

# Repérage grâce à un système de balise à ultrason et filtrage

Clément Besnier

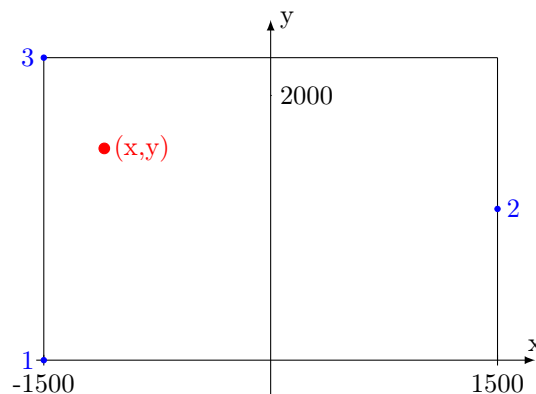
22 mai 2016

## Table des matières

1	Introduction	1
2	Système de balisage	1
3	Repérage	1
4	Modélisation et Simulation	2
5	Filtrage	3
6	De la mesure venant des balises à la mesure qui rejoint le modèle	3
7	Annexe I : Notations	3
8	Annexe II : mesure vers état	3

## 1 Introduction

Chaque année a lieu la coupe de France de robotique. Durant cette compétition, deux robots évoluent sur une table de deux mètres sur trois en 90 secondes pour obtenir un maximum de points. La table étant partagée en même temps par deux robots, il est nécessaire d'éviter le robot adverse pour des raisons de sécurité et pour des raisons stratégiques. C'est la raison même d'un tel système de balisage.



## 2 Système de balisage

Le système de balisage est composé d'une partie émettrice et d'une partie réceptrice. La partie émettrice (PE) est placée sur le robot adverse et émet des ultrasons. La partie réceptrice (PR) est composée de trois balises qui se trouvent autour de la table (Voir figure 1). Les trois balises sont synchronisées entre elles mais ne le sont pas avec la partie émettrice.

## 3 Repérage

PE émet à intervalle régulier un signal particulier. Soient  $t_1, t_2, t_3$  respectivement les dates de réception du même signal par PR. Comme PE et PR ne sont pas synchronisées il est impossible de calculer la durée entre l'émission du signal et sa réception et donc on ne peut pas trianguler le signal pour en retrouver l'origine. Cependant, on est certain des trois valeurs suivantes :

$$\begin{aligned} t_1 - t_2 \\ t_1 - t_3 \\ t_2 - t_3 \end{aligned} \tag{1}$$

, ce qui est proportionnel à

$$\begin{aligned} m_1 &= d((x, y), PR1) - d((x, y), PR2) \\ m_2 &= d((x, y), PR1) - d((x, y), PR3) \\ m_3 &= d((x, y), PR2) - d((x, y), PR3) \end{aligned} \quad (2)$$

, il faut juste multiplier les différences de dates par la vitesse du son dans l'air dans des conditions normales. Les mesures  $m_1, m_2, m_3$  permettent de retrouver (x,y) en résolvant le système. Vous pouvez trouver en annexe les équations (pas belles). Mais ce n'est pas comme ça qu'on va en réalité trouver ce qu'on veut.

## 4 Modélisation et Simulation

Soit  $X_t$ , la position du robot à l'instant t.

$$X_t = \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (3)$$

Soit  $Y_t$  la mesure prise à l'instant t.

$$Y_t = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \quad (4)$$

Soit H la fonction qui à une position (x, y) associe la différence de distance entre cette position et les balises suivant le schéma suivant :

$$H(x, y) = \begin{pmatrix} d((x, y), PR1) - d((x, y), PR2) \\ d((x, y), PR1) - d((x, y), PR3) \\ d((x, y), PR2) - d((x, y), PR3) \end{pmatrix} \quad (5)$$

On considère les  $Y_t$  ainsi parce que c'est plus clair de raisonner avec des distances qu'avec des différences de temps. On notera les valeurs estimées avec un accent circonflexe sur la lettre, les valeurs simulées avec un tilde. Soit  $E([t_1, t_2], PAS)$  une liste dont les éléments sont les multiples de  $t_1$  compris entre  $t_1$  et  $t_2$ . On considèrera dans la suite que l'unité de ses éléments est en seconde.  $[t_1, t_2]$  sera  $[0, 90]$  parce qu'un match dure 90 s. Le pas n'est pas encore décidé, cela dépendra des ressources disponibles du robot. On prendra pour l'exemple  $PAS = 0.025$  s.

Pour le filtrage, on considèrera ces équations avec Q, matrice de covariance de l'équation d'évolution et R, matrice de covariance de l'équation d'observation

L'équation d'observation est :

$$X_t = H(Y_t) + \mathcal{N}(0, R) \quad (6)$$

L'équation d'évolution est :

$$X_{t+1} = G(X_t) + \mathcal{N}(0, Q) \quad (7)$$

**Données:**  $C = [(x_t, y_t), t \in E([0, 90], 0.025)]$  le chemin,  $\sigma^2$  la variance

**Résultat:**  $MB = [(m_1(t), m_2(t), m_3(t))]$

**forall the**  $(x_t, y_t) \in C$  **do**

|  $m_1, m_2, m_3 = h(x_t, y_t)$   
|  $\tilde{m}_1, \tilde{m}_2, \tilde{m}_3 = m_1 + \mathcal{N}(0, \sigma^2), m_2 + \mathcal{N}(0, \sigma^2), m_3 + \mathcal{N}(0, \sigma^2)$   
| Ajouter  $(\tilde{m}_1, \tilde{m}_2, \tilde{m}_3)$  à MB

**end**

**Algorithm 1:** Simulation du bruitage de mesures.

## 5 Filtrage

On utilise ici un filtre de Kalman unscented.

**Données:**  $d = 2$  : dimension des mesures,  $\alpha = 1$ ,  $\beta = 0$ ,  $\kappa = 2$ ,  $\mu_0$  : position initiale,  $\Sigma_0$  matrice de covariance initiale,  $\lambda = \alpha^2 * (d + \kappa) - d$ ,  $\gamma = d + \lambda$ ,

$Q$  : matrice de covariance de l'équation d'évolution,  $R$  : matrice de covariance de l'équation d'observation,

$H$  est décrite telle que dans la partie 4 et  $G$  est la fonction identité.

$$w_m^0 = \frac{\lambda}{d+\lambda}, w_c^0 = w_m^0 + (1 - \alpha^2 + \beta), w_c^i = w_m^i = \frac{1}{2*(d+\lambda)}$$

**répéter**

Partie prédiction :

**début**

$\sqrt{\Sigma} = \text{decompositionDeCholesky}(\gamma * \Sigma);$   
 $\dot{\sigma}_{t-1} = [\mu_{t-1}; \mu_{t-1} - \sqrt{\Sigma}_{:,i} (i \in \{1, \dots, d\}); \mu_{t-1} + \sqrt{\Sigma}_{:,i} (i \in \{1, \dots, d\})];$   
**pour**  $i \in \{0, \dots, 2 * d\}$  **faire**  
     $\bar{z}_t^{*i} = G(\dot{\sigma}_{t-1}^i);$   
**fin**  
 $\bar{\mu}_t = \sum_{i=0}^{2*d} w_m^i * \bar{z}_t^{*i};$   
 $\bar{\Sigma}_t = \sum_{i=0}^{2*d} w_c^i (\bar{z}_t^{*i} - \bar{\mu}_t)(\bar{z}_t^{*i} - \bar{\mu}_t)^T + Q_t;$

**fin**

Partie mise-à-jour grâce à l'observation :

**début**

$\sqrt{\bar{\Sigma}} = \text{decompositionDeCholesky}(\gamma * \bar{\Sigma});$   
 $\bar{\sigma}_t = [\mu; \mu - \sqrt{\bar{\Sigma}}_{:,i} (i \in \{1, \dots, d\}); \mu + \sqrt{\bar{\Sigma}}_{:,i} (i \in \{1, \dots, d\})];$   
**pour**  $i \in \{0, \dots, 2 * d\}$  **faire**  
     $\bar{y}_t^{*i} = H(\bar{\sigma}_t^i);$   
**fin**  
 $\hat{y}_t^{*i} = \sum_{i=0}^{2*d} w_m^i * \bar{y}_t^{*i};$   
 $S_t = \sum_{i=0}^{2*d} w_c^i (\bar{y}_t^{*i} - \hat{y}_t^{*i}) * (\bar{y}_t^{*i} - \hat{y}_t^{*i})^T + R_t;$

**fin**

Partie estimation de l'état actuel :

**début**

$\bar{\Sigma}_t^{z,y} = \sum_{i=0}^{2*d} w_c^i (\bar{z}_t^{*i} - \bar{\mu}_t) * (\bar{y}_t^{*i} - \hat{y}_t^{*i})^T;$   
 $K_t = \bar{\Sigma}_t^{z,y} * S_t^{-1};$   
 $\mu_t = \bar{\mu}_t + K_t * (y_t - \hat{y}_t);$   
 $\Sigma_t = \bar{\Sigma}_t - K_t * S_t * K_t^T;$

**fin**

**jusqu'à plus aucune mesure;**

**Algorithm 2:** Algorithme de Kalman Unscented

## 6 De la mesure venant des balises à la mesure qui rejoint le modèle

Sylvain donne les mesures et obtient trois valeurs pour une salve de mesures qu'on va appeler  $t_1, t_2, t_3$ . J'ai vu que nous avons tous les 3 un positionnement des balises différent. J'ai donc fait des échanges comme il convient. Position (-1500, 0). Sylvain : 1. Clément : 1. Ivan : 2. Position (-1500, 2000). Sylvain : 2. Clément : 3. Ivan : 3. Position (1500, 1000). Sylvain : 3. Clément : 2. Ivan : 1. Donc il faut faire les échanges de valeurs en conséquence. Les valeurs sont en microsecondes donc il faut les convertir en secondes en divisant par  $10^6$ . Il faut convertir ce temps en distance (qui pour l'instant ne veut rien dire) en multipliant par la vitesse du son en millimètre par seconde. On obtient alors des distances que l'on appelle  $d_1, d_2, d_3$ . Il suffit de faire les différences de distance comme il suit :  $m_1 = d_1 - d_2, m_2 = d_1 - d_3, m_3 = d_2 - d_3$ .

Pour résumer, Ivan, tu dois faire :  $m_1 = (t_3 - t_1) * v_s / 10^6, m_2 = (t_3 - t_2) * v_s / 10^6, m_3 = (t_1 - t_2) * v_s / 10^6$ .

Pour résumer, et moi je dois faire :  $m_1 = (t_1 - t_3) * v_s / 10^6, m_2 = (t_1 - t_2) * v_s / 10^6, m_3 = (t_3 - t_2) * v_s / 10^6$ .

## 7 Annee I : Notations

Soit  $M$  une matrice de taille  $(m, n)$ . On note  $M_{:,i}$  le vecteur colonne correspondant à la colonne  $i$  ( $i \in \{1, \dots, n\}$ ) de  $M$ .

## 8 Annexe II : mesure vers état

Les trois équations du système représentent trois équations d'hyperbole.  $(x, y)$  se retrouvent à l'intersection de trois branches d'hyperbole. Cependant, la formule donnée à Mathematica (CHANGER LA FORMULATION) donne, pour chaque couple possible d'équation, deux solutions. Au total il y a 6 solutions.

• équation 1 1 2

$$\begin{aligned}
 a &= \sqrt{(-10 + m_1^2) * (m_1 + m_2)^2 * (-10 + m_2^2) * (-4 + m_1^2 - 2 * m_1^2 + m_2^2)} \\
 x &= \frac{-(-12 + 6 * m_1^2 - 12 * m_1 * m_2 + 3 * m_1^3 * m_2 + 6 * m_2^2 - 6 * m_1^2 * m_2^2 + 3 * m_1 * m_2^3 + a)}{4 * (-18 + 5 * m_1^2 - 8 * m_1 * m_2 + 5 * m_2^2)} \\
 Q &= 4 * (m_1 + m_2) * (-18 + 5 * m_1^2 - 8 * m_1 * m_2 + 5 * m_2^2) \\
 y &= \frac{-1}{Q * (72 * m_1 - 28 * m_1^3 + 72 * m_2 + 4 * m_1^2 * m_2 + m_1^4 * m_2 + 20 * m_1 * m_2^2 + m_1^3 * m_2^2 - 12 * m_2^3 - m_1^2 * m_2^3 - m_1 * m_2^4 - 3 * m_1 * a + 3 * m_2 * a)} \quad (8)
 \end{aligned}$$

• équation 2 1 2

$$\begin{aligned}
 a &= \sqrt{(-10 + m_1^2) * (m_1 + m_2)^2 * (-10 + m_2^2) * (-4 + m_1^2 - 2 * m_1 * m_2 + m_2^2)} \\
 x &= \frac{-(-12 + 6 * m_1^2 - 12 * m_1 * m_2 + 3 * m_1^3 * m_2 + 6 * m_2^2 - 6 * m_1^2 * m_2^2 + 3 * m_1 * m_2^3 - a)}{4 * (-18 + 5 * m_1^2 - 8 * m_1 * m_2 + 5 * m_2^2)} \\
 Q &= 4 * (m_1 + m_2) * (-18 + 5 * m_1^2 - 8 * m_1 * m_2 + 5 * m_2^2) \\
 y &= \frac{-1}{Q * (72 * m_1 - 28 * m_1^3 + 72 * m_2 + 4 * m_1^2 * m_2 + m_1^4 * m_2 + 20 * m_1 * m_2^2 + m_1^3 * m_2^2 - 12 * m_2^3 - m_1^2 * m_2^3 - m_1 * m_2^4 + 3 * m_1 * a - 3 * m_2 * a)} \quad (9)
 \end{aligned}$$

• équation 1 2 3

$$\begin{aligned}
 a &= \sqrt{(-10 + m_2^2) * (-2 * m_2 + m_3)^2 * (-4 + m_3^2) * (-10 + m_2^2 - 2 * m_2 * m_3 + m_3^2)} \\
 x &= \frac{-(-12 + 6 * m_3^2 + 3 * m_2^2 * m_3^2 - 3 * m_2 * m_3^3 - a)}{4 * (-18 + 2 * m_2^2 - 2 * m_2 * m_3 + 5 * m_3^2)} \\
 y &= \frac{-144 * m_2 + 16 * m_2^3 + 72 * m_3 - 56 * m_2^2 * m_3 + 4 * m_2^4 * m_3 + 80 * m_2 * m_3^2 - 8 * m_2^3 * m_3^2 - 28 * m_3^3 + 5 * m_2^2 * m_3^3 - m_2 * m_3^4 + 3 * m_3 * a}{4 * (2 * m_2 - m_3) * (-18 + 2 * m_2^2 - 2 * m_2 * m_3 + 5 * m_3^2)} \quad (10)
 \end{aligned}$$

• équation 2 2 3

$$\begin{aligned}
 [a &= \sqrt{(-10 + m_2^2) * (-2 * m_2 + m_3)^2 * (-4 + m_3^2) * (-10 + m_2^2 - 2 * m_2 * m_3 + m_3^2)} \\
 x &= \frac{-(-12 + 6 * m_3^2 + 3 * m_2^2 * m_3^2 - 3 * m_2 * m_3^3 + a)}{4 * (-18 + 2 * m_2^2 - 2 * m_2 * m_3 + 5 * m_3^2)} \\
 y &= \frac{-144 * m_2 + 16 * m_2^3 + 72 * m_3 - 56 * m_2^2 * m_3 + 4 * m_2^4 * m_3 + 80 * m_2 * m_3^2 - 8 * m_2^3 * m_3^2 - 28 * m_3^3 + 5 * m_2^2 * m_3^3 - m_2 * m_3^4 - 3 * m_3 * a}{4 * (2 * m_2 - m_3) * (-18 + 2 * m_2^2 - 2 * m_2 * m_3 + 5 * m_3^2)} \quad (11)
 \end{aligned}$$

• équation 1 3 1

$$\begin{aligned}
 a &= \sqrt{(-(-10 + m_1^2) * (2 * m_1 + m_3)^2 * (-4 + m_3^2) * (-10 + m_1^2 + 2 * m_1 * m_3 + m_3^2))} \\
 x &= \frac{-(-12 + 6 * m_3^2 + 3 * m_1^2 * m_3^2 + 3 * m_1 * m_3^3 + a)}{4 * (-18 + 2 * m_1^2 + 2 * m_1 * m_3 + 5 * m_3^2)} \\
 y &= \frac{-144 * m_1 + 16 * m_1^3 - 72 * m_3 - 8 * m_1^2 * m_3 + 4 * m_1^4 * m_3 + 16 * m_1 * m_3^2 + 8 * m_1^3 * m_3^2 + 12 * m_3^3 + 5 * m_1^2 * m_3^3 + m_1 * m_3^4 - 3 * m_3 * a}{4 * (2 * m_1 + m_3) * (-18 + 2 * m_1^2 + 2 * m_1 * m_3 + 5 * m_3^2)} \quad (12)
 \end{aligned}$$

• équation 2 3 1

$$\begin{aligned}
 a &= \sqrt{(-(-10 + m_1^2) * (2 * m_1 + m_3)^2 * (-4 + m_3^2) * (-10 + m_1^2 + 2 * m_1 * m_3 + m_3^2))} \\
 x &= \frac{-(-12 + 6 * m_3^2 + 3 * m_1^2 * m_3^2 + 3 * m_1 * m_3^3 - a)}{4 * (-18 + 2 * m_1^2 + 2 * m_1 * m_3 + 5 * m_3^2)} \\
 y &= \frac{-144 * m_1 + 16 * m_1^3 - 72 * m_3 - 8 * m_1^2 * m_3 + 4 * m_1^4 * m_3 + 16 * m_1 * m_3^2 + 8 * m_1^3 * m_3^2 + 12 * m_3^3 + 5 * m_1^2 * m_3^3 + m_1 * m_3^4 + 3 * m_3 * a}{4 * (2 * m_1 + m_3) * (-18 + 2 * m_1^2 + 2 * m_1 * m_3 + 5 * m_3^2)} \quad (13)
 \end{aligned}$$

## Références

- [1] Ali Boroujerdi and Jeffrey K. Uhlmann. An efficient algorithm for computing least cost paths with turn constraints. *Inf. Process. Lett.*, 67(6) :317–321, 1998.
- [2] Rudolph van der Merwe and Eric A. Wan. The square-root unscented kalman filter for state and parameter-estimation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7-11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings*, pages 3461–3464, 2001.