Le filtre de Kalman : Recherches & Analyses

Soit un état représentant les paramètres que l'on souhaite déterminer d'un objet en déplacement.
Le filtre de Kalman est un procédé récursif se servant des **mesures de l'observation précédente** et de **l'estimation** de l'état par le filtre pour fournir le prochain état probable de l'objet en mouvement. En effet, le **gain de Kalman** joue le rôle d'un coefficient permettant d'attribuer plus d'importance à l'observation ou à l'estimation selon leur vraisemblance avec l'évolution du système. Par exemple, si une donnée très corrompue a été observée au milieu d'autres données standards alors le gain de Kalman ne donnera que très peu d'impact à cette valeur sur le modèle.
Maintenant que le fonctionnement grossier du filtre est compris, rentrons dans les détails.

Pour débuter le filtre de Kalman, il faut avant tout être en possession des **informations théoriques liées aux incertitudes** (qui représenteront les éléments diagonaux de la matrice de covariance de l'erreur de l'état estimé pour chaque paramètre de l'état) mais aussi d'un **état initial**. L'état initial contient les valeurs initiales de chacun des paramètres que l'on souhaite observer. Dans notre cas, ce sera la position sur les axes X et Y et la vitesse associée.

En partant de notre état initial, nous allons essayer de **prédire le prochain état du système** dans un laps de temps pré-définie (∇t) qui sera évidemment assez court pour que les rectifications de trajectoire du véhicule soient moins directes permettant un mouvement plus fluide. Dans le même intervalle, on **prédira l'évolution de la matrice de covariance des erreurs** en fonction des mouvements du système.

Une fois ces deux prédictions faites, nous allons **calculer le gain de Kalman** qui est représenté par une matrice diagonale dont chaque terme diagonal est compris entre 0 et 1 selon à quel point un paramètre influence le système. Plus on est proche de 1, et plus le paramètre influence le système (inversement vers 0). Faisons maintenant une **observation de l'état courant** (après ∇t secondes suivant l'état précédent) que l'on garde en mémoire.

On peut maintenant **calculer l'état « réel »** du système car le filtre de Kalman s'occupera de donner de l'importance à chaque paramètre ou pas. Cet état est renvoyé comme étant le plus proche de la réalité en prenant en compte l'observation et l'évolution théorique qu'aurait du avoir le système.

Enfin, pour terminer l'itération, on **met à jour la matrice de covariance de l'erreur de l'état estimé** dont les éléments diagonaux sont sensés converger vers 0 au fur et à mesure des itérations, preuve de la bonne convergence du filtrage de Kalman.

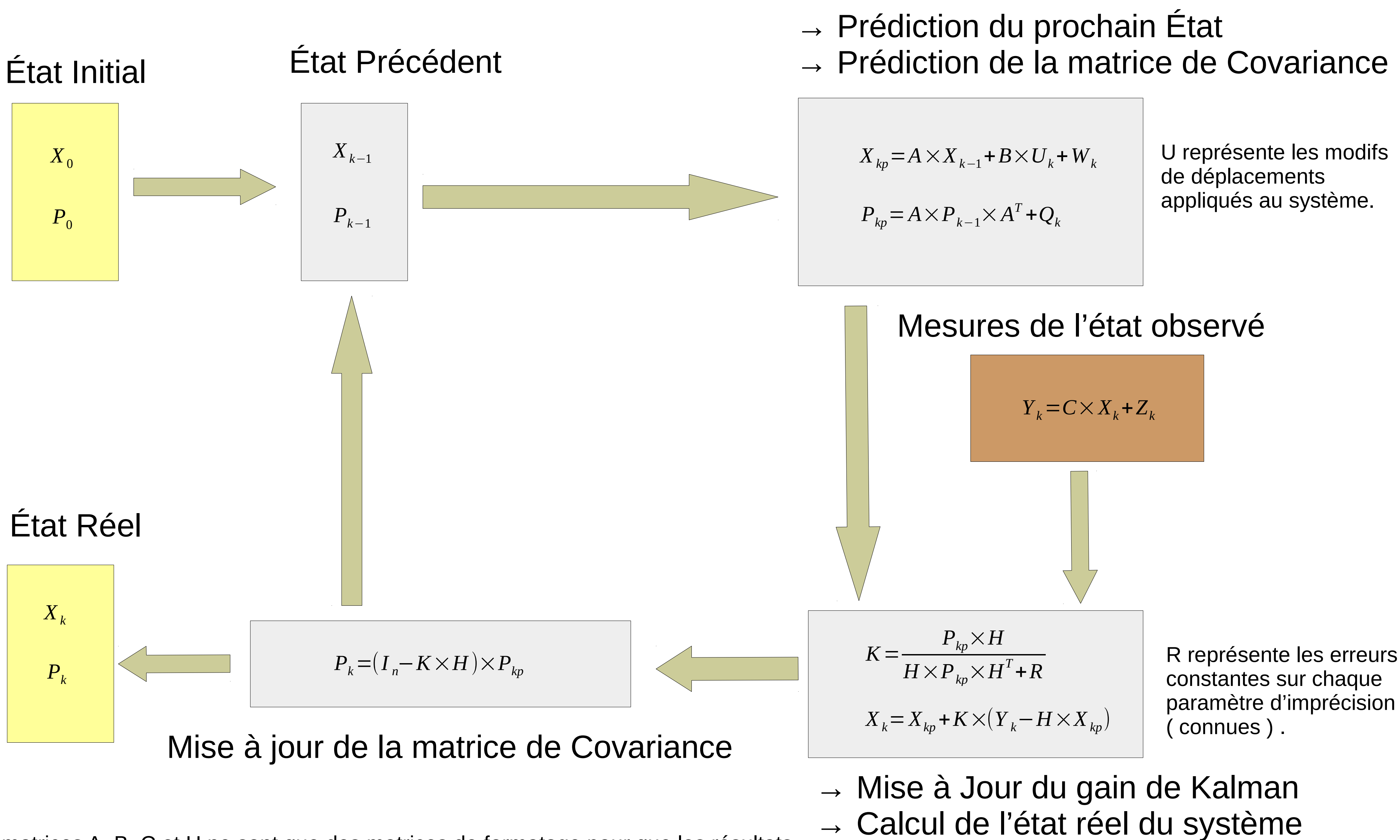
On **répète ce processus** en considérant le nouvel état calculé comme étant le précédent jusqu'à obtenir une précision que l'on considère comme suffisante mais cette méthode **converge très rapidement** sur des systèmes réels qui restent peu souvent chaotiques (ie les données observées peuvent être bruitées mais suivent une logique régie par les lois physiques de notre monde).

Modélisation de notre cas et axes de réflexion :

Dans notre cas, nous sommes dans un espace à 2D où l'on veut pouvoir être capable de déterminer la position du véhicule (x, y) et la vélocité sur chacun de ces axes (Vx, Vy). Il faudra donc partir d'un état Xo = (xo, yo, Vxo, Vyo) que l'on veut voir converger vers les valeurs réelles du véhicule dont on suit le déplacement. On sait par ailleurs que notre système peut en fait se modéliser grâce aux équations horaires dans le cas de mouvements uni/bidimensionnel non uniformément accélérés. L'accélération n'est donc pas constante (et dépendra des commandes reçues) par la voiture autonome, la vitesse se calculera ainsi $V_k = a_{k-1} \times (t_k - t_{k-1}) + V_{k-1}$ et enfin la position pourra être déterminée par l'équation $X_k = X_{k-1} + V_{k-1} \times (t_k - t_{k-1}) + \frac{1}{2} \times a_{k-1} \times (t_k - t_{k-1})^2$. Il ne reste plus qu'à adapter ces équations à un système matriciel et contextualiser lors de la réunion du 17.01 avec vous les dimensions du projets.

Questions :

- Langage de programmation : confirmez-vous Python 3 ?
- Filtre de Kalman : validez-vous nos recherches ci-dessus ?
- Contraintes sur les caméras : budget, connexion avec PC ?
- Reconnaissance des véhicules : apprentissage profond ?
- Concernant le circuit : est-il créé ? Toujours les mêmes dimensions (~ 2,5x2,5 m) ? Disponible durant les tests ?
- Objectifs : quels sont vos objectifs concrets ? Qu'attendez-vous de nous ?
- A-t-on un lien particulier avec les autres groupes intervenant dans le projet général ?
- Nous avons envisagé des réunions récurrentes pour valider l'avancement du projet. Seriez-vous d'accord de prévoir des réunions avec environ 2 semaines d'intervalle.



→ Les matrices A, B, C et H ne sont que des matrices de formatage pour que les résultats soient cohérents et aux bonnes dimensions pour les opérations matricielles. Elles permettent aussi de modéliser le fonctionnement du modèle étudié.

→Les matrices W, Q et Z sont des matrices permettant d'ajouter de l'incertitude dans des systèmes complexes où beaucoup de paramètres sont capables d'être influencés par des événements extérieurs (ex : gravité, vent, pente, …) ou par des erreurs de mesures (ex : temps de traitement de la donnée) .