tortue 8 :

Comparison of Trajectories (01/01/1950 - 20/04/2001)



Il semble ici que que les deux méthode ne réstituent pas la même trajectoire apres le virage effectué par l'animal. Apres vérification, il semblerait que la trajectoire obtenue par la méthode de lissage soit plus pres des données que ne l'est celle obtenue par filtrage. Cela peut fortement venir du fait que notre filtre n'est pas encore assez "tuné" et ne possède pas le sbonnes estimations de bruits de dynamique.

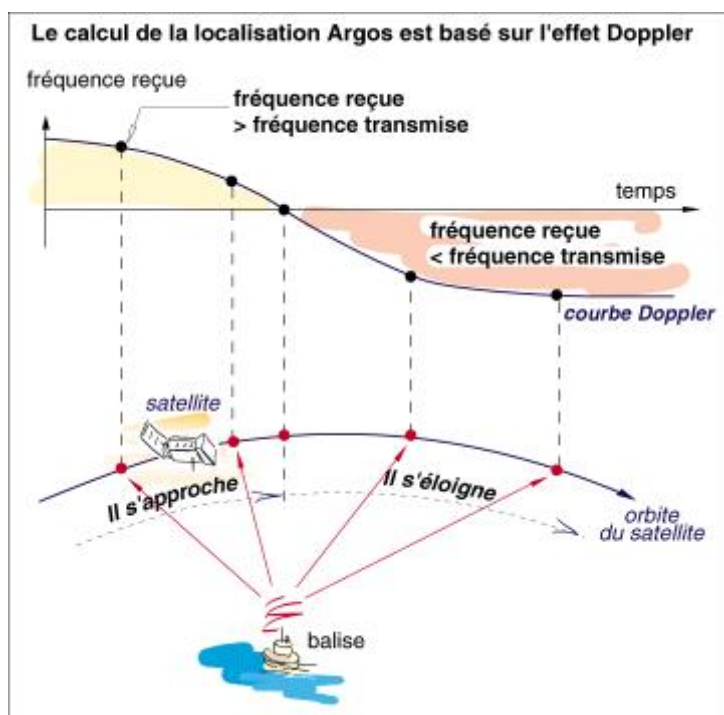
Pour cepe2 :



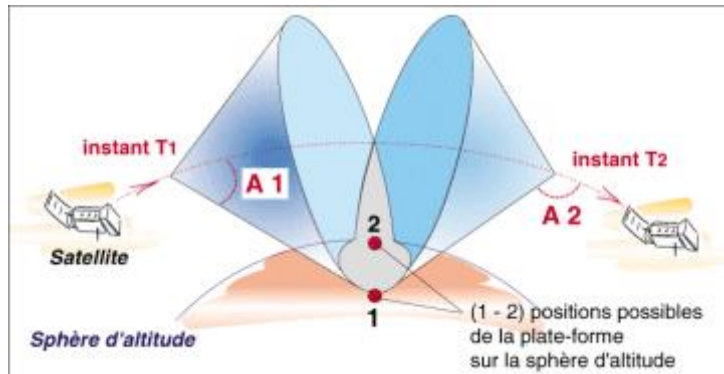
7 Annexes

7.1 Principe de fonctionnement des balises Argos

La localisation réalisée par le système Argos est basée sur la mesure de l'effet Doppler-Fizeau. Cet effet se traduit par le changement de fréquence d'une onde sonore ou électromagnétique lorsque la source des vibrations et l'observateur sont en mouvement l'un par rapport à l'autre. Lorsque le satellite se "rapproche" de la plate-forme, la fréquence des ondes électromagnétiques mesurées par l'équipement embarqué est supérieure à la fréquence réelle d'émission ; le phénomène est inversé lorsque le satellite s'éloigne de la plate-forme.



A chaque réception de message, l'équipement embarqué Argos effectue une mesure de la fréquence et date l'arrivée de ce message. Pour chaque message, le lieu des positions possibles de l'émetteur est un cône dont le sommet est la position du satellite à la réception du message et l'ouverture fonction de l'écart entre la fréquence reçue et la fréquence de l'émetteur.



Une première estimation de la position de la plate-forme est calculée à partir des premier et dernier messages collectés au cours du passage et de la dernière fréquence calculée de l'émetteur. L'intersection des deux cônes correspondant à ces deux messages et du globe terrestre, augmenté de l'altitude déclarée pour l'émetteur, donne deux positions possibles. Pour chacune de ces deux positions, la résolution de l'ensemble des équations de mesure par la méthode des moindres carrés permet de préciser la position de l'émetteur ainsi que sa fréquence. La solution choisie parmi les deux correspond à celle donnant la meilleure continuité en fréquence. Ce choix est ensuite confirmé (ou infirmé) par des tests de vraisemblance. Dans le jargon Argos ce deuxième point est appelé point "image"

Nous détaillons dans ce tableau quelles sont pour chacune des classe de localisation, les distances de fiabilité de la position correspondante.

Classe de localisation	Précision correspondante
3	$< 150m$
2	entre 150 m et 350 m
1	entre 350 m et 1000 m
0	$> 1000m$
<i>A</i>	Pas d'estimation
<i>B</i>	Pas d'estimation
<i>Z</i>	Pas d'estimation

Les localisations de classes Z sont très souvent de mauvaise qualité et ne sont pas utilisées par les méthode d'estimation. Il est observable, au vue des données reçus, que même si les localisations de classe A et B n'ont pas d'estimation d'erreur, elles peuvent être de meilleure qualité que certaines localisation de de classe 1, 2 et 3.

7.2 Description des principales routines écrites sous Matlab

- argoschi2b : effectue le stets de chi2 noté b dans l'ouvrage de Yaakoc Bar Shalom, sur des vraies données Argos
- argoschi2c : idem pour le test c
- FiltreK : filtre destiné à être transmis à Beatriz Calmettes, ne comporte pas de pré-traitement des données.
- FiltreKalmanArgos : effectue les pré-traitement et le filtrage de Kalman de vraies données Argos
- FiltreKalmanchi2 : effectue les pré-traitement et le filtrage de Kalman de données simulées, destinées à être testée par le test de chi2
- meserror : crée des classe de localisation, avec des probabilités observées dans la réalité, pour les imulations
- param2 : determine la moyenne et la covariance de la vitesse selon u et v initiale fonction de l'angle de départ
- param : idem lorsque la vitesse moyenne de départ de la tortue n'est pas connue
- removenan : Elimine les données qui n'ont pas de valeurs pour les positions ou la classe de localisation
- sansvitesseexcess : elimine les positions qui entrainent une vitesse excessive
- SwitchLoc : Echange les localisation/Localisation image lorsqu'il y a une erreur Argos
- testchi2a : test de chi2 pour une trajcetoire simulée et filtrée par Filtrechi2
- testchi2b : idem
- testchi2c : idem

8 Conclusions

Ce stage basé sur l'estimation de trajectoire devait permettre de réaliser un filtre de Kalman. Le filtre réalisé au cours de ce stage prend pour modèle de dynamique un déplacement non perturbé par une action extérieure, et les trajectoires sont ensuite corrigées, avec les données du courant marin, pour obtenir la trajectoire "réelle" de la tortue, celle qu'elle aurait suivi si elle n'avait pas été poussée par le courant. Une autre approche à étudier serait donc de rajouter à l'équation de dynamique le terme d'action du courant, qui nécessite de connaître celui-ci tout au long de la trajectoire de l'animal. Nous pouvons aussi remarquer que les traitements préalables, qu'il s'agisse d'élimination de localisations image ou de positions impliquant des vitesses excessives, pourraient être évités si le filtre pouvait lui-même corriger ces erreurs. Ainsi, il n'y aurait pas de perte d'information par élimination systématique des positions éloignées. La partie apprentissage du filtre sera l'étape suivante du travail qui a été réalisé, afin d'obtenir un filtre optimal. Il serait aussi intéressant d'obtenir un filtre capable de changer de modèle de dynamique lorsque la tortue a effectivement modifié son mode de déplacement, ce que l'on peut observer après des morceaux de trajectoires rectilignes (migration) lorsque la tortue semble se déplacer aléatoirement (nourissage).

Références

- [1] Yaakov Bar Shalom, Xiao-Rong Li **Estimation and tracking : Principles, techniques and software**, Artech house, 1998.
- [2] Xiao-Rong Li, Vesselim P. Jilkov, **Survey on maneuvering Target Tracking**, IEEE Transactionn on aerospace and electronic systems.
- [3] Peter.S Maybeck, **Stochastic models, estimation and control** Academic Press.
- [4] Brian D.O Anderson, John B. Moore, **Optimal filtering**.