

# Kalman Filter for Self-Driving-Car Applications

## Simulasi Kalman Filter

Pada tugas ini akan disimulasikan penggunaan Kalman Filter untuk *self-driving-car*, yaitu memprediksi posisi kendaraan objek dengan menggunakan bahasa pemrograman python. Posisi yang diprediksi dapat berupa posisi skalar atau jarak [1] dan posisi koordinat menggunakan library OpenCV [2]. Pada kasus ini kecepatan kendaraan objek dianggap konstan dan menggunakan timestep 0.5 s.

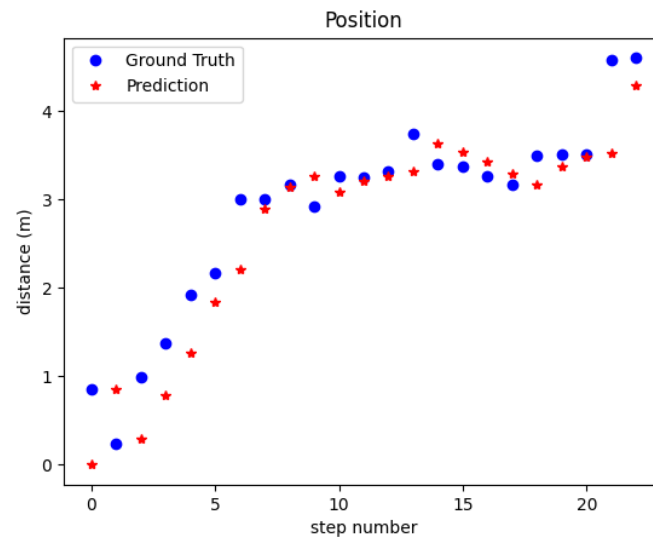
Pada *prediction step*, Kalman filter akan menghitung mean dan varians dengan mendefinisikan *state transition matrix*,  $F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ . *process noise*  $v$  diasumsikan nol untuk mempermudah simulasi, dan *gain noise*  $Q = GaG^T$ , dimana  $G = \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}$  dan  $a$  parameter akselerasi yang dimodelkan sebagai *zero-mean Gaussian noise*. Hal ini bertujuan sebagai toleransi perhitungan apabila kendaraan objek mengganti arah atau melakukan akselerasi dan deselerasi.

Pada *update step*, didefinisikan *state transition matrix*,  $H = [1 \ 0]$  yang memfilter pengukuran untuk memberikan pengukuran posisi dan *noise pengukuran*  $R$  sebagai *zero-mean Gaussian noise*. Pada bagian ini, nilai pengukuran adalah jarak yang dihitung dari dataset nuScenes [3] dimana dataset tersebut memberikan koordinat posisi tiap 0.5 s. Nilai pengukuran yang digunakan seperti ditunjukkan pada Tabel 1 dan hasil simulasi seperti ditunjukkan pada Gambar 1.

Tabel 1. Nilai pengukuran.

timestep	x,y-coord
0	[ 393.357 1149.173]
0.5	[ 392.945 1148.426]
1	[ 392.836 1148.208]
1.5	[ 392.641 1147.242]
2	[ 392.081 1145.988]
2.5	[ 391.347 1144.208]
3	[ 390.512 1142.201]
3.5	[ 389.29 1139.46 ]
4	[ 388.168 1136.675]
4.5	[ 386.878 1133.783]
5	[ 385.638 1131.14 ]
5.5	[ 384.005 1128.317]
6	[ 382.188 1125.617]
6.5	[ 380.066 1123.072]
7	[ 377.248 1120.613]

7.5	[ 374.393 1118.767]
8	[ 371.364 1117.297]
8.5	[ 368.254 1116.32 ]
9	[ 365.092 1116.247]
9.5	[ 361.627 1116.685]
10	[ 358.196 1117.398]
10.5	[ 354.824 1118.349]
11	[ 350.381 1119.451]
11.5	[ 346.001 1120.849]



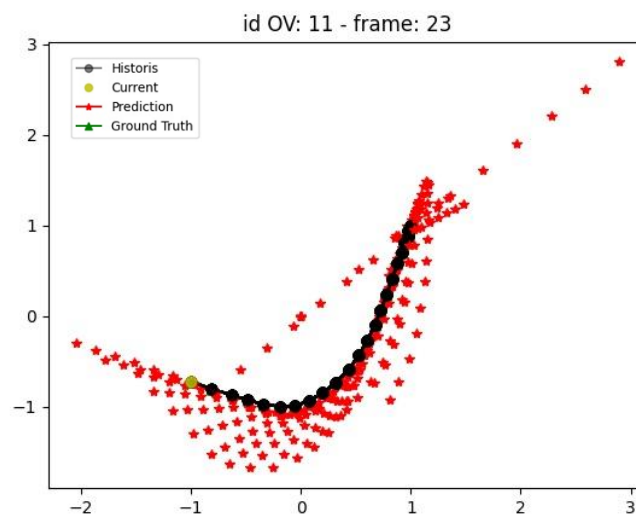
Gambar 1. Simulasi Kalman Filter untuk memprediksi Jarak.

Gambar 1 merupakan hasil simulasi Kalman Filter untuk memprediksi jarak salah satu kendaraan objek yang memiliki id 11. Titik berwarna biru adalah jarak yang dihitung dari pengukuran sebagai *ground truth* dan titik merah adalah jarak yang diprediksi Kalman filter sebagai *prediction*. Dalam simulasi ini diberikan nilai awal mean  $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$  dan varians  $P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ , serta noise pengukuran  $R = 0.1^2$  dan  $a = 0.1$ . *Gap* antara *ground truth* dan *prediction* terus mengecil seiring dengan banyaknya data pengukuran yang diperoleh dan diperoleh bahwa varians yang menyatakan *error*/ketidakpastian antara perhitungan dan pengukuran telah konvergen ke nol, yaitu

$$P = \begin{bmatrix} 0.00709919 & 0.00851589 \\ 0.00851589 & 0.029182 \end{bmatrix}$$

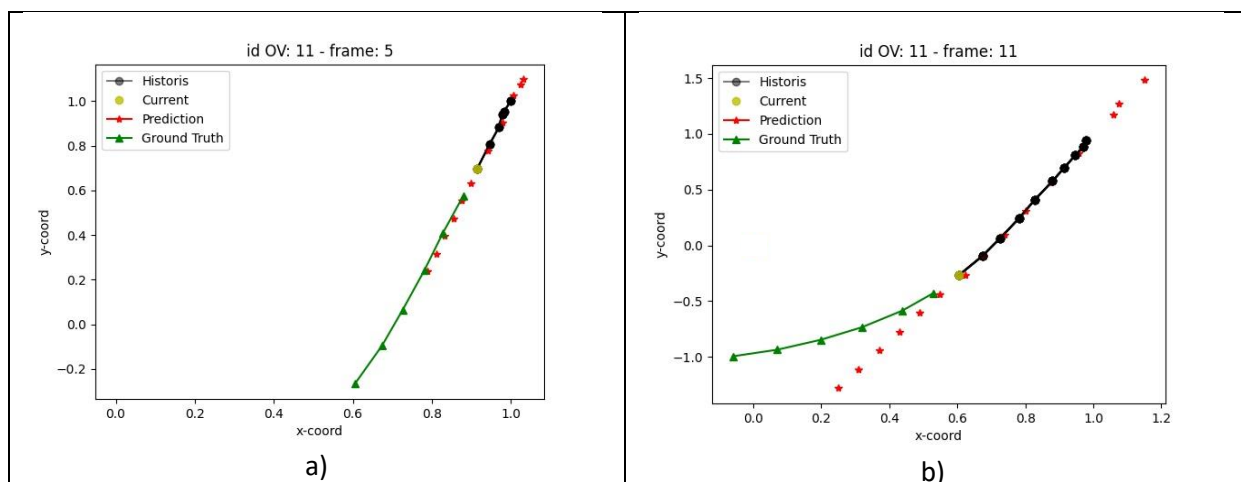
Hasil simulasi Kalman filter untuk memprediksi posisi dalam koordinat x-y seperti ditunjukkan pada Gambar 2 dan Tabel 2. titik hitam mewakili titik-titik historis kendaraan objek yaitu titik-titik yang telah dilalui kendaraan objek dan titik kuning mewakili titik *current position* kendaraan objek yaitu titik dimana kendaraan objek berada pada waktu ditinjau, sedangkan titik hijau mewakili *ground truth* yaitu titik-titik yang sebenarnya akan dilalui kendaraan objek. Prediksi diwakili oleh titik berwarna merah. Pada Tabel 1 (a) dengan 4 titik historis terlihat bahwa titik prediksi berada digaris yang sama dengan titik *ground truth* namun tidak pada posisi yang tepat. Pada Tabel 1 (b) titik historis bertambah menjadi 9 dan terlihat bahwa titik prediksi tidak sesuai dengan *ground truth*. Pada Tabel 1 (c) titik historis sejumlah 9 dan terlihat bahwa titik prediksi tidak sesuai dengan *ground*

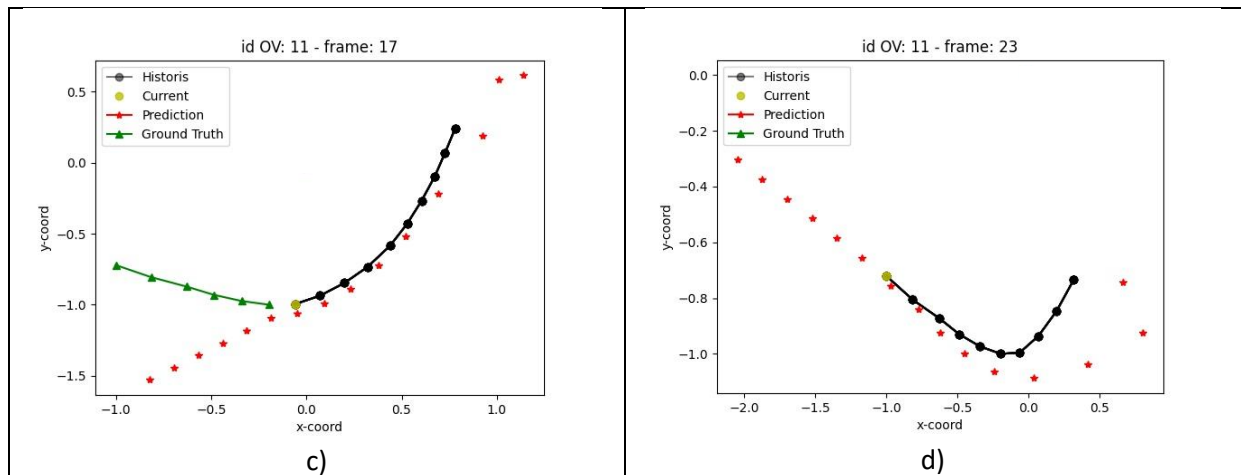
*truth*. Pada Tabel 1 (d) titik historis sejumlah 9 dan terlihat bahwa prediksinya adalah lurus namun tidak dapat diketahui kebenarannya karena Tabel 1 (d) adalah akhir dari pengukuran. Dari simulasi ini dapat diketahui bahwa prediksi dengan Kalman filter akan memperbarui prediksi tiap kali pengukuran baru diperoleh dan bahwa Kalman Filter yang diterapkan bersifat linear. Hal ini ditunjukkan dengan baiknya prediksi Ketika pergerakan kendaraan objek lurus, sedangkan prediksi tidak dapat ‘mengikuti’ *ground truth* ketika pergerakan kendaraan berbelok. Perubahan prediksi dari awal hingga akhir frame pengukuran seperti ditunjukkan pada Gambar 2. Berikut source code yang digunakan, [Khansa11/KalmanFilter: Tugas besar simulasi Kalman Filter untuk self-driving-car, motion prediction \(github.com\)](https://github.com/Khansa11/KalmanFilter).



Gambar 2. Perubahan prediksi dari awal hingga akhir frame.

Tabel 2. Prediksi tiap frame.





## Referensi

- [1] Tutorial Kalman filter by cppMonk. Link: [Understand & Code a Kalman Filter \[Part 2, Python\] - YouTube](#)
- [2] Kalman filter, predict the trajectory of an object by Sergio Canu. Link: [Kalman filter, predict the trajectory of an Object - Pysource](#)
- [3] Mini dataset nuScenes. Link: [nuScenes](#)