

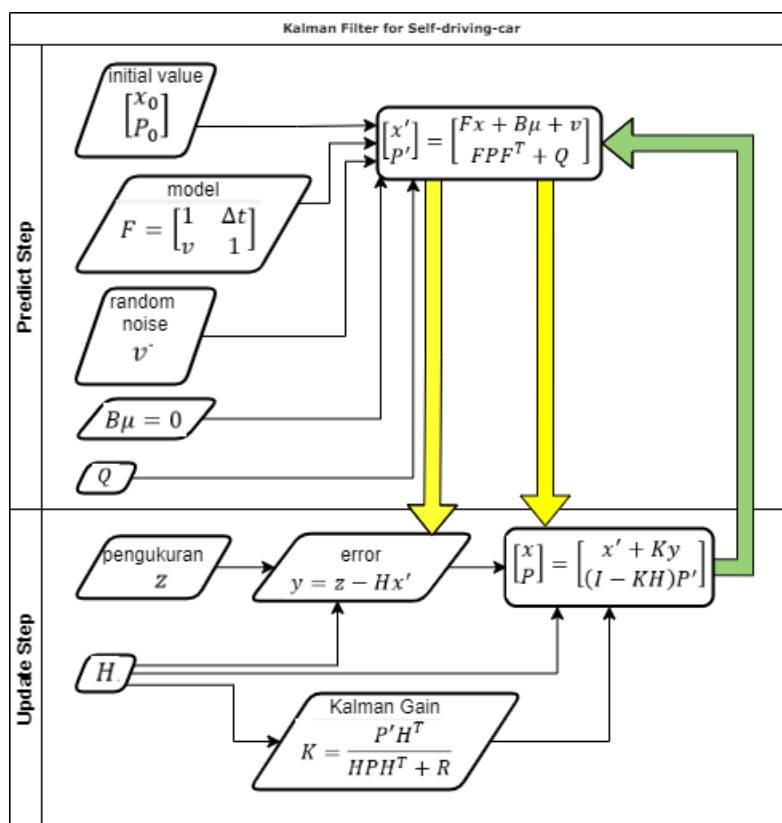
Kalman Filter for Self-Driving-Car Applications

Matematika Kalman Filter

Pada *self driving car*, dimisalkan yang akan diprediksi adalah posisi dan kecepatan kendaraan objek, maka mean dalam distribusi Gaussian mewakili vector posisi dan kecepatan yang dilambangkan dengan x dan varians dalam distribusi Gaussian mewakili *error*/ketidakpastian yang dilambangkan dengan P . Kemudian model dibangun dengan mengasumsikan kecepatan mobil konstan atau tidak berakselerasi maupun deselerasi, sehingga dengan menggunakan rumus Gerak berubah beraturan (GLB), diperoleh model sebagai berikut:

$$\begin{bmatrix} p' \\ v' \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ v & 1 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix}$$

Maka *state transition matrix*, $F = \begin{bmatrix} 1 & \Delta t \\ v & 1 \end{bmatrix}$.



Gambar 1. Flowchart 'Predict and Update step'.

Flowchart *predict step* dan *update step* seperti yang ditunjukkan oleh Gambar 1. *Prediction step* dihitung dengan memberikan random noise v , dan gain noise Q . *Random noise* diberikan untuk

mewakili noise yang mungkin ada di channel dan *gain noise* adalah faktor peningkatan ketidakpastian yang merupakan toleransi perhitungan pada keadaan dimana kendaraan objek mengganti arah atau melakukan akselerasi dan deselerasi. Selain itu terdapat *state transition matrix* dari model dan diasumsikan $B\mu$ yang merupakan *control input matrix* dan *control vector* adalah nol, karena dalam konteks *self-driving-car* tidak dapat dimodelkan gaya eksternal yang bekerja di kendaraan objek dari *ego vehicle* (EV). Maka pada iterasi pertama kita dapat memberikan nilai awal untuk mean dan varians untuk kemudian dihitung dengan melibatkan parameter-parameter diatas sehingga diperoleh mean dan varians baru.

Mean dan varians baru dari perhitungan kemudian akan diperbarui di *update step*. Pada step ini, diperoleh nilai pengukuran posisi dan kecepatan sehingga mean baru yang berisi matriks posisi dan kecepatan hasil perhitungan dapat dibandingkan dengan nilai pengukuran posisi dan kecepatan, dan diperoleh *error y*. Kemudian Kalman Gain dihitung dengan menggunakan *error* perhitungan dan *error* pengukuran (konstan dan diperoleh dari manufaktur). Nilai *Kalman Gain K* bervariasi antara 0 dan 1, nilai ini menunjukkan keyakinan pada nilai perhitungan atau nilai pengukuran. Jika *K* mendekati 0, maka nilai perhitungan mendekati nilai aktualnya, sedangkan jika *K* mendekati 1 maka nilai pengukuran mendekati nilai aktualnya. *Error* dan *Kalman gain* digunakan untuk memperbarui mean dan varian. Pada tiga persamaan tersebut juga melibatkan *state transition matrix H* yang memiliki cara kerja yang sama dengan *F*.

Setelah dilakukan *prediction step* dan *update step*, kita memiliki mean dan varian yang mempertimbangkan perhitungan dan pengukuran. Mean dan varian yang telah diperbarui di *update step* kemudian akan di berikan kembali ke *prediction step* dan mengulangi siklus *prediction – update* hingga *error*/ketidakpastian antara perhitungan dan pengukuran konvergen ke nol.

Referensi

[Kalman Filter Interview. I am currently into Term 2 of my Self... | by Harveen Singh Chadha | Towards Data Science](#)