Kalman Filter

a. Pengertian

Kalman Filter merupakan sebuah teknik yang digunakan untuk mengestimasi nilai sebuah parameter atau *state* dalam suatu waktu tertentu. *State* adalah kumpulan variabel-variabel yang menentukan kondisi dari sistem yang diukur dalam suatu waktu tertentu. Akan tetapi, walaupun pengukuran berhasil dilakukan dan nilai *state* pada tiap satuan waktu dapat diperoleh, nilai yang dihasilkan ini cukup *noisy*, karena nilai dari setiap komponen *state* berubah-ubah tiap saat (sistemnya dinamis), dan terdapat ketidakpastian dalam pengukuran.

Teknik Kalman Filter digunakan untuk memperoleh taksiran nilai *state* tersebut dengan meminimalisir *noise* atau secara formal meminimalisir nilai ekspektasi dari *squad-error* (atau *Mean Square Error,* yaitu sebuah metrik yang umum digunakan sebagai *error function* untuk mengukur akurasi model) dengan memanfaatkan data-data yang diperoleh dari waktu sebelumnya.

Algoritma Kalman Filter adalah salah satu algoritma yang digunakan untuk memperkirakan hasil berikutnya berdasarkan data-data yang sudah ada sebelumnya. Algoritma ini biasanya digunakan untuk melakukan estimasi data sebenarnya berdasarkan data observasi yang mengandung noise dan beberapa faktor ketidaktepatan lainya.

b. Penggunaan Kalman Filter

Kalman Filter banyak digunakan dalam pengembangan teknologi, beberapa diantaranya adalah:

1. Sistem Kontrol
2. *Tracking* pergerakan objek dan estimasi lokasinya
3. Navigasi dan kendali pada robot dan rudal

c. Rumus Kalman Filter Linear

Kalman Filter cukup powerful karena dapat mengestimasi nilai *state* dari *past*, *present*, dan *future*, tanpa perlu mengetahui cara sistem bekerja, serta karena algoritma yang digunakan hanya bergantung pada statistik dari data yang diobservasi. Dalam Kalman Filter digunakan sebuah asumsi, yakni data pengukuran adalah distribusi Gaussian, sehingga *noise* nya dapat diasumsikan sebagai Gaussian *noise*. Algoritma ini biasanya digunakan untuk memodelkan sebuah sistem yang dinamis dan linear (*Linear Dynamic Systems*), yakni sebuah sistem yang dapat dinyatakan sebagai berikut:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

Filter Kalman adalah estimator kuadrat terakhir rekursif yang juga mencakup model gerak. Secara umum, proses pada Kalman Filter memakai konsep pemrograman dinamis serta rekursif, yang jika digabung serupa dengan *back propagation.*

Terdapat beberapa tahapan utama dalam algoritma Kalman Filter, yaitu:

1. *Prediction*

   Melakukan perhitungan estimasi nilai *state* pada waktu berikutnya

Time Update
(prediction)

1 Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

2 Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

2. *Correction*

   Dilakukan untuk melakukan smoothing serta mencari nilai optimal dari Kalman Gain dengan menggunakan measured value pada saat itu.

Measurement Update
(correction)

1 Compute the **Kalman Gain**

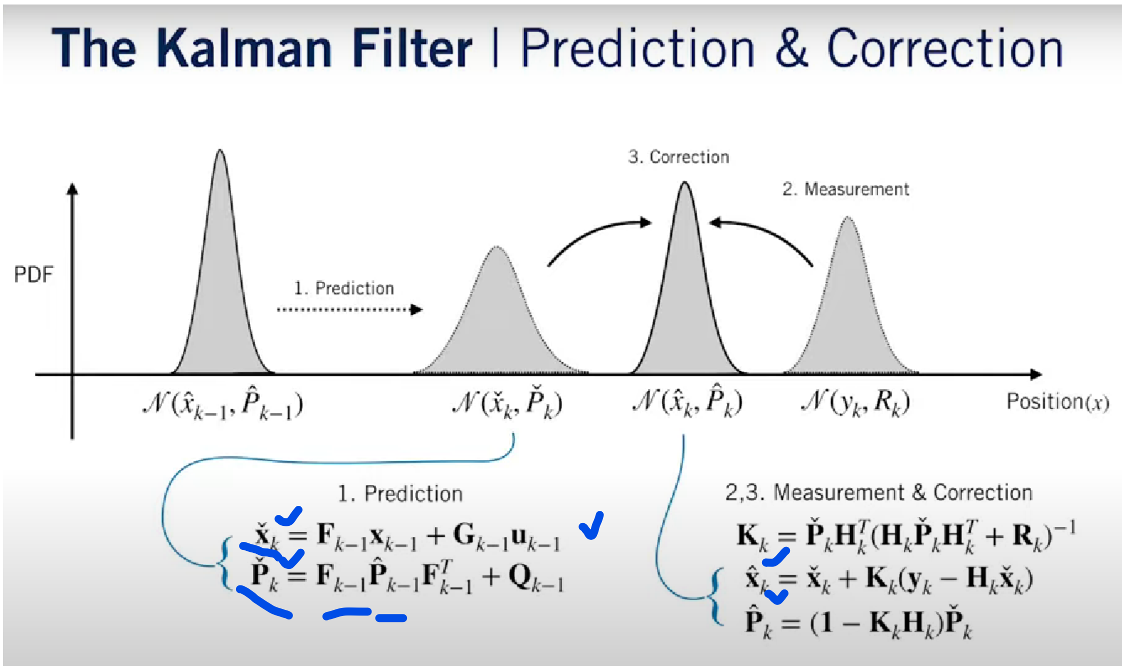
$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

2 Update the estimate via z_k

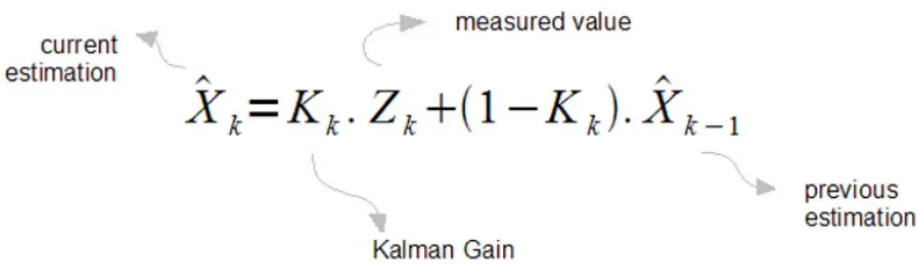
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

3 Update the error covariance

$$P_k = (I - K_k H)P_k^-$$



Karenanya Kalman FIlter disebut sebagai *predictor-corrector algorithm*.
Adapun persamaan umum dari Kalman Filter:



$$\hat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \hat{X}_{k-1}$$

Dalam hal ini, variabel X(k) merupakan estimasi nilai *state* pada waktu ke-k, variabel Z(k) merupakan *measured value* pada waktu ke-k, dan K(k) merupakan nilai dari koefisien Kalman Gain. Nilai koefisien ini lah yang perlu dihitung agar X(k) dapat mengestimasi nilai *state* lebih baik daripada Z(k), karena jika Z(k) sudah baik, kita mungkin tidak perlu lagi melakukan estimasi lebih lanjut. Estimasi ini dilakukan karena perhitungan Z(k) melibatkan ketidakpastian (noise), dan pada algoritma ini diasumsikan distribusi nilai dari data Z mengikuti distribusi Gaussian.

matriks transisi state

$$F_{k-1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

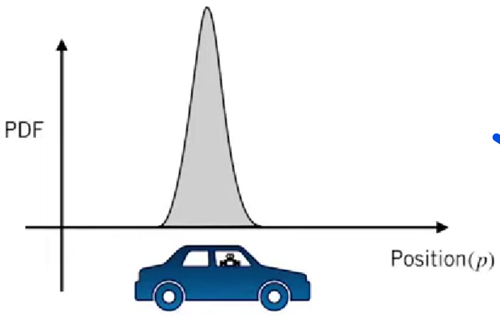
$$G = \begin{bmatrix} 0 \\ \Delta t \end{bmatrix}$$

matriks state kovariansi

$$P = \begin{bmatrix} \sigma_p^2 & \sigma_p \\ \sigma_v & \sigma_v^2 \end{bmatrix}$$

d. Contoh Soal Kalman Filter Linear

The Kalman Filter | Short Example



posisi

Motion/Process Model

$$\mathbf{x}_k = \begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix}\mathbf{x}_{k-1} + \begin{bmatrix}0\\ \Delta t\end{bmatrix}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

Position Observation

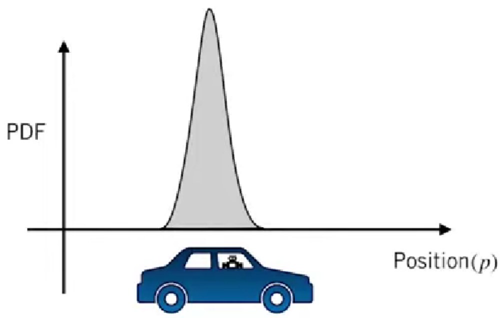
$$y_k = \begin{bmatrix}1 & 0\end{bmatrix}\mathbf{x}_k + v_k$$

$$\mathbf{x} = \begin{bmatrix} p \\ \frac{dp}{dt} = \dot{p}\end{bmatrix} \qquad \mathbf{u} = a = \frac{d^2p}{dt^2}$$

Noise Densities

$$v_k \sim \mathcal{N}(0,\ 0.05) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0},\ (0.1)\mathbf{1}_{2\times 2})$$

The Kalman Filter | Short Example



Data

$$\hat{\mathbf{x}}_0 \sim \mathcal{N}\left(\begin{bmatrix}0\\5\end{bmatrix}, \begin{bmatrix}0.01 & 0\\ 0 & 1\end{bmatrix}\right)$$

$$\Delta t = 0.5\text{s}$$

$$u_0 = -2\ [m/s^2] \qquad y_1 = 2.2\ [m]$$

The Kalman Filter | Short Example Solution

Prediction

$$\check{\mathbf{x}}_k = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + wP$$

$$\begin{bmatrix}\check{p}_1\\ \check{\dot{p}}_1\end{bmatrix} = \begin{bmatrix}1 & 0.5\\ 0 & 1\end{bmatrix}\begin{bmatrix}0\\5\end{bmatrix} + \begin{bmatrix}0\\0.5\end{bmatrix}(-2) = \begin{bmatrix}2.5\\4\end{bmatrix}$$

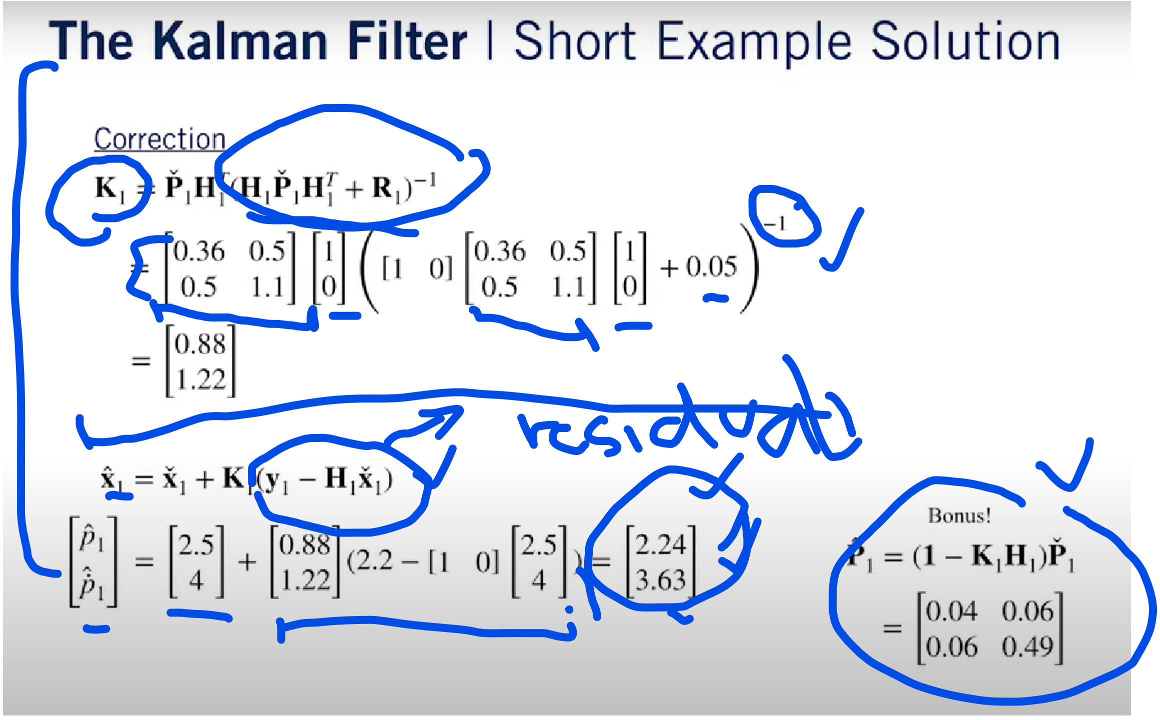
$$\begin{bmatrix}2{,}5\\5\end{bmatrix} + \begin{bmatrix}0\\-1\end{bmatrix}$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$$

$$\check{\mathbf{P}}_1 = \begin{bmatrix}1 & 0.5\\ 0 & 1\end{bmatrix}\begin{bmatrix}0.01 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0.5\\ 0 & 1\end{bmatrix}^T + \begin{bmatrix}0.1 & 0\\ 0 & 0.1\end{bmatrix} = \begin{bmatrix}0.36 & 0.5\\ 0.5 & 1.1\end{bmatrix}$$

$$Q = 0{,}1\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

varians prediksi



e. Non Linear

Dalam Kalman Filter, perubahan pada setiap iterasi dihasilkan dan diinterpretasikan dalam hal fungsi kepadatan probabilitas Gaussian. Proses inovasi dari KF terkait dengan filter yang mewakili prediksi baru dari informasi dalam lingkungan bising yang disampaikan ke estimasi keadaan dengan menggunakan data dalam pengukuran sistem terakhir. Ketika dinamika keadaan sistem (*state model*) atau dinamika pengamatan (*measurement model*) adalah non-linear, fungsi kepadatan probabilitas bersyarat yang memberikan perkiraan rata-rata kuadrat minimum tidak lagi Gaussian. Filter non-linear optimal menghasilkan propagasi fungsi-fungsi non-Gaussian ini dan mengevaluasi rata-ratanya, yang mewakili beban komputasi yang tinggi. Pendekatan non-optimal untuk memecahkan masalah ini dalam kerangka filter linear adalah Extended Kalman Filter (EKF).

Extended Kalman Filter (EKF) adalah perluasan dari Kalman Filter yang menangani dinamika sistem dan pengukuran yang non-linear. EKF mengatasi batasan linearitas dari Kalman Filter standar dengan melakukan linearisasi fungsi non-linear menggunakan matriks Jacobian. EKF memperkirakan fungsi non-linear menggunakan ekspansi deret Taylor orde pertama, dan kemudian menerapkan persamaan pembaruan Kalman Filter biasa untuk melakukan estimasi keadaan.

Bagian paling penting dari penerapan EKF adalah pembuatan model menggunakan pengetahuan matematika untuk membuat fungsi transisi yang non-linear untuk parameter yang tidak diketahui di setiap keadaan estimasi. Ada 2 model untuk EKF yakni:

$$\text{State Model} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k + \mathbf{w}_k)$$

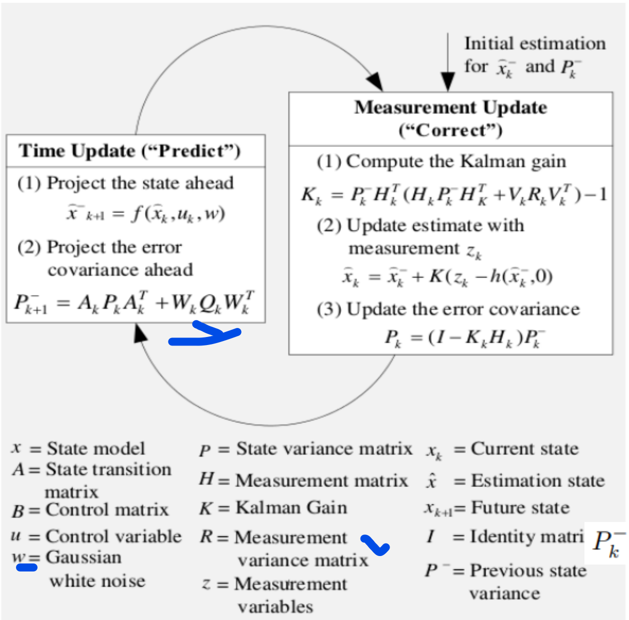
dengan:

x adalah *state model* yang terdiri dari parameter yang digunakan untuk prediksi pada setiap keadaan.
x_{k+1} adalah keadaan berikutnya (*next state*) yang harus kita dapatkan data prediksi dari menggunakan fungsi transisi yang merupakan fungsi non-linear.
u_k adalah data kontrol yang bersifat opsional.
w_k adalah Gaussian *white noise*.

$$\text{Measurement Model} \quad \mathbf{z}_k = h(\mathbf{x}_k + \mathbf{v}_k)$$

dengan:

z_k adalah *measurement model* yang mencakup parameter-parameter yang datanya berasal dari berbagai sensor.
v_k adalah Gaussian *white noise.*



Proses algoritma rekursif pada EKF ditunjukkan oleh bagan berikut. Ada dua bagian utama yaitu prediksi (*prediction*) dan keadaan koreksi (*correction state*). Proses algoritma rekursif EKF digambarkan dalam gambar 1. Terdapat 2 bagian utama, yaitu prediksi dan koreksi keadaan.

Pada tahap koreksi, menggunakan $\hat{\mathbf{x}}_k^-$ dan P_k^- yang diperoleh dari tahap prediksi, di mana $\hat{\mathbf{x}}_k^-$ dihitung menggunakan model transisi keadaan dan P_k^- ditentukan oleH pengguna untuk menghitung *Gain* Kalman atau K yang mewakili nilai yang dapat dipercaya dari model keadaan dan variabel pengukuran. Jika R mendekati 0, itu berarti variabel pengukuran lebih dapat dipercaya daripada model keadaan, tetapi jika P_k^- mendekati 0, maka sebaliknya. Setelah itu, data yang diestimasi diperbarui (*update*) berdasarkan nilai K. Akhirnya, error kovariansi diperbarui untuk iterasi selanjutnya. Oleh karena itu, semua parameter diperbarui pada setiap iterasi, sehingga data yang diprediksi dan diestimasi terus menjadi lebih akurat.

Fitur penting dari EKF adalah linearisasi fungsi non-linear f dan h menggunakan matriks Jacobian yang hampir merupakan turunan orde pertama. Contoh perhitungan matriks Jacobian untuk matriks *state transition* ditunjukkan sebagai persamaan berikut.

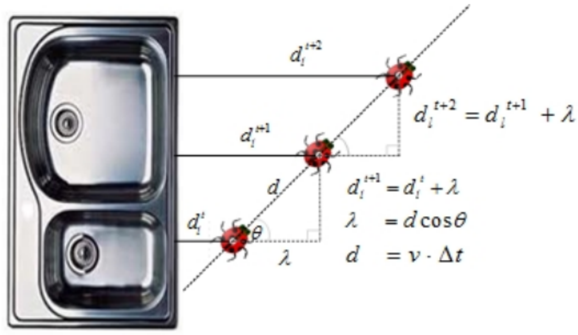
$$\mathbf{A}_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[i]}}(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0)$$

Jacobian dari fungsi-f_i (x_1, x_2, ..., x_n), i = 1, 2, ..., n dari variabel-variabel riil x_i adalah determinan dari matriks yang baris ke-i mencantumkan semua turunan parsial orde pertama dari fungsi f_i (x_1, x_2, ..., x_n).

Nilai estimasi untuk model keadaan dan model pengukuran berasal dari persamaan-persamaan yang ditunjukkan di bawah ini.

$$\mathbf{x}_{k+1} \approx \tilde{\mathbf{x}}_{\mathbf{k+1}} + A(x_k - \hat{x}_k) + Wwk$$
$$\mathbf{z}_k \approx \tilde{\mathbf{z}}_{\mathbf{k}} + H(x_k - \hat{x}_k) + Vvk$$

Pengaplikasian *Extended Kalman Filter* (EKF) menggunakan pergerakan kumbang pada garis lurus yang nampak pada gambar berikut.



1. Memahami status sistem saat ini

Seperti yang terlihat dapat dikatakan bahwa aplikasi ini melibatkan sistem dinamik (karena kumbang bergerak). Apabila diasumsikan ketika kumbang bergerak menjauhi wastafel, derajat sudut dan kecepatannya saat bergerak dalam lintasan lurus diukur. Pergerakan bersifat linear dan satu-satunya parameter yang berubah adalah jarak kumbang dari wastafel. Tujuan yang hendak dicapai adalah mengestimasikan jarak lateral di antara keduanya. Jika *noise* selama pengukuran terjadi karena peralatan pengukuran seperti alat pengukur jarak digital. Fungsi kosinus digunakan untuk membuat *state model*. Oleh karena itulah dalam kasus ini diterapkan *Extended Kalman Filter* (EKF) karena fungsi ini merupakan fungsi non-linear.

2. Melakukan permodelan *state process*

State model ditunjukkan dalam persamaan di bawah ini di mana x_t merepresentasikan kondisi sekarang (*current state*) dan x_{t+1} adalah state selanjutnya yang perlu diprediksi. Fungsi transisi yang ditentukan adalah $d_l + \lambda$ yang mengestimasikan jarak lateral kumbang terhadap wastafel.

$$\mathbf{x}^t = \begin{bmatrix} {d_l}^t \\ \theta^t \\ \mathbf{v} \end{bmatrix}, \mathbf{x}^{t+1} = \begin{bmatrix} {d_l}^t + \lambda \\ \theta^t \\ \mathbf{v} \end{bmatrix}$$

dengan:

x adalah *state model* (model kondisi)
d_l adalah jarak lateral antara sisi wastafel dan kumbang pada titik terdekat ke titik asal kumbang
θ adalah sudut tangensial sisi wastafel dari titik terjauh terhadap titik asal kumbang
v adalah kecepatan pergerakan kumbang

Matriks transisi A_t diperoleh dari matriks Jacobian untuk linearisasi yakni sebagai berikut.

$$A_t = \begin{bmatrix} 1 & \Delta_t * v * (-sin\theta) & \Delta_t * cos\theta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Untuk mempermudah demonstrasi, hal yang hanya didiskusikan disini adalah prediksi jarak lateral dengan jarak yang berubah terus menerus dan nilai lainnya dianggap konstan.

3. Mengukur permodelan proses
   Dalam hal ini, parameter pengukuran dibuat konstan untuk menyederhanakan kasus. Nilai konstan ini diperoleh dari alat ukur jarak digital.

   z adalah nilai yang terukur pada alat ukur jarak digital.
   H bernilai 1 karena tidak ada transformasi unit.

4. Melakukan permodelan *noise*
   KF dan EKF menggunakan Gaussian *white noise* yang membuat varian dan kovariansi berpengaruh. Noise diperkenalkan dalam filter seperti dalam kode Matlab.

   link : http://chadaphone.wordpress.com/2012/10/19/simple-example-of-applying-extended-kalman-implementation-with-matlab/
5. Pengujian Filter

   Model berhasil bekerja dengan baik dan nilai yang diestimasikan sangat dekat dengan nilai sebenarnya meskipun ada banyak noise.

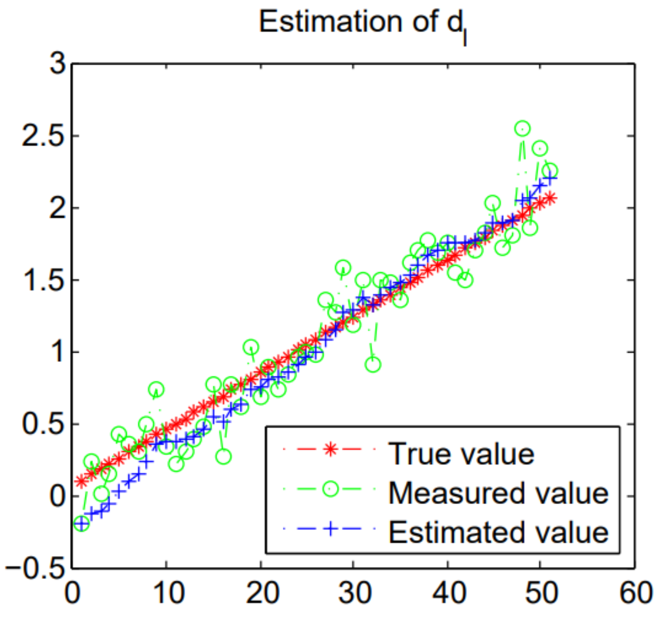


Fig. 3. A successfully working model simulation with R=0.05

6. Perbaikan filter (*filter refinement*)

EKF diperbaiki dengan mengubah nilai parameter *noise* seperti P (variabel varian keadaan) atau R (variabel varian pengukuran). Dalam eksperimen ini dilakukan pengubahan nilai R, yang merupakan kandidat yang dapat dipercaya dari variabel pengukuran, dan menganalisis pengaruhnya.

Mendefinisikan parameter *noise* menunjukkan bahwa setiap parameter dalam EKF penting untuk menyesuaikan tingkat akurasi model.

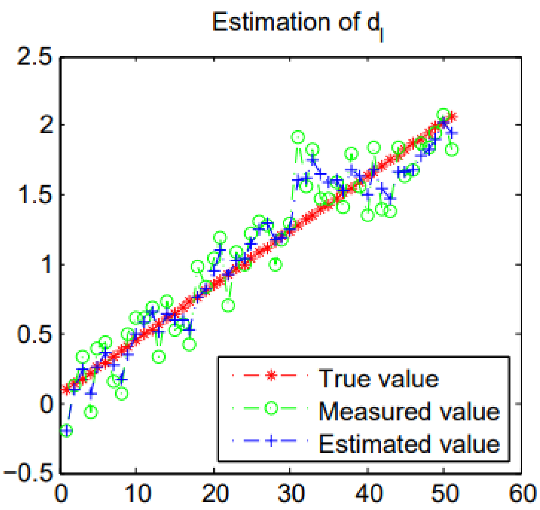


Fig. 4. The simulation with R=0.001

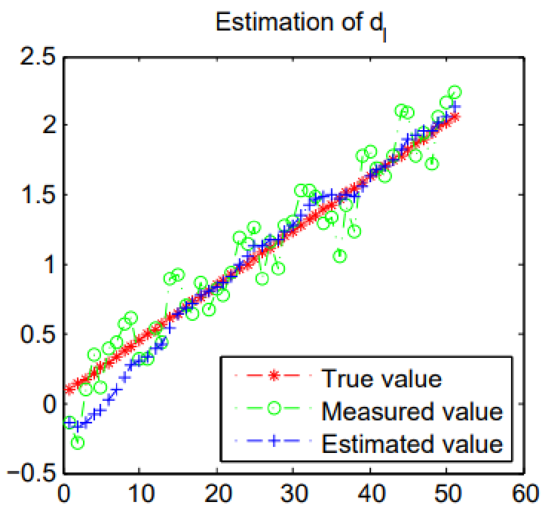


Fig. 6. The simulation with R=0.08

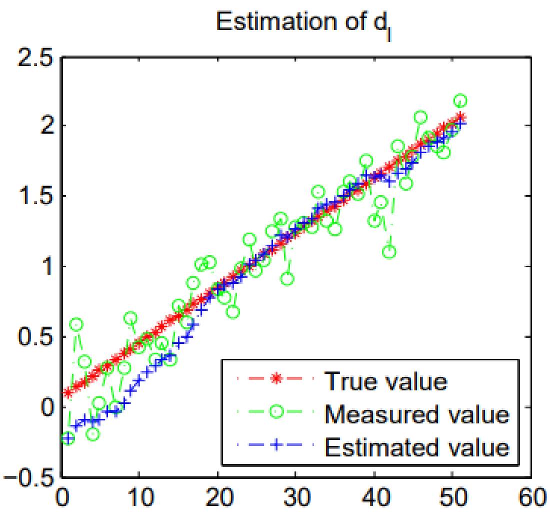


Fig. 5. The simulation with R=0.1

KALMAN FILTER

A. Pengertian