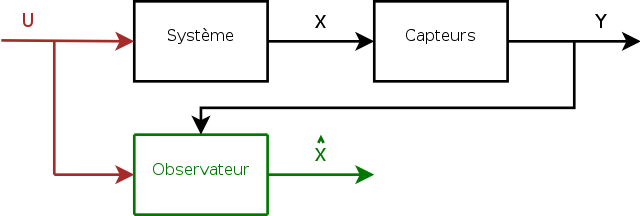
**Le filtre de Kalman**

En [automatique](http://fr.wikipedia.org/wiki/Automatique) et en traitement du signal, lorsque l'[état](http://fr.wikipedia.org/wiki/Repr%C3%A9sentation_d%27%C3%A9tat) d'un système n'est pas mesurable, on implémente un **observateur** qui permet de reconstruire l'état à partir d'un modèle du [système dynamique](http://fr.wikipedia.org/wiki/Syst%C3%A8me_dynamique) et des mesures d'autres grandeurs.

La théorie de l'observateur d'état [déterministe](http://fr.wikipedia.org/wiki/D%C3%A9terministe) a été introduite dans les années soixante par Luenberger pour les systèmes linéaires. [Kalman](http://fr.wikipedia.org/wiki/Kalman) a également formulé un observateur en considérant un système linéaire [stochastique](http://fr.wikipedia.org/wiki/Stochastique). Pour les systèmes non-linéaires, l'observation reste un domaine où la recherche est très active, mais l'utilisation la plus commune est l'emploi d'un filtrage de Kalman étendu (EKF).

Le filtre de Kalman estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Il s’agit d’un estimateur récursif. Cela signifie que pour estimer l'état courant, seul l'état précédent et les mesures actuelles sont nécessaires. L'historique des observations et des estimations n'est pas requis.

**Filtre de Kalman classique (systèmes linéaires)**

Les équations d’état d’un système linéaire s’écrivent :



Avec :

* k l’itération courante
* F,B les matrices qui relient respectivement l’état précèdent et la commande à l’état courant xk
* wk le bruit du processus, caractérisé par sa matrice de covariance Q
* H la matrice qui relie l’état courant à la mesure yk
* vk le bruit de mesure, caractérisé par sa matrice de covariance R

Remarquez que pour le système étudié on a :

avec  et on peut utiliser 

La sortie du filtre est représentée par deux variables :

*  l'estimation de l'état à l'instant k;
* la matrice de covariance de l'erreur (une mesure de la précision de l'état estimé).

Puisque c’est un filtre récursif, il doit avoir en entrée les valeurs initiales  et .  
Le filtre de Kalman a deux phases distinctes: **prédiction** et **mise à jour**. La phase de prédiction utilise l'état estimé de 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 afin d'obtenir une estimation plus précise.

**Prédiction**



**Mise à jour**



**Filtre de Kalman étendu (pour systèmes non-linéaires)**

Les équations d’état d’un système linéaire s’écrivent :



Avec f et h des fonctions différentiables.

La fonction f peut être utilisée pour calculer l'état prédit à partir de l'état estimé précédent et la fonction h pour calculer l'observation prédite à partir de l'état prédit. Cependant, f et h ne peuvent pas être appliqués directement au calcul de la covariance : une matrice des dérivées partielles, la **Jacobienne**, est calculée. À chaque instant, les Jacobiennes de f et h sont évaluées avec les états estimés courants et employées dans les équations du filtre. Ce processus linéarise essentiellement la fonction non linéaire autour de l'estimation courante. On obtient les équations du filtre de Kalman étendu suivantes :

**Prédiction**



**Mise à jour**



Où les [matrices](http://fr.wikipedia.org/wiki/Matrice_%28math%C3%A9matiques%29) de transition et d'observation sont définies comme étant les [Jacobiennes](http://fr.wikipedia.org/wiki/Jacobienne) suivantes :



Remarque : la convergence de ce filtre n'est aucunement assurée car il s'agit d'une convergence locale. En fait, il existe de nombreux exemples pour lesquels la convergence du filtre dépend de l'initialisation de l'état.

**Conseil pour l’experimentation**

Ecrivez ainsi vos fonctions matlab (dans des m-files):

function [ XfromSpeed, YfromSpeed ] = EstimateFromSpeeds ( X\_m, Y\_m, DX, DY, t, sigma)

function [ XKal, YKal, CovKalman ] = EstimateFromKalman ( InitState, X\_m, Y\_m, DX, DY, t, initCov, sigmaAcc)

function [ XKal, YKal, CovKalman ] = EstimateFromExtKalman ( InitState, X\_m, Y\_m, V, t, initCov, sigmaAcc)