

MODUL PENGATURAN KECEPATAN MOTOR SATU PHASA DENGAN PID BEBRBASIS MIKROKONTROLER MENGGUNAKAN INTERFACE LABVIEW

Ratih Putri Nastiti

D3 Teknik Elektro, FTI, ITS.

Wahyu Susanto

D3 Teknik Elektro, FTI, ITS.

Motor induksi merupakan motor listrik arus bolak-balik yang banyak penggunaannya. Kelemahan motor induksi saat terjadi perubahan beban tidak mampu mempertahankan kecepatannya. Pada Tugas Akhir ini diuraikan sistem pengaturan kecepatan motor induksi satu fasa menggunakan metode PID untuk menghasilkan kecepatan motor sesuai dengan *setpoint* saat motor tanpa beban atau berbeban. Pengaturan kecepatan ini dikendalikan dengan tegangan jala-jala 220VAC diubah menjadi tegangan DC diubah kembali menjadi tegangan AC dengan frekuensi yang sesuai menggunakan metode *switching* yang dikendalikan oleh mikrokontroler. Dengan adanya sistem ini, motor dengan sistem *open loop* menggunakan *setpoint* 2500 rpm tanpa beban didapatkan nilai *rise time* 30 detik. Ketika motor diberi beban, kecepatan motor turun hingga motor berhenti. Sedangkan motor pada sistem *close loop* dengan PID menggunakan *setpoint* yang sama didapatkan nilai *rise time* 1,2 detik. Kecepatan motor turun ketika diberi beban, akan tetapi dapat kembali ke nilai *setpoint*-nya dengan nilai *rise time* 4 detik. Hal ini dikarenakan pada sistem *open loop* tidak terdapat umpan balik sedangkan pada sistem *close loop* dengan PID terdapat umpan balik berupa nilai kecepatan motor.

Kata Kunci : PID, LabVIEW, Rise Time

I. PENDAHULUAN

Motor induksi satu fasa digunakan secara luas pada berbagai aplikasi industri kecil dan rumah tangga karena harga yang murah, handal dan kuat. Sumber tegangan tiga fasa umumnya tidak tersedia pada pelanggan rumah tangga, oleh karena itu motor induksi satu fasa merupakan pilihan utama karena dapat langsung dihubungkan dengan sumber tegangan AC.

Kecepatan putaran motor induksi satu fasa dipengaruhi oleh frekuensi, jumlah kutub, tegangan jala-jala dan beban mekanisnya. Pada keadaan tanpa beban, kecepatan putaran motor induksi mendekati kecepatan medan magnet putar (kecepatan sinkron). Bila motor dibebani, kecepatan putaran motor berkurang akibat adanya torka beban yang menyebabkan slip pada motor induksi.

Dalam pengoperasian motor induksi satu fasa, kecepatan putaran motor tidak konstan. Kecepatan maksimum motor terjadi ketika motor tidak dibebani. Kecepatan motor jauh berkurang karena adanya beban yang harus dipikul oleh motor. Sedangkan, pada sebuah industri diperlukan kecepatan putaran mesin yang dapat diatur dengan tetap mempertahankan kestabilan putaran. Hal ini mengindikasikan bahwa kecepatan putaran yang diperlukan adalah kecepatan dimana motor sedang mendapat beban penuh. Pengaturan tegangan yang masuk ke motor dapat mengubah kecepatan motor mengikuti nilai tegangan yang disalurkan pada motor AC satu fasa yang digunakan.

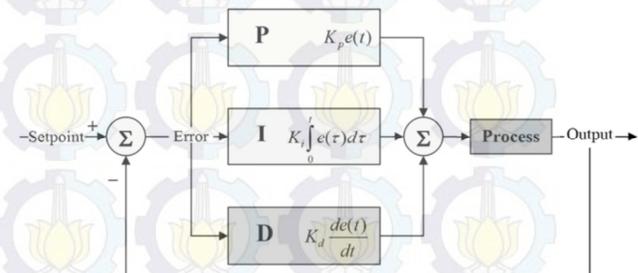
II. LANDASAN TEORI

2.1 Kontrol PID

Proporsional-integral-derivatif (PID) sering disebut sebagai kontroler 'jangka tiga'. Saat ini salah satu pengendali yang paling sering digunakan dalam industri

proses. Dalam kontroler PID, variabel kontrol yang dihasilkan dari jangka waktu proporsional untuk *error*, jangka waktu yang mana merupakan integral dari *error*, dan jangka waktu yang mana merupakan turunan dari *error*.

Salah satu tugas pengendali PID adalah mereduksi sinyal kesalahan, yaitu perbedaan antara sinyal setting dan sinyal aktual. Semakin cepat respon pengendali dan semakin kecil nilai kesalahan yang didapat maka semakin baik pula pengendali PID tersebut. Blok diagram pengendalai PID diperlihatkan pada Gambar 2.1 dimana setiap persamaan dari P, I, D saling berhubungan untuk mendapatkan *output* yang diinginkan.



Gambar 2.1 Blok Diagram Kontrol PID

2.2 Persamaan PID

Gabungan dari proposional, integral, dan derivatif dapat dinyatakan pada Gambar 2.2 yang menunjukkan diagram blok klasik waktu kontinu kontroler PID.

Menyetel kontroler meliputi penyesuaian parameter K_p , K_d dan K_i untuk mendapatkan respon yang memuaskan. Karakteristik kontroler PID diketahui dengan baik dan benar, dan sebagian besar pengendali modern berdasarkan pada beberapa bentuk PID. Hubungan *input-output* kontroler PID dapat dinyatakan sebagai berikut :

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right]$$

dengan,

$u(t)$ = sinyal *output* dari kontroler

K_p = Konstanta Proporsional

T_i = waktu integral

T_d = waktu derivatif

$e(t)$ = sinyal *error*

Perhatikan bahwa Persamaan diatas kadang-kadang ditulis sebagai

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} + u_0,$$

dimana,

$$K_i = \frac{K_p}{T_i} \text{ dan } K_d = K_p T_d$$

Setelah mendapatkan persamaan PID di atas maka untuk memasukkan parameter PID ke dalam program menggunakan persamaan PID kontrol seperti Gambar 2.2. Sehingga, didapatkan parameter sebagai berikut:

$$\alpha = K_p T / T_i$$

$$\beta = K_p T_D / T$$

Dengan mengacu pada Gambar 2.2 dapat ditulis persamaan perbedaan berikut:

untuk bagian proporsional,

$$w_k = K_p e_k$$

untuk bagian integral,

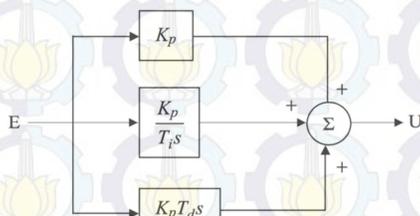
$$p_k = \alpha e_k + p_{k-1}$$

dan untuk bagian derivatif,

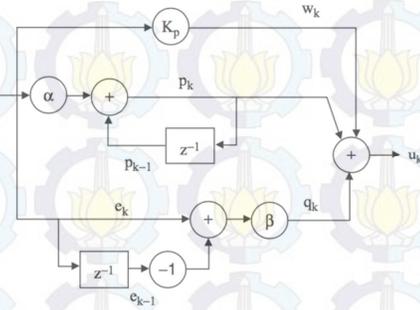
$$q_k = \beta (e_k - e_{k-1})$$

Maka, keluaran yang dihasilkan sebagai Persamaan berikut :

$$u_k = w_k + p_k + q_k$$



Gambar 2.2 Continuous-time Sistem Kontrol PID



Gambar 2.3 Kontrol PID Dalam Struktur Paralel

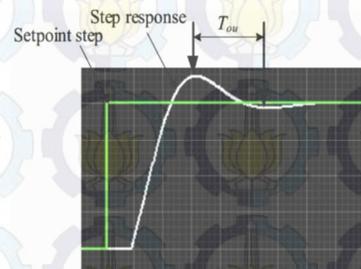
2.3 Metode Good Gain

Metode *good gain* merupakan salah satu cara untuk mendapatkan parameter PID yang sesuai dengan sistem. Metode ini berbeda dengan metode Ziegler Nichols dimana pada metode ini tidak mengatur sistem kontrol sampai pada keadaan beresilasi. Adapun prosedur yang harus dilakukan dalam metode ini untuk menentukan parameter PID yaitu sebagai berikut:

1. Proses pertama dilakukan pada sistem *loop* tertutup dimana sinyal kontrol dimasukkan secara manual.
2. Kontrol proporsional dimasukan secara bertahap dimulai dari 0 atau 1 ($K_p = 0$), dan untuk waktu integral dan derivatif diset ∞ dan 0 (set $T_i = \infty$ dan $T_d = 0$). Penambahan atau pengurangan nilai K_p berpengaruh terhadap respon *overshoot* suatu sistem.
3. Set waktu integral T_i sesuai Persamaan berikut :

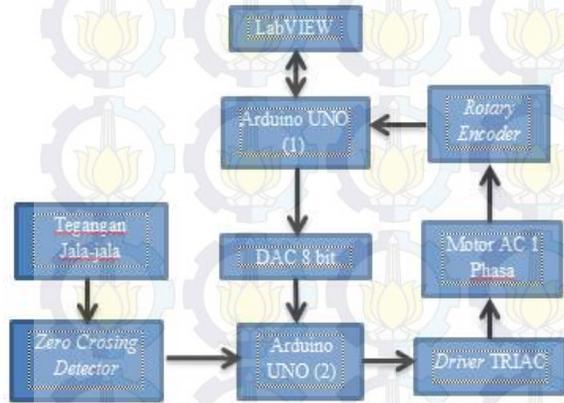
$$T_i = 1,5 T_{ou}$$
 Dimana pada Gambar 2.4 menunjukkan T_{ou} adalah waktu respon suatu sistem setelah mengalami *overshoot* menuju *steady state*.
4. Cek kesetabilan sistem kontrol setelah digunakan langkah *setpoint* karena ini merupakan permulaan untuk parameter integral. Parameter integral sangat mempengaruhi parameter proporsional. Jika pada kontrol PI kestabilan sistem menjadi buruk maka langkah yang harus dilakukan adalah mengurangi K_p 80% dari harga aslinya.
5. Jika ingin memasukkan parameter derivatif agar kontroler menjadi PID kontrol, maka *setting* sesuai dengan Persamaan berikut

$$T_d = \frac{T_i}{4}$$



Gambar 2.4 Respon T_{ou}

III. PERANCANGAN SISTEM



Gambar 3.1 Blok Fungsional Sistem

Tahapan-tahapan yang ada pada blok fungsional untuk perancangan Tugas Akhir ini sebagai berikut :

1. LabVIEW : digunakan untuk mengirim parameter-parameter pengaturan seperti *setpoint*, *kp*, *ki*, dan *kd*. Selain itu juga menerima data berupa nilai kecepatan motor
2. Arduino UNO (1) : digunakan untuk menerima parameter yang dikirim dari LabVIEW, menerima sinyal pulsa dari *rotary encoder* untuk diubah menjadi nilai kecepatan, mengirim data nilai kecepatan ke LabVIEW, dan mengirim *output* kecepatan berupa data digital ke DAC 8 bit.
3. DAC 8 bit: digunakan untuk mengubah data digital yang dikirim oleh Arduino UNO (1) menjadi data analog untuk dikirim ke Arduino UNO (2)
4. *Zero Crossing Detector*: digunakan untuk mendeteksi persilangan nol tegangan jala-jala PLN
5. Arduino UNO (2) : digunakan untuk menerima data analog DAC dan melakukan *switching driver*
6. *Driver TRIAC* : digunakan untuk mengatur kecepatan motor dengan metode *switching*
7. *Rotary encoder* : digunakan untuk membaca kecepatan motor dan mengeluarkannya dalam bentuk sinyal pulsa
8. Motor AC 1 fasa : *plant* yang akan dikontrol
9. Tegangan Jala-jala : sebagai *input zero crossing detector*

IV. HASIL DAN ANALISA

4.1 Pengujian Respon Motor terhadap *Rise Time*

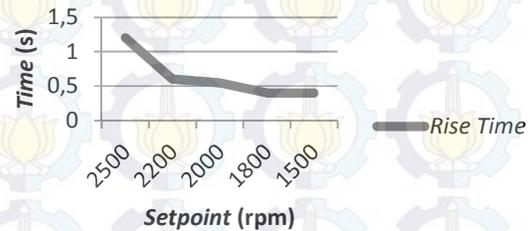
Pengujian ini dilakukan untuk melihat respon motor dalam keadaan tanpa beban mencapai nilai kecepatan referensi atau *setpoint*. Berikut Tabel 4.1 hasil pengujian nilai *rise time* respon motor tanpa beban sebagai berikut :

Tabel 4.1 Pengujian *Rise Time* Respon Motor Tanpa Beban

Kecepatan (rpm)	<i>Rise Time</i> (s)	<i>Settling Time</i> (s)
2500	1,2	4
2200	0,6	9

2000	0,55	9
1800	0,4	12
1500	0,4	15

Tabel 4.1 menunjukkan bahwa semakin kecil nilai *setpoint* kecepatan motor maka semakin cepat respon motor mencapai nilai *rise time*. Karena, sistem pengendali lebih cepat mengatur kecepatan pada kecepatan referensi yang lebih rendah daripada kecepatan referensi yang lebih tinggi. Gambar 4.16 menunjukkan perbandingan antara kecepatan referensi motor dengan *rise time*. Semakin besar nilai kecepatan referensi maka semakin lama waktu *rise time* motor. Hal ini dikarenakan, pada sistem kendali memerlukan waktu untuk mengatur motor pada kecepatan referensinya.



Gambar 4.1 Grafik Perbandingan *Rise Time* Terhadap *Setpoint*

4.2 Pengujian Respon Motor Berbeban

Pengujian respon motor berbeban ini dilakukan dengan memberi beban mekanik pada motor sehingga dapat diketahui respon motor untuk mencapai kembali kecepatan referensi atau *setpoint*. Dimana beban naik ialah hambatan mekanik yang diberikan saat motor berputar, sedangkan beban turun ialah saat hambatan mekanik dilepas. Berikut Tabel 4.2 hasil pengujian *steady state* motor berbeban sebagai berikut:

Tabel 4.2 Pengujian *Steady State* Berbeban

Kecepatan (rpm)	<i>Settling Time</i> (s)	
	Beban naik	Beban turun
2500	4	4
2200	4	4
2000	3	4
1800	4	3,9
1500	5	4,9

Tabel 4.2 menunjukkan bahwa pada nilai *steady state* akan berbeda pada setiap perubahan kecepatan. Hal ini, dikarenakan alat pengendali membutuhkan waktu untuk menghitung dan mengembalikan putaran pada kecepatan referensi saat adanya hambatan atau beban Gambar 4.2 menunjukkan grafik perbandingan *steady state* pada saat beban naik dengan beban turun disetiap perubahan kecepatannya. Terjadi perbedaan waktu $\pm 0,1$ detik saat beban naik maupun beban turun untuk mencapai nilai *steady state*.



Gambar 4.2 Grafik Perbandingan *Settling Time* Saat Beban Naik dan Turun

V. PENUTUP

A. Kesimpulan

1. Motor dengan sistem *open loop* menggunakan *setpoint* 2500 rpm tanpa beban didapatkan nilai *rise time* 30 detik. Sedangkan motor pada sistem *close loop* dengan PID menggunakan *setpoint* yang sama didapatkan nilai *rise time* 1,2 detik.
2. Ketika motor dengan sistem *open loop* menggunakan *setpoint* 2500 rpm diberikan beban, kecepatan motor turun hingga motor berhenti. Sedangkan motor pada sistem *close loop* dengan PID menggunakan *setpoint* yang sama, kecepatan motor juga turun akan tetapi dapat kembali ke nilai *setpoint*-nya dengan nilai *rise time* 4 detik.
3. Perbedaan antara sistem *open loop* dengan sistem *close loop* PID adalah pada sistem *open loop* tidak terdapat umpan balik sedangkan pada sistem *close loop* dengan PID terdapat umpan balik berupa nilai kecepatan motor.

B. Saran

Saran untuk penelitian selanjutnya adalah untuk mendapatkan sistem pengaturan kecepatan yang lebih stabil dan mudah diperlukan perbandingan dengan metode yang berbeda.

DAFTAR PUSTAKA

- [1] Sinaga, Ridwan. “*Analisis Karakteristik Berbeban Motor Induksi Satu Fasa Kapasitor Start*”, Tugas Akhir. Universitas Sumatera Utara, 2011
- [2] Sigiro, Stevani Agnesia. (2015). “*Instrumentasi Virtual Menggunakan Labview Dan Soundcard*”, Tugas Akhir. Universitas Sumatera Utara, 2015
- [3] Wisesa, Tirta. (2014). “*Perancangan Pengaturan Kecepatan Motor Induksi Satu Fasa dengan PWM Menggunakan Pengendali PID Berbasis Arduino*”, Tugas Akhir. Universitas Bengkulu, 2014.
- [4] Wirawan, Giri Nata. (2015). “*Sistem Instrumentasi Elektronika*.” <http://dokumen.tips/documents/zero-crossing.html>[10 Mei 2016]