

IMPLEMENTASI SISTEM ADAPTIF ALGORITMA LMS PADA TMS320C25

Priyatmadi*, Bambang Sutopo**

* Staf Pengajar pada Jurusan Teknik Elektro – UGM, Jl. Grafika 2, Yogyakarta 55281
Telp. 0274-902202, email : pyatmadi@te.ugm.ac.id

** Staf Pengajar pada Jurusan Teknik Elektro – UGM, Jl. Grafika 2, Yogyakarta 55281
Telp. 0274-902202, email : bsutopo@te.ugm.ac.id

ABSTRACT

Sistem adaptif mampu mengatasi banyak masalah seperti nonlinieritas dan *time-variance*, namun untuk implementasi membutuhkan pemroses cepat. Pemroses sinyal digital (DSP) dapat memproses isyarat dengan kecepatan tinggi. Perlu dirancang program untuk implementasi sistem adaptif di DSP.

Digunakan kartu komputer Dalanco Spry model 250 yang berisi DSP TMS320C25, memori, ADC-DAC, dan antarmuka ke komputer untuk implementasi sistem adaptif. Algoritma adaptif LMS (Least Mean Square) dari Widrow sangat cocok bila diimplementasikan dengan DSP.

Program telah dirancang untuk sistem adaptif orde-6 dan membutuhkan waktu eksekusi perhitungan bobot $250\mu s$. Sistem adaptif sudah diuji untuk identifikasi sistem yang disimulasikan oleh komputer dan konvergensi bobot tercapai setelah iterasi ke 150. *Overflow* terjadi bila nilai konstanta μ terlalu besar. Sistem sudah dicoba untuk identifikasi invers sistem motor DC.

Kata kunci Adaptive, LMS, TMS320C25, motor DC.

I. PENDAHULUAN

Sistem adaptif menawarkan cara untuk mengatasi sistem yang mempunyai tingkat non-linearitas dan tingkat *time-varying* yang tinggi. Implementasi sistem ini membutuhkan kecepatan perhitungan yang lebih tinggi dari pada pesat pencuplikan. DSP diharapkan dapat melakukannya. Permasalahannya adalah implementasi algoritma adaptif itu ke dalam DSP agar dapat diaplikasikan pada sistem waktu nyata. Makalah ini mengungkapkan pengembangan sistem adaptif berbasis DSP dan aplikasinya untuk identifikasi sistem nyata (motor DC).

Berbagai penelitian tentang sistem adaptif sudah dilakukan. Widrow[6] mencapai pesat pencuplikan 0,2 Hz dan Tzou [5] mencapai 1kHz. Lin[2] mengimplementasikan secara hardware menghabiskan 35 *opamp* dan 2 ICL-8031. Seung [3] meneliti kendali adaptif dengan simulasi.

Salah satu implementasi dari sistem adaptif adalah struktur transversal yang menggunakan penggabung linier adaptif[7]. Sinyal keluaran sistem adaptif orde N untuk masukan dan keluaran tunggal adalah

$$y(k) = W_k^T X_k \quad (1)$$

dengan $W_k^T = [w_0(k) \ w_1(k) \ \dots \ w_{N-1}(k)]$ adalah vektor bobot, $X_k^T = [x(k) \ x(k-1) \ \dots \ x(k-N+1)]$ adalah vektor masukan, T menyatakan transpose, dan k adalah indeks waktu. Dalam sistem adaptif, bobot W_k selalu diperbaiki agar nilai $y(k) = d(k)$, dengan $d(k)$ adalah nilai yang diharapkan. Nilai $x(k)$ diperoleh dari pencuplikan.

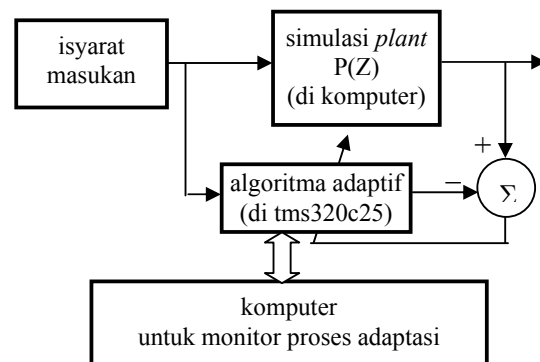
Perbaikan bobot menurut algoritma LMS[7] adalah:

$$W_{k+1} = W_k + 2\mu e(k)x(k) \quad (2)$$

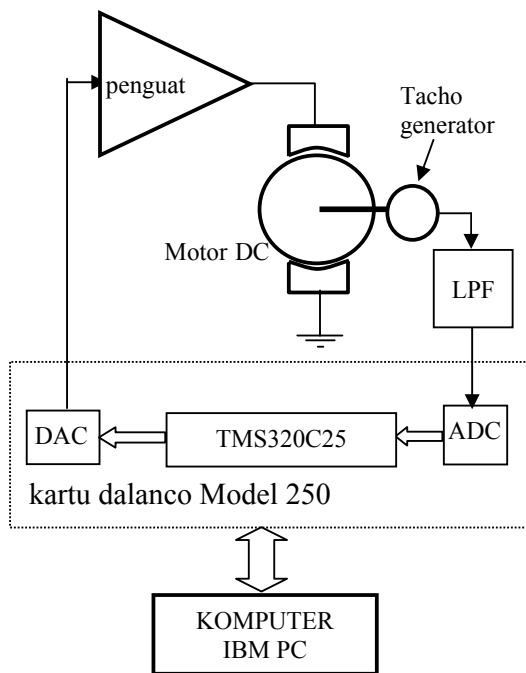
dengan μ = konstanta yang nilainya >0 , dan $e(k) = d(k) - y(k)$.

II. PENGEMBANGAN PROGRAM ADAPTIF

Alat (komputer dan kartu Dalanco[1]) disusun seperti pada gambar 1 untuk pengembangan dan pengujian program dan pada gambar 2 untuk implementasi sistem adaptif dengan sistem nyata.

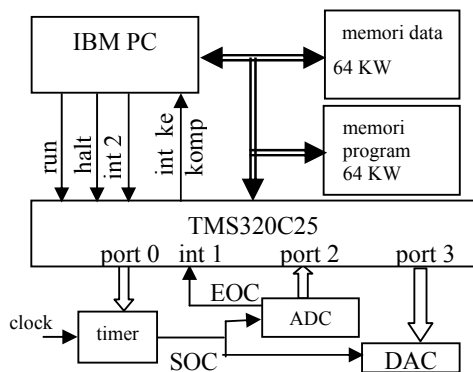


Gambar 1 Susunan alat untuk pengembangan dan pengujian program sistem adaptif.



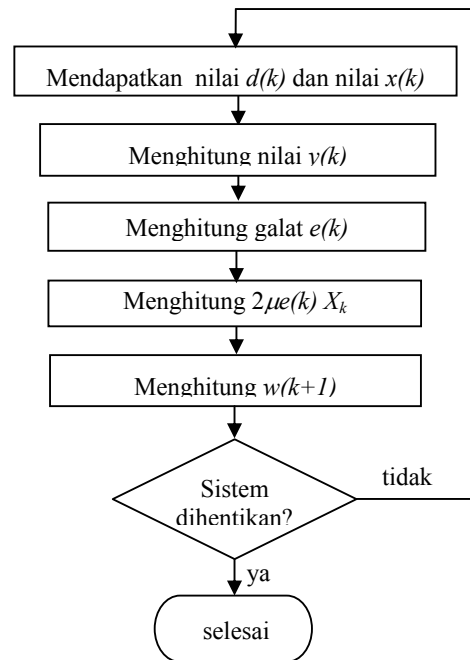
Gambar 2. Identifikasi motor DC

Kartu *Dalanco-Spry-model-250* [1] ditunjukkan dalam diagram kotak Gambar 3. TMS320C25 mempunyai lebar bus alamat dan bus data 16 bit sehingga semua port dan memori berukuran 16 bit.



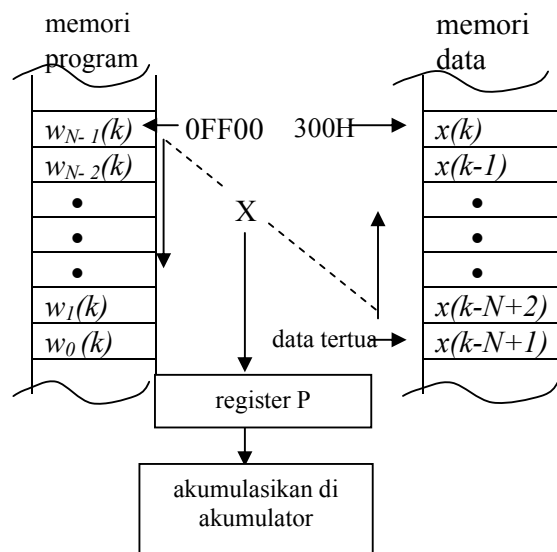
Gambar 3 Diagram kotak fungsional Kartu Dalanco model 250

Program adaptasi yang dikembangkan mempunyai diagram alir seperti gambar 4. Untuk test validasi sistem isyarat $d(k)$ dan $x(k)$ diperoleh dari hasil simulasi oleh komputer. Untuk sistem nyata $x(k)$ diperoleh dari ADC dan $d(k)$ dibangkitkan sendiri oleh program.



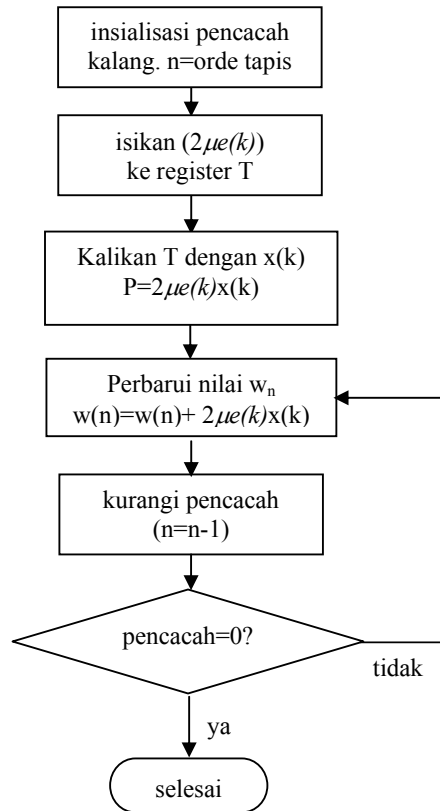
Gambar 4. Diagram alir program proses adaptasi

Instruksi MACD (Gambar 5) digunakan untuk menghitung $y(k)$. Instruksi ini bekerja cepat dalam menghitung perkalian dua vektor karena mengambil dua elemen vektor secara paralel (menggunakan bus data dan bus program) dan melakukan perkalian dan akumulasi secara paralel (pengali dan akumulator terpisah). Instruksi ini diulang N-kali untuk mengalikan titik dua buah vektor orde-N. Nilai awal bobot w adalah nilai random



Gambar 5. Ilustrasi instruksi MACD.

Galat $e(k)$ dihitung dengan penjumlahan skalar. Dalam program $2\mu = u$. Besaran $2\mu e(k)$ nilainya tetap selama perhitungan N buah bobot maka nilai $2\mu e(k)$ disimpan dalam register T, selanjutnya dikalikan dengan masing masing elemen vektor X_k untuk menghitung W_{k+1} (gambar 6).



Gambar 6. Diagram alir program proses menghitung w_{k+1}

Program adaptasi ditulis sebagai rutin untuk melayani interupsi-1 TMS320C25, dan hasilnya sebagai berikut.

Program sistem adaptif

```

; baca data dari komputer
lrlk 4,>300 ;addr blok-1(300H) di ar4
blkd >54f,u ;isi u dengan data dari komp
blkd >550,* ;kopikan x(k) ke >300
blkd >551,d ;d diisi keluaran aktual
; hitung y (estimasi nilai d)
lar3 3 ;ar3 sebagai pointer
cnfp ; B0 sbg memori prog
mpyk 0 ;nolkan reg P
lac satu,13 ;pembulatan*

```

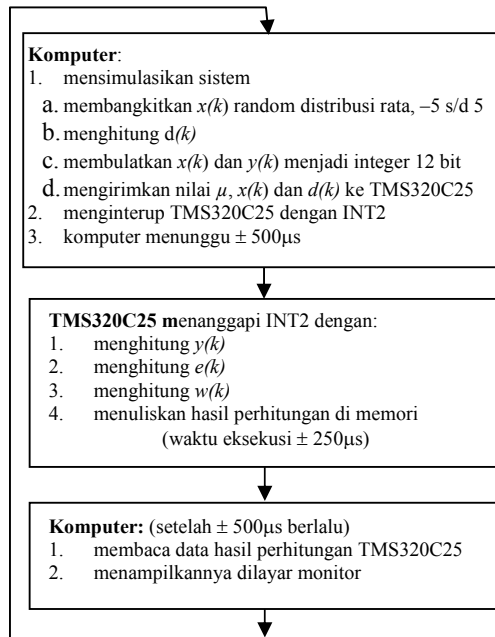
```

lrlk 3,>302 ;menunjuk sampel tertua (orde-2)
rptk >2 ;ulangi 2 kali (orde 2)
macd 0ff00h,*- ;estimasi d
cnfd ;B0 sbg memori data
apac ;akumulasi terakhir
sach y ;simpan y(n) (estimasi d)
;hitung galat
neg ;ACC = -y(n)
add d,14 ;ACC=ACC + d tergeser
sach gal,2 ;gal(n)=D(n)-y(n)
;perbaiki bobot
lt gal ;T=gal(n)
mpy u ;P=2\mu e(k)
pac ;ACC=P
add one,13 ;pembulatan ACC (2\mu e(k))
sach fgal,5 ;fgal=2\mu e(k)
lark 1,>2 ;set ar1 sbg counter=2
lrlk 2,>200 ;ar2 menunjuk bobot w di B0
lrlk 3,>303 ;ar3 menunjuk x tertua di B1
lt fgal ;T reg = 2\mu e(k)
mpy *-,2 ;P=T*x=2\mu e(k) x(k)
lagi:
zaln *-,3 ;ACC=w dengan pembulatan
mpya *-,2 ;ACC=ACC+P, (w=w+2\mu ex)
;P= 2\mu e(k) x(k-1)
sach *+,0,1 ;simpan w yang baru
banz lagi,*-,2 ;lagi bila belum nol, kurangi pencacah
;pindahkan hasil untuk dibaca oleh komputer
lar3 4 ;
lrlk 4,>553 ;
blkd y,*+ ;
blkd gal,*+ ;
blkd fgal,*+ ;
lrlk 4,>600 ;
rptk >2 ;
blkd >200,*+ ;
} Untuk dikirim ke monitor
;kembali ke program utama
eint
ret

```

Validasi program ini dilakukan dengan cara mengidentifikasi sistem yang sudah diketahui (Gambar 8) yang disimulasikan oleh komputer (Gambar 1). Kecepatan perhitungan adaptasi dihitung dengan menggunakan pencacah turun 16 bit dengan lonceng 1 MHz. Untuk satu iterasi sistem adaptif orde 6 diperlukan $250\mu s$. Diagram alir program untuk validasi dituangkan dalam gambar 7.

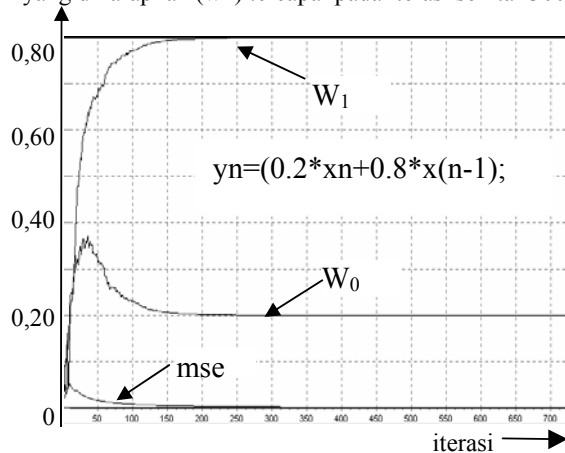
Besaran 2μ diubah dari 0.01 sampai 0,09. ADC akan mengubah nilai analog $x(k)$ menjadi bilangan integer 12 bit. Untuk pengolahan nilai 2μ , $d(k)$, $x(k)$, dan $y(k)$ digunakan bilangan *fixed point* 16 bit. Sistem adaptif dicoba untuk diaplikasikan pada sistem nyata seperti gambar 2.



Gambar 7. Diagram alir program untuk menguji program sistem adaptif

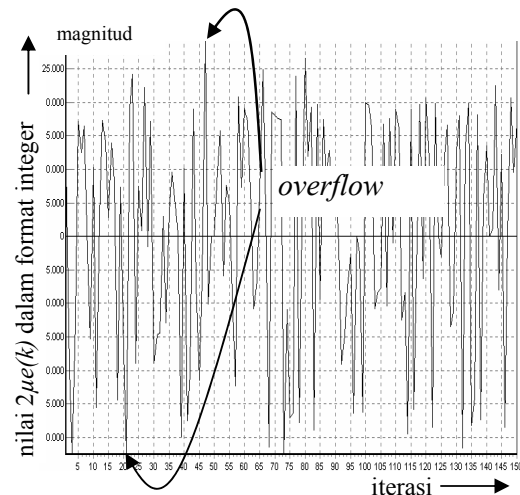
III. HASIL DAN PEMBAHASAN.

Gambar 8 memperlihatkan nilai w_0 dan w_1 untuk nilai $\mu=0,08$. Nilai awal w_0 dan w_1 adalah nol. Nilai yang diharapkan (w^*) tercapai pada iterasi sekitar 300.



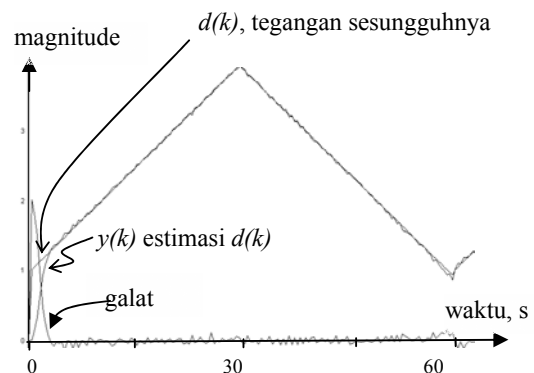
Gambar 8 adaptasi nilai w dengan nilai $\mu=0,08$. Konvergensi tercapai pada sekitar iterasi ke 160.

Kurva pembelajaran, mse, terlihat semakin mengecil dengan bertambahnya iterasi. Target nilai w_0 adalah 0,2 sedangkan w_1 adalah 0,8. Nilai w_0 terbaik yang dapat dicapai adalah 0,20007 sedangkan nilai w_1 terbaik yang dapat dicapai adalah 0,79974, kesalahan ini disebabkan oleh adanya pembulatan, pada kedua nilai ini galat sudah terlalu kecil untuk disajikan dengan bilangan bulat 16 bit sehingga tidak mampu memperbaiki bobot lagi.



Gambar 9. *Overflow* terjadi pada perhitungan $2\mu e(k)$ sehingga w_k tidak konvergen

Konvergensi dipercepat dengan memperbesar μ , namun untuk $\mu=0,09$ keatas perhitungan dapat menghasilkan nilai diatas 32767 atau dibawah -32768 sehingga terjadi *overflow* (gambar 9). Nilai μ haruslah dipilih kecil agar *overflow* tidak terjadi.

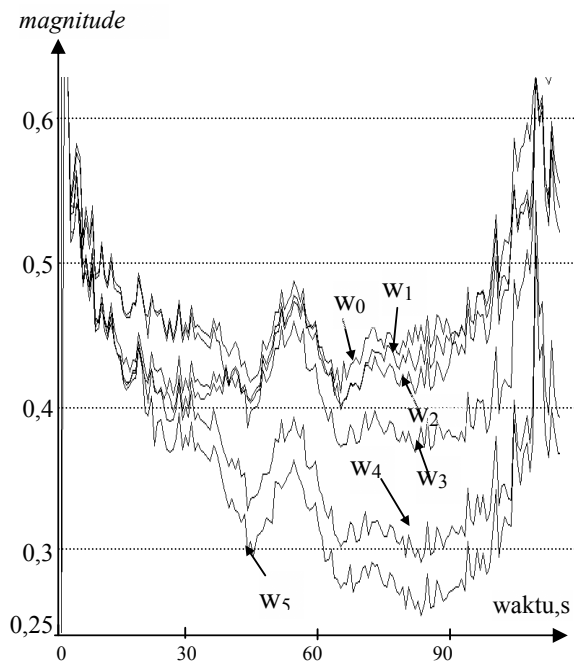


Gambar 10. Keluaran yang diharapkan $d(k)$, keluaran sistem adaptif $y(k)$ orde 6, dan error $e(k)$ pada aplikasi sistem adaptif untuk identifikasi invers sistem motor DC

Sistem adaptif ini sudah diaplikasikan untuk identifikasi invers sistem motor DC, yang terdiri atas ADC-DAC, penguat, motor-DC, dan LPF, seperti gambar 2. Sistem adaptif ini akan memodelkan setiap sistem dengan fungsi *transfer* orde- n sebagai berikut[7]:

$$P(z)=w_0z+w_1z^{-1}+w_2z^{-2}....+w_nz^{-n} \quad (3)$$

Invers sistem motor DC dengan demikian dimodelkan dengan polinomial semacam itu. Besaran yang diamati adalah nilai bobot $w(k)$, nilai $d(k)$, nilai $y(k)$ dan galat $e(k)$. Nilai $d(k)$, nilai $y(k)$ dan galat $e(k)$ terlihat pada Gambar 10. Nilai $w(k)$ terlihat pada gambar 11.



Gambar 11. Bobot w_0 sampai dengan w_5 sistem adaptif orde 6.

IV. KESIMPULAN DAN SARAN

Dari uraian diatas dapat ditarik kesimpulan bahwa pertama program sistem adaptif algoritma LMS telah berhasil dikembangkan dengan menggunakan Kartu Dalanco, kedua program membutuhkan waktu proses 250 μ s untuk sekali iterasi sistem adaptif orde 6, ketiga program dapat mengadaptasi dengan tepat sistem linear yang disimulasikan oleh komputer, dan keempat sistem sudah dapat digunakan untuk identifikasi waktu nyata, dan terakhir instruksi MACD

dan ZALR sangat berdaya guna untuk perhitungan yang cepat.

Saran

Penelitian yang berkelanjutan dengan modal dasar ini perlu terus dikembangkan antara lain untuk mengimplementasikan algoritma lain dan aplikasi sistem adaptif ini untuk masalah nyata dalam bidang pengendalian, prediksi, penghapusan interferensi, antenna larik adaptif dan sebagainya.

Daftar Pustaka

1. Dalanco Spry, 1990, Model 250 Data Acquisition and Signal Processing Board for IBM PC and ISA Bus Compatibles, New York
2. Lin F.J. Liaw C.M., 1993 *Reference Model Selection an Adaptive Control for Induction Motor Drives*, IEEE Transaction on Automatic Control, vol 38, No 10 Oct 1993, hal 1594-1599.
3. Seung M. B., Tae Y.K., 1997, An Adaptive PID Learning Control of DC Motors, IEEE International Conference on Systems, Man, and Cybernetics, jilid 3, hal 2877-2882.
4. Texas Instrument, 1989, Second Generation of TMS320 user's guide, Dallas.
5. Tzou Y. Y., 1996, *DSP Based Robust Control of an AC Induction Servo Drive for Motion Control*, IEEE Transaction on Automatic Control, vol 4, No 6 Nov 1996, hal 614-221.
6. Widrow B., Stearns S. D., 1985, Adaptive Signal Processing, Prentice hall, hal 274, New Jersey.

Daftar istilah

antarmuka interface
bobot weight
cuplik sample
galat error
invers inverse
isyarat signal
kartu card
keluaran output
lonceng clock
masukan input
pesat rate