

DESAIN KONTROL TRAKSI PADA MOTOR DC MENGUNAKAN FUZZY LOGIC BERBASIS FIELD PROGRAMABLE GATE ARRAY (FPGA)

Yuwono Marta Dinata¹⁾, Helmy Widyantara²⁾

^{1,2)} Dosen S1-Sistem Komputer STIKOM Surabaya, email : yuwono@stikom.edu, helmywid@stikom.edu

Abstract : Traksi control dalam dunia otomotif dapat sangat bermanfaat untuk mengatasi slip roda. Penerapan traksi control ini terkadang masih terkendala dengan desain *hardware* yang memiliki respon tinggi. Pada penelitian ini menggunakan *Field Programmable Gate Array* (FPGA), merupakan untaian yang dapat dibuat dengan metode *schematic* maupun dengan bahasa pemrograman. Dengan melakukan pengujian set point untuk memberikan masukkan berapa besar nilai RPS yang dikehendaki sistem untuk tuning kecepatan motor. Hasil yang didapatkan bahwa penggunaan rangkaian *schematic* yang terdapat pada FPGA dapat optimal kira-kira 35% dari kapasitas maksimalnya.

Keyword : *FPGA, traksi, fuzzy*

Teknologi kendali traksi merupakan rekayasa putaran roda yang terkomputerisasi untuk mengatasi mobil kala mengalami slip, spin, maupun guncangan akibat buruknya permukaan yang bersentuhan dengan roda, sehingga dapat dikatakan sistem kendali traksi dikembangkan sebagai penjaga stabilitas arah laju.

Traksi pada roda adalah gaya dorong/tarik yang dapat dihasilkan roda ketika sebuah torsi 'T' bekerja pada roda tersebut, Sudarsono (2003). Sistem kendali traksi merupakan sistem yang mampu mempertahankan *ratio slip* di antara roda dan permukaan lintasan dengan cara mengendalikan peralatan-peralatan pendukung sehingga kecepatan perputaran roda tersebut selaras antara *grip* roda dengan permukaan lintasan.

Penerapan sistem kendali traksi pada saat ini masih terbatas pada aplikasi elektronik untuk mendukung aplikasi *mechanical engineering*, belum banyak yang mengarah pada pengembangan elektronik untuk mendukung *electrical engineering*.

Pada masa mendatang diperkirakan semua mobil yang melintas di jalan raya adalah mobil listrik. Mobil listrik menggunakan baterai sebagai sumber tenaganya, sedangkan untuk mengubah energi listrik menjadi energi kinetik dapat menggunakan Motor DC. Untuk mengendalikan mobil listrik DC agar memiliki kenyamanan sebagaimana mobil konvensional maka diperlukan sistem kendali traksi pada Motor DC.

Penerapan sistem kendali traksi pada Motor DC memerlukan *hardware* pendukung yang bagus, *hardware* tersebut harus memiliki kemampuan untuk merespon Input dan Output dengan cepat, memiliki jumlah Input dan Output yang banyak, membutuhkan sumber tegangan yang rendah, memiliki daya tahan yang tinggi, dan proses implementasinya mudah.

Tuntutan akan kecepatan proses dan unjuk kerja pengendali yang handal mendorong untuk mengimplementasikan sistem kendali traksi pada motor DC dalam bentuk untaian logika perangkat keras (*hardware logic*). Operasi dalam bentuk perangkat keras mempunyai kecepatan lebih tinggi dibandingkan dengan operasi yang dilakukan secara perangkat lunak oleh mikrokontroler, hal ini dikarenakan operasi dengan perangkat lunak membutuhkan waktu untuk menerjemahkan perintah-perintah program.

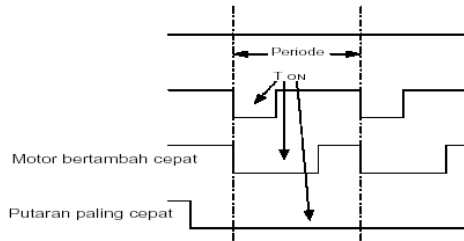
Implementasi operasi-operasi logika digital dalam bentuk perangkat keras dapat dilakukan dengan *Field Programmable Gate Array* (FPGA). FPGA di dalamnya memuat ribuan gerbang logika yang dapat diprogram untuk membentuk suatu operasi logika. FPGA dapat digunakan untuk mengimplementasikan sistem kombinasional dan sekuensial berkecepatan tinggi dengan lebar jalur data yang tidak terbatas. Hal ini membuat FPGA mampu melakukan operasi dengan tingkat keparalelan tinggi yang tidak mungkin dilakukan oleh mikrokontroler.

Konsep teknologi untaian terintegrasi (*integrated circuit*) yang dapat diprogram dan dihapus dengan konsep FPGA telah dikembangkan oleh Xilinx. Metode yang digunakan untuk mengimplementasikan operasi logika dengan FPGA yaitu dengan metode *Schematic* (berupa gambar untaian elektronik digital dari perangkat lunak) maupun dengan bahasa pemrograman *Very High Speed Hardware Description Language* (VHDL).

Pulse Width Modulation (PWM)

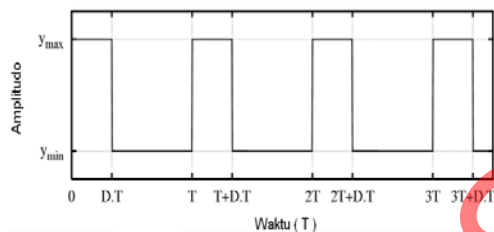
Pulse Width Modulation (PWM) adalah suatu teknik manipulasi dalam pengendalian kecepatan motor dengan menggunakan prinsip *cut-off* dan saturasi.

Driver penggerak motor didesain bekerja dengan karakteristik mirip 'linier' namun sebenarnya menggunakan teknik *on-off* (http://en.wikipedia.org/wiki/Pulse-width_modulation#Principle). Gambar 1 mengilustrasikan sinyal PWM.



Gambar 1 Signal PWM

Pada prinsipnya PWM menggunakan signal kotak yang dimodulasikan dan diatur lebar datanya, dengan frekuensi $f(t)$ untuk nilai rendah y_{min} dan nilai tinggi y_{max} dan dengan *duty cycle* D , dapat kita tampilkan korelasi bentuk signalnya dalam Gambar 2.



Gambar 2. Signal terhadap waktu

Signal dari Gambar 2, dapat kita tuliskan persamaan matematisnya sebagai berikut:

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt \quad (1)$$

Dimana : $f(t)$ adalah signal kotak, dengan nilai y_{max} $0 < t < D.T$ dan y_{min} untuk $D.T < t < T$ sehingga dapat diekspresikan

$$\begin{aligned} \bar{y} &= \frac{1}{T} \left(\int_0^{D.T} Y_{max} dt + \int_{D.T}^T Y_{min} dt \right) \quad (2) \\ &= \frac{D.T.Y_{max} + T(1-D)Y_{min}}{T} \\ &= D.Y_{max} + (1-D)Y_{min} \end{aligned}$$

untuk selanjutnya persamaan diatas apabila $Y_{min} = 0$, nilai $\bar{y} = D.Y_{max}$ nilai rata rata dari signal (\bar{y}) secara langsung akan tergantung pada *duty cycle* (D). (http://en.wikipedia.org/wiki/Pulse-width_modulation#Principle)

Apabila dikehendaki kecepatan motor penuh maka pada jalur *enable* pada driver motor diberi lebar pulsa *high* secara konstan. Sebaliknya jika dikehendaki kecepatan motor rendah maka pada jalur *enable* diberi lebar pulsa *low* lebih banyak dari pada pulsa *high*. Dengan mengatur prosentase lebar pulsa *high* dan *low* maka dapat diperoleh kecepatan motor

yang bervariasi. pembangkitan pulsa PWM dapat diperoleh melalui *multivibrator*, rangkaian *timer* yang terdapat pada mikrokontroler atau osilator yang terdapat pada FPGA.

Fuzzy Logic

Dalam langkah ini, dilakukan *fuzzyfikasi* kecepatan motor DC dengan *set point* kecepatan yang diinginkan. Dalam hal kontrol kecepatan rotasi dari sebuah motor yang universal, nilai inputnya diperoleh dari :

1. kesalahan mutlak di dalam kecepatan putaran
 $Error = Setpoint - CurrentSpeed$

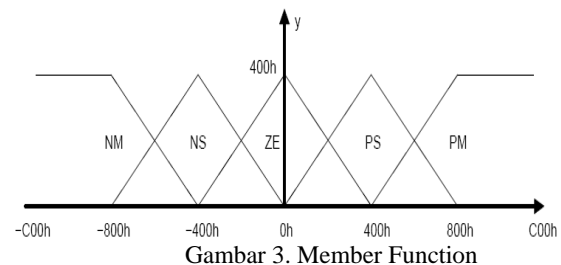
2. kesalahan kecepatan *differential* rotasi nilai ini didapatkan dari pengurangan error terdahulu dari nilai nilai error yang sekarang.

$$dError = Error - LastError$$

Misalnya terdapat lima set *fuzzy* yang didefinisikan untuk nilai input Error dan dError :

1. NM: negative medium
2. NS: negative small
3. ZE: zero equal
4. PS: positive small
5. PM: positive medium

*Membership function*nya berupa bentuk segitiga (Dannenberga., Courtesy of: Dr. Odry Peter: *Fuzzy Logic Motor Control with MSP430x14x*) seperti pada Gambar 3 dan nilai maksimumnya diskalakan sampai 400h. Untuk membuat sederhana proses perhitungannya maka operasi yang digunakan adalah penambahan atau pengurangan.

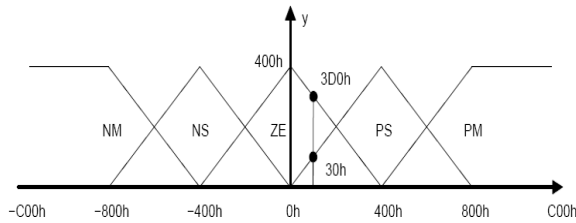


Gambar 3. Member Function

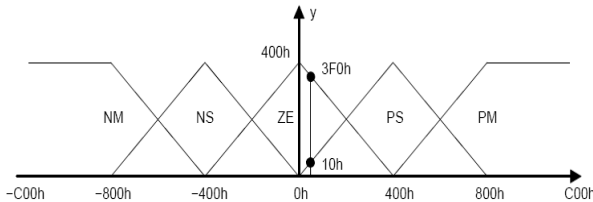
Hasil dari fuzzifikasi dari sebuah nilai input yaitu berupa vektor dengan 5 element membership function, dan nilai dari setiap anggota memberikan definisi keanggotaan nilai input untuk fuzzy tertentu (nilai - y).

Vektor untuk kesalahan mutlak dan *differential* yang merupakan hasil dari fuzzifikasi disimbolkan dengan $X1[]$ dan $X2[]$. Contoh : misal Error 30h dan dError 10h, sehingga berdasarkan Gambar 4 dan Gambar 5, diperoleh hasil sebagai berikut :

$$\begin{aligned} X1[] &= [0h, 0h, 3D0h, 30h, 0h] \\ X2[] &= [0h, 0h, 3F0h, 10h, 0h] \end{aligned}$$



Gambar 4. Membership Function For Error (X1[])



Gambar 5. Membership Function For Error (X2[])

Fuzzy Inference

Hubungan max-min Zadeh digunakan karena dari hasil percobaan membuktikan bahwa percobaan yang dilakukan memberikan hasil yang baik. Prosesnya terdiri dari :

1. Hapus vektor output Y[].
2. Ambil anggota pertama dari vektor X1[].
3. Bandingkan dengan setiap anggota vektor X2[]. Nilai yang lebih kecil di rekam pada setiap kali membandingkan.
4. Nilai maksimum ditemukan diantara anggota yang di dapat pada langkah ke 3.
5. Nilai maksimum ini yang kemudia ditambahkan ke output member vektor yang didefinisikan oleh aturan inference.
6. Anggota berikutnya dari vektor X1[] diambil dan proses dari 3 sampai 6 di ulang. Aturan inference diperlihatkan pada table 1.

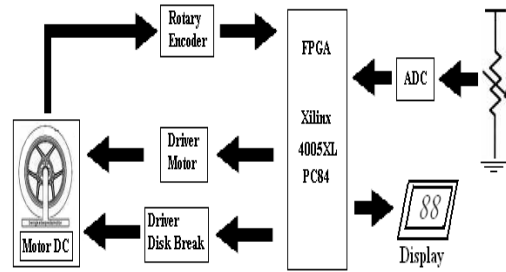
Table 1. Fuzzy Inference Rule Table

		dError (X2[])				
		NM	NS	ZE	PS	PM
Error (X1[])	NM	PM	PM	PM	PS	ZE
	NS	PM	PM	PS	ZE	NS
	ZE	PM	PS	ZE	NS	NM
	PS	PS	ZE	NS	NM	NM
	PM	ZE	NS	NM	NM	NM

Pada contoh yang diberikan output vektor Y[] adalah: Y[] = [0h, 30h, 3D0h, 0h, 0h]

METODE

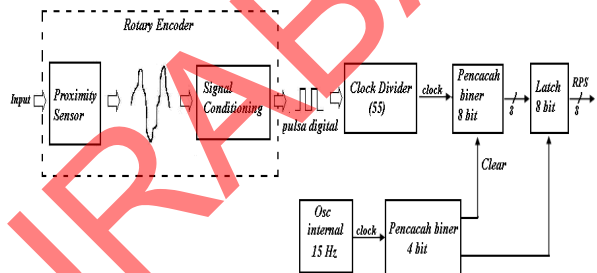
Keseluruhan sistem yang dibuat pada penelitian ini sesuai dengan blok diagram pada Gambar 6.



Gambar 6 Blok Diagram Sistem Secara Keseluruhan

Perancangan Perangkat Keras

Perancangan perangkat keras meliputi pembacaan kecepatan putar roda dalam *rotate per second* (RPS), dan sistem kendali motor. Diagram kotak dari pembacaan kecepatan putar roda dapat dilihat pada Gambar 7.



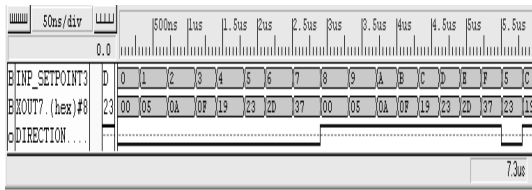
Gambar 7 Pembacaan RPS

Keluaran dari *proximity sensor* masih dalam bentuk signal analog, oleh karena itu perlu dikondisikan menjadi pulsa digital. Pulsa digital dari *rotary encoder* ini perlu dicacah melalui *clock divider* (55), hal ini dikarenakan dalam satu putaran penuh roda terdapat lima puluh lima lubang yang dideteksi oleh sensor *proximity*.

Keluaran dari *clock divider* selanjutnya akan dihitung oleh pencacah biner delapan bit. Nilai dari pencacah delapan bit ini akan di *latch* (ditahan) dan di *reset* sesuai dengan *time sampling* yang besarnya satu detik. Data RPS dari pencacah delapan bit selanjutnya bersama dengan data *setpoint* akan dimasukkan sebagai input system fuzzy untuk menentukan besar kecil dari PWM.

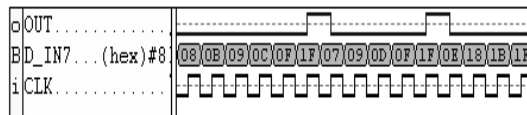
Pengujian Sistem

Set point berfungsi untuk memberikan masukan berapa besar nilai RPS yang dikehendaki oleh sistem. Hasil simulasi dari sub unit ini dapat dilihat pada Gambar 8, dari hasil simulasi tersebut ditampilkan bahwa unit *set point* bekerja sesuai dengan apa yang diharapkan. Masukan data dari ADC dapat dibaca dan dikonversikan kedalam RPS dalam bentuk heksa desimal.



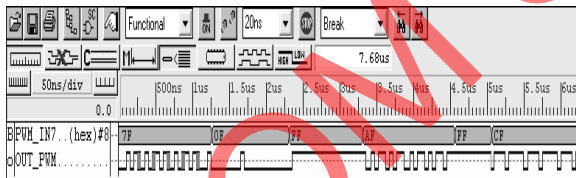
Gambar 8 Hasil simulasi *set point*

Pendeteksi *slip* berfungsi untuk menjaga agar sistem berjalan secara normal. Apabila terjadi lonjakan yang sangat besar (antara t dan t_1) maka keluaran dari pendeteksi *slip* akan berlogika "1". Hasil simulasi dari sub unit pendeteksi *slip* dapat dilihat pada Gambar 9.



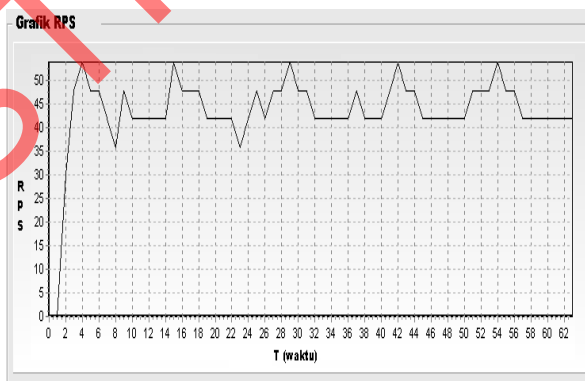
Gambar 9 Hasil simulasi pendeteksi *slip*

Sub unit PWM berfungsi untuk mengatur kecepatan RPS Motor DC. Nilai PWM yang digunakan mulai dari 0 sampai dengan 255 desimal, atau 00 sampai dengan FF heksa. Semakin besar nilai PWM yang diberikan maka akan berakibat semakin besar lebar pulsa berlogika "1". Pada penelitian ini sistem kendali kecepatan Motor DC dirancang apabila semakin besar lebar pulsa berlogika "1" maka akan menyebabkan semakin cepat Motor DC berputar. Hasil simulasi dari sub unit PWM dapat dilihat pada Gambar 10.



Gambar 10 Hasil simulasi sub unit PWM

Hasil yang diperoleh dari perekaman RPS disajikan dalam grafik pada Gambar 11. Pada grafik terlihat apabila terjadi *ofer shut* sistem akan segera dibawa kembali pada keadaan yang diharapkan.



Gambar 11 Grafik RPS

SIMPULAN

Dari hasil pengujian dan analisis yang telah dilakukan pada sistem kendali traksi pada Motor DC ini, maka dapat ditarik kesimpulan sebagai berikut :

1. Sistem kendali traksi pada Motor DC dapat dirancang dan diimplementasikan secara perangkat keras menggunakan FPGA Xilinx XC4005XL.
2. Berdasarkan file laporan (file *report*) jumlah CLB yang terpakai adalah 42 atau 21% dari total jumlah CLB yang terdapat pada XC4005XL-PC84 dan jumlah pin I/O (IOBs) sebanyak 23 buah atau 35% dari kapasitas maksimalnya, hal ini mengindikasikan bahwa penggunaan gerbang logika telah dirancang secara optimal.

RUJUKAN

Siahaan, I.H., & Sutantara, I.N. 2005. Jurnal Teknik Mesin Vol.7 No.1, April 2005: *Simulasi Model Matematis Kontrol Sistem Traksi*, (Online), (URL: <http://puslit.petra.ac.id/~puslit/journals/articles.php?PublishedID=MES05070106>, diakses 11Maret 2008).

Bejo, A, 2004, *Kendali Kecepatan Motor Induksi Satu Fase Dengan Inverter PWM Berbasis Mikrokontroler dan FPGA*, UGM, Yogyakarta.

Sutikno, T. 2004. *Pembangkit Sinyal PWM Sinusoida Dua Fasa Berbasis FPGA*, UGM, Yogyakarta.

Sudarsono, B., & Dewanto, J.2003. Jurnal Teknik Mesin Vol.5 No.2, Oktober 2003: *Pemodelan Sistem Gaya dan Traksi Roda*, (Online), (URL: <http://www.petra.ac.id/~puslit/journals/pdf.php?PublishedID=MES03050204>, diakses 27 November 2008).

Kusumawardani, S. 2001. *Implementasi Sandi BCH (15,5) dengan FPGA XC4013*, UGM, Yogyakarta.

Gendroyono, P. 1999. *Sistem Penggerak Motor Induksi Dengan Beban Berubah Menggunakan Inverter PWM Berbasis Mikrokontroler*, UGM, Yogyakarta.

Xilinx Corporation. 1999. *The Programmable Logic Data Book.2100 Logic Drive*. San Jose. California 95124. United States America.

Chan, P.K., & Mourad, S. 1994, *Digital Design Using Field Programmable Gate Arrays*,

Prentice-Hall, New Jersey. United States
America.

[http://en.wikipedia.org/wiki/Pulse-
width_modulation#Principle](http://en.wikipedia.org/wiki/Pulse-width_modulation#Principle)

Dannenber A., Courtesy of: Dr. Odry Peter, Diveki Szabolcs, Csasznyi Andor, Burany Nandor, Februari 2005., Fuzzy Logic Motor Control with MSP430x14x, Application report SLA235, Texas Instrument.

STIKOM SURABAYA