

## IMPLEMENTASI DAN PENGUJIAN ALGORITMA PARALLEL KALMAN FILTER PADA PLATFORM KOMPUTER MULTICORE

Arkan Muhammad Irsyad Sadeli

Teknik Komputer Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung (STEI ITB)  
arkan.irsyad@gmail.com

### ABSTRAK

Kalman filter merupakan algoritma penting dalam dunia pemrograman yang penggunaannya sangat luas. Dari mulai penggunaan pada pemrosesan gambar hingga ke kontrol navigasi pesawat terbang. Secara teoritis, Kalman filter jika diimplementasikan dalam algoritma sekuensial akan menghasilkan performa yang kurang optimal dari segi waktu, meskipun dalam platform komputer multicore. Hal ini dikarenakan kompleksitas algoritma kalman filter memiliki tingkat yang cukup tinggi yaitu perhitungan iterasi yang berulang-ulang. Pada kalman filter terdapat celah untuk modifikasi algoritma ke tingkat pemrograman parallel. Pada riset kali ini akan dianalisis bagaimana pengaruh dari program kalman filter yang sudah diparalelisasi pada platform multicore. Penilaian mencakup waktu eksekusi dari masing-masing program parallel maupun sekuensial.

Kata kunci; kalman filter; multicore; waktu eksekusi; algoritma parallel

### 1. Pendahuluan

Kalman Filter memiliki peranan penting dalam proses komputasi yang melibatkan banyak noise dalam sistem. Setelah 50 tahun penemuan kalman filter, saat ini masih menjadi algoritma yang penting dalam menggabungkan dua algoritma. Kelebihan dari kalman filter membutuhkan kebutuhan komputasi yang kecil, sifat iteratif yang baik, dan statusnya sebagai estimator optimal. Beberapa aplikasi kalman filter adalah untuk memperbaiki data yang memiliki banyak noise, penerima GPS, PLL pada equipment radio, penyelerasan output pada trackpad laptop dan banyak lagi yang lainnya.

Industri komputer sedang dalam era multicore dimana setiap komputer yang diproduksi umumnya memiliki inti yang lebih dari satu.

Secara alami, dengan performa yang besar, energi yang lebih rendah, disipasi daya yang rendah, arsitektur multicore mengharuskan beberapa program yang dijalankan bersifat parallel untuk meningkatkan performa speedup yang dijanjikan.

Dari kebanyakan algoritma sederhana kalman filter, implementasi yang dilakukan masih bersifat sekuensial, sehingga performa komputasi belum membuahkan hasil yang signifikan terhadap waktu pemrosesan. Untuk itu, diperlukan penelitian tentang modifikasi algoritma kalman filter linear dari pengkodean secara sekuensial ke tahap parallel. Ini dimaksudkan agar performa multicore yang dijanjikan dapat digunakan secara optimal pada pemrosesan kalman filter.

### 2. Tinjauan Pustaka

#### Model State Space

Model state space secara singkat didefinisikan sebagai bentuk perpindahan dari state-state yang akan kita teliti disaat input tertentu sehingga menghasilkan output tertentu. Jika sistem memiliki multi input maka penulisan input dapat dijabarkan dalam bentuk berikut:

$$u_1(t), u_2(t), \dots, u_r(t)$$

pada sistem multioutput, outputnya dapat dituliskan ke dalam bentuk berikut:

$$y_1(t), y_2(t), \dots, y_m(t).$$

setiap state dalam ruang dapat dituliskan dalam bentuk demikian.

$$x_1(t), x_2(t), \dots, x_n(t).$$

Sistem dapat didiskripsikan sebagai bentuk

$$\dot{x}_1(t) = f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_r; t)$$

$$\dot{x}_2(t) = f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_r; t)$$

⋮

$$\dot{x}_n(t) = f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_r; t)$$

Setiap next state adalah fungsi dari current state dengan inputnya. Sedangkan untuk output dapat dijabarkan sebagai fungsi dari current state dengan inputnya. Bentuk ini dijabarkan dalam bentuk berikut.

$$\begin{aligned}
 y_1(t) &= g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\
 y_2(t) &= g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 y_m(t) &= g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t)
 \end{aligned}$$

Maka, dengan menggunakan notasi ini, bentuk standar dari persamaan keadaan ruang adalah berikut:

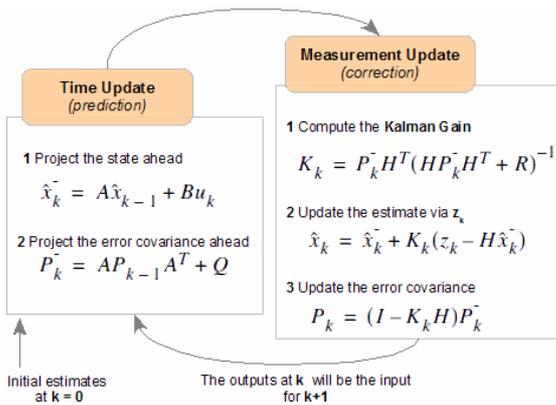
$$\begin{aligned}
 \mathbf{x}(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, & \mathbf{f}(\mathbf{x}, \mathbf{u}, t) &= \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix} \\
 \mathbf{y}(t) &= \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, & \mathbf{g}(\mathbf{x}, \mathbf{u}, t) &= \begin{bmatrix} g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \\ \vdots \\ g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \end{bmatrix}, & \mathbf{u}(t) &= \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}
 \end{aligned}$$

Bentuk sederhana dari persamaan matrix diatas adalah sebagai berikut:

$$\begin{aligned}
 \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\
 \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \\
 \dot{\mathbf{x}}(t) &= \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \\
 \mathbf{y}(t) &= \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t)
 \end{aligned}$$

**Prinsip Iterasi Kalman Filter**

Secara umum, prinsip dari algoritma kalman filter digambarkan dalam gambar 1



Gambar 1 Diagram Perhitungan Umum Kalman Filter

Kalman filter pada dasarnya ialah estimator, yaitu persamaan yang memprediksi keluaran dari sebuah ruang keadaan ke dalam hasil yang optimal. selain itu juga, kalman filter dapat membuat pengukuran dalam kecepatan yang

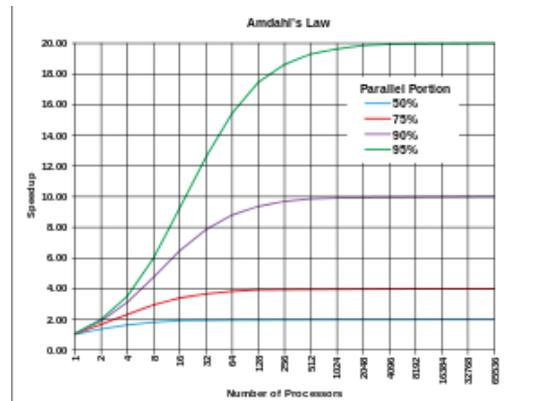
konstan terhadap data yang berubah dengan sangat cepat.

Kalman filter bersifat diskrit, sangat bergantung pada pengukuran yang terus menerus dan bersifat rekursif, artinya state berikutnya sangat bergantung pada nilai state yang ada sekarang

Pada kalman filter diasumsikan terdapat gangguan atau derau yang dimodelkan sebagai white noise. Tugas dari kalman filter ialah mengestimasi state x dari hasil pengukuran atau input.

**Prinsip Komputasi Berbasis Multicore**

Prosesor multicore adalah komponen komputasi tunggal dengan dua atau lebih independent CPU yang membaca dan mengeksekusi instruksi program. Keuntungan dari multiple core adalah mampu untuk menjalankan beberapa instruksi sekaligus. Penggunaan multicore sudah sangat luas mencakup penggunaan umum, embedded, network, DSP, dan grafik. Peningkatan performa pada komputer multicore sangat bergantung pada algoritma software yang mampu berjalan secara bersamaan/parallel pada platform multicore. Efek ini dijelaskan oleh hukum amdahl.



Gambar 2 Grafik Amdahl's Law

Dengan menggunakan hukum amdahl ini, dapat dilihat bahwa faktor speedup akan berbanding dengan jumlah prosesor. Hukum ini menjadi acuan untuk menurunkan waktu eksekusi dari sebuah pemrosesan instruksi dengan beban kerja yang cenderung tetap.

**3. Pengujian dan Analisa Implementasi dan Pengujian Algoritma Kalman Filter Sekuensial dan Paralel**

## Implementasi Dan Pengujian Algoritma Parallel Kalman Filter Pada Platform Komputer Multicore

Dalam menguji performa pada platform multicore, maka algoritma akan diuji pada komputer yang memiliki spesifikasi:

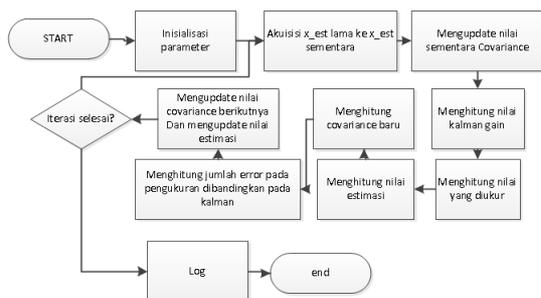
- Intel Core™2 Duo Processor T6500 2,1 GHz, 2MB L2 Cache, 800MHz FSB
- RAM DDR2 4096 MB 800MHz

Selanjutnya, skema dari pengujian algoritma kalman filter menggunakan 3 parameter yang penting: yaitu

$z_{real}$ : adalah parameter kebenaran yang hendak dicapai. Parameter ini seringkali dimodelkan sebagai kondisi ideal yang ingin dicapai dan menjadi patokan terhadap error pengukuran ataupun error estimasi

$z_{measured}$ : adalah parameter yang diukur melalui sebuah sensor atau alat pengukur/decoder/receiver data yang biasanya nilainya bergeser karena adanya derau putih (gaussian noise). Pada simulasi kali ini, nilai dari penghitungan  $z_{measured}$  adalah nilai  $z_{real}$  yang disimpangkan karena adanya noise dengan menggunakan generator angka acak, sehingga sulit untuk diprediksi.

$x_{est}$ : adalah nilai dari parameter yang diestimasi, nilai ini bergantung terhadap nilai estimasi terakhir sebelumnya dan nilai penguatan kalman.  $x_{est}$  ini merupakan nilai dari pengukuran yang mendekati kondisi ideal  $z_{real}$ . Diagram alir dari program algoritma kalman filter sekuensial ditunjukkan pada gambar 3



Gambar 3 Diagram Alir dari Algoritma Kalman Filter Sekuensial

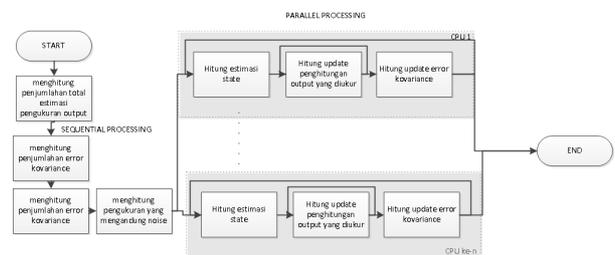
Hasil yang didapatkan pada 10.000 kali iterasi data di kalman filter menghasilkan waktu eksekusi.

Tabel 1 Waktu Eksekusi Algoritma Sekuensial

Percobaan	Total Pengurangan Error (pengukuran vs kalman)	Waktu Eksekusi(detik)
1	72%	0,34
2	71%	0,345
3	71%	0,345
4	71%	0,344
5	72%	0,335

	filter)	
1	72%	0,796
2	71%	0,78
3	71%	0,78
4	72%	0,794
5	72%	0,795
6	72%	0,811
7	72%	0,795
8	72%	0,764
9	71%	0,795
10	73%	0,779

Kemudian untuk membuat algoritma parallel pada kalman filter, dibutuhkan beberapa trik sederhana. Trik penyederhanaan ialah dengan menggunakan beberapa perhitungan yang diurutkan kembali untuk dikelompokkan berdasarkan keterhubungan antar instruksi. Jika M adalah jumlah CPU yang digunakan untuk mengujian, dapat diobservasi bahwa keterhubungan antara loop berada pada estimasi output, estimasi error kovariance, dan penghitungan nilai penguatan kalman. keterhubungan tersebut dapat dipecah dengan menggunakan reduksi, dimana setiap CPU menghitung kontribusi lokal yang nantinya dijumlahkan pada baris program sekuensial. Flowchart program parallel yang digunakan adalah



Gambar 4 Diagram Alir dari Algoritma Kalman Filter Parallel

Hasil yang didapatkan dari pengujian menghasilkan tabel 2.

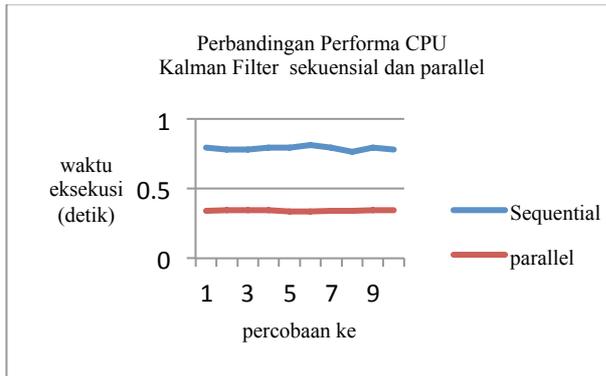
Tabel 2 Waktu Eksekusi Algoritma Parallel

Percobaan	Total Pengurangan Error (pengukuran vs kalman filter)	Waktu Eksekusi(mili detik)
1	72%	0,34
2	71%	0,345
3	71%	0,345
4	71%	0,344
5	72%	0,335

6	74%	0,335
7	72%	0,338
8	72%	0,34
9	71%	0,344
10	73%	0,344

**Implementasi dan Pengujian Algoritma Kalman Filter Sekuensial Penuh dan Semi Paralel**

Analisis yang dapat dilakukan dari data yang didapatkan pada pengujian algoritma kalman filter ini ditunjukkan pada gambar 5.



Gambar 5 Perbandingan Performa Speedup dari Sekuensial ke Paralel

Dari grafik ini, rata-rata waktu eksekusi untuk implementasi kalman filter sekuensial adalah 0,7889 dan untuk parallel adalah 0,34115. Sehingga speedup faktornya adalah 2,312.

Hasil ini menunjukkan bahwa nilai speedupnya tidak terlalu tinggi. Sifat dari paralelisme program ini adalah bersifat software dan paralelisme secara software menunjukkan hasil yang tidak signifikan dari performa speedupnya, ini menunjukkan bahwa dengan spesifikasi komputer multicore yang digunakan, performa paralelisme software tidak kentara dibandingkan dengan paralelisme yang dilakukan oleh hardware. Parallelisme hardware lebih berpengaruh disini. Karena semua eksekusi sudah dilakukan secara parallel dengan CPU yang multicore, hasil paralelisme secara software memiliki speedup yang sedikit. Selain itu beberapa faktor penyebab speedup yang tidak signifikan ini adalah faktor pengaksesan waktu shared memory dan ketidakseimbangan beban antara satu CPU dengan yang lainnya yang tidak dipertimbangkan dalam pengujian ini.

**4. Kesimpulan**

Dari pengujian dan analisis data yang dihasilkan, kesimpulan yang bisa diambil ialah

1. Kalman filter dapat dipecah dan diparallelisasi dengan menggunakan teknik reduksi yaitu pemrosesan sekuensial yang menangani jumlah kumulatif dari semua variabel yang menentukan pemrosesan dan memparallel instruksi yang bersifat iteratif.
2. Nilai speedup faktor dari perubahan algoritma sekuensial ke algoritma parallel adalah 2,312.
3. Penyebab speedup faktor yang tidak signifikan adalah perbandingan paralelisasi hardware lebih berpengaruh terhadap faktor speedup ketimbang hardware dan beberapa faktor lain yang tidak dipertimbangkan seperti performa akses shared memori dan ketidakseimbangan beban antara CPU.

**5. Daftar Pustaka**

- [1] Burns, Roland S., Advanced Control Engineering, Butterworth-Heinemann, 2001
- [2] Rosen, Olov, Efficient Parallel Implementation of Kalman Filter for Single Output Systems on Multicore Computational Platforms, IEEE Conference and Control and European Control Conference, Orlando, USA, 2011
- [3] Woods, John. Parallel Realization of 2-D Recursive Kalman Filter. Rensselaer Polytechnic Institute, New York, 1987
- [4] Glielmo L, Parallel Kalman Filter Algorithm for State Estimation in Bilinear Systems. Universita degli Studi di Napoli Federico II. Napoli.1994
- [5] Bell W. Jeffery. Simple Kalman Filter Alternative: The Multi Fractional. IET Radar, Sonar, and Navigation. USA. 2013
- [6] dipetik Oktober 30, 2013, dari [www.adrianboeing.com](http://www.adrianboeing.com)
- [7] dipetik Oktober 30, 2013 <http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx>