PERANCANGAN PI KONTROLER PADA KONTROL KECEPATAN MOTOR DC DENGAN KOMBINASI *POLE PLACEMENT* DAN *SYMMETRICAL OPTIMUM*

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

CHRISTOPHER IMANTAKA
NIM. 105060300111038-63

UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2016

LEMBAR PENGESAHAN

PERANCANGAN PI KONTROLER PADA KONTROL KECEPATAN MOTOR DC DENGAN KOMBINASI *POLE PLACEMENT* DAN *SYMMETRICAL* OPTIMUM

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

CHRISTOPHER IMANTAKA
NIM. 105060300111038 - 63

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
Pada tanggal 18 Agustus 2016

Dosen Pembimbing I	**Dosen Pembimbing II**
Ir. Mochammad Rusli, Dipl. –Ing.	**Dr. Ir. Erni Yudaningtyas, M.T.**
NIP. 19630104 198701 1 001	**NIP. 19650913 199002 2 001**

Mengetahui
Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

JUDUL SKRIPSI:

PERANCANGAN PI KONTROLER PADA KONTROL KECEPATAN MOTOR DC DENGAN KOMBINASI *POLE PLACEMENT* DAN *SYMMETRICAL OPTIMUM*

Nama Mahasiswa : Christopher Imantaka

NIM : 105060300111038-63

Program Studi : Teknik Elektro

Konsentrasi : Teknik Kontrol

Komisi Pembimbing :

Ketua : Ir. Mochammad Rusli, Dipl. -Ing. ……

Anggota : Dr. Ir. Erni Yudaningtyas, M.T. ……

TIM DOSEN PENGUJI:

Dosen Penguji 1 : Dr. Ir. Bambang Siswojo, M.T. ……

Dosen Penguji 2 : Rahmadwati, S.T., M.T., Ph.D ……

Dosen Penguji 3 : Ir. Purwanto, M.T. ……

Tanggal Ujian : 9 Agustus 2016

SK Penguji : No.974/UN10.6/SK/2016

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 11 April 2016

Mahasiswa,

CHRISTOPHER IMANTAKA
NIM. 105060300111038

RINGKASAN

Christopher Imantaka, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, April 2016, Perancangan PI Kontroler Pada Kontrol Kecepatan Motor DC Dengan Kombinasi *Pole Placement* dan *Symmetrical Optimum*, Dosen Pembimbing: Mochammad Rusli dan Erni Yudaningtyas.

Terdapat banyak metode dalam perancangan sistem kontrol. Salah satu metode dalam merancang sistem kontrol adalah dengan menggunakan metode *pole placement*. Penempatan *pole*/kutub pada metode *pole placement* masih bersifat *trial* dan *error*. *Symmetrical optimum* merupakan metode sistem kontrol dengan memaksimalkan *phase margin* dari sistem kontrol dan mengarahkan ke fasa yang simetri dan karakteristik amplitudo. Dengan menggunakan *symmetrical optimum*¸ dicari nilai kontrol PI yang nantinya akan digunakan untuk mendapatkan nilai *pole*/kutub untuk metode *pole placement*. Kombinasi dari metode *pole placement* dengan *symmetrical optimum* adalah salah satu cara untuk mendapatkan *pole* yang terbaik. Hasil yang dicapai dalam penelitian ini adalah respon kecepatan motor DC yang mampu mengikuti setiap *setpoint*.

Kata kunci: kontroler, kontrol PI, motor DC, *pole placement, symmetrical optimum*.

SUMMARY

Christopher Imantaka, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, July 2016, Design PI Controller In DC Motor Speed Control Using the combination of Pole Placement and Symmetrical Optimum , Academic Supervisor:* Mochammad Rusli and Erni Yudaningtyas.

*There are many methods in designing control systems. One method in designing the control system is using pole placement. Placement pole/*kutub *to pole placement method is still trial and error. Symmetrical optimum control system is a method to maximize the phase margin of the control system and leads to a symmetric phase and amplitude characteristics. Using symmetrical sought optimal PI control value that will be used to get the pole/*kutub *to pole placement method. The combination of symmetrical method with optimal pole placement is one way to get the best pole. The results achieved in this study was the response speed of a DC motor which is able to follow each setpoint.*

Kata kunci: *controller,* PI *controller, DC motor, pole placement*, *symmetrical optimum*.

PENGANTAR

Puji Syukur kepada Tuhan Yang Maha Esa yang telah memberikan berkat dan kasih karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul ”Perancangan PI Kontroler Pada Kontrol Kecepatan Motor DC Dengan Kombinasi *Pole Placement* dan *Symmetrical Optimum*” dengan baik. Skripsi ini disusun sebagai salah satu syarat memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

- Tuhan Yesus Kristus yang telah memberikan kelancaran, berkat dan kasih karunia-Nya.
- Keluarga tercinta, kedua orang tua Edy Sukamto dan Nengah Suwartini yang selalu memberikan kasih sayang dan memberikan doa serta pengorbanan yang tiada henti. Adik Calvin, Samuel dan Gisella tercinta yang selalu memberikan dukungan.
- Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya
- Bapak Ali Mustofa, ST., MT. selaku Ketua Program Studi S1 Jurusan Teknik Elektro Universitas Brawijaya
- Bapak Ir. Mochammad Rusli, Dipl. Ing. sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Ibu Dr. Ir. Erni Yudaningtyas, MT. selaku Ka. Lab Sistem Kontrol sekaligus sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Joko Purnomo, terima kasih atas bantuan dan waktunya untuk membantu menyelesaikan skripsi ini.

- Teman-teman Ayo Main, Atika, Ayu, Abdulaziz, Dancing, Davin, Enov, Rara, Resi dan Zara. Terima kasih atas dukungan dan bantuan dalam menyelesaikan skripsi ini.
- Keluarga besar Sistem Kontrol angkatan 2010, teman-teman angkatan 2010 "MAGNET" atas semangat dan dukungan yang diberikan pada penulis.
- Keluarga besar Kingkong Hardcore, Emak Winda, Apit, Rizki, Evi, Kephet, Dedi, Pamin, Fery, Ali, Dion, Firman terima kasih atas dukungan dan bantuan dalam menyelesaikan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Juli 2016

Penulis

DAFTAR ISI

hal

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Motor *Direct Current* (DC) adalah penggerak yang sering digunakan dalam teknologi kontrol baik dalam industri maupun rumah tangga. Motor DC memudahkan pekerjaan sehingga proses industri dapat berjalan efisien. Motor DC memiliki respon yang cepat, namun masih memiliki *error steady state*. Oleh karena itu, dibutuhkan suatu kontroler yang tepat dan sesuai dengan plant sistem (Faisol, 2014).

Dalam suatu sistem kontrol ada beberapa macam kontroler, salah satunya adalah kontroler *proporsional* (P) dan kontroler *integral* (I). Pengontrolan *proporsional* (P) memiliki keluaran yang sebanding atau *proporsional* dengan besarnya sinyal kesalahan selisih antara besaran yang diinginkan dengan harga aktualnya. Pengontrolan *integral* (I) berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan *steady state*. Masing-masing kontroler ini mempunyai keunggulannya tertentu, dimana aksi kontrol *proporsional* (P) mempunyai keunggulan *rise time* yang cepat dan aksi kontrol *integral* mempunyai keunggulan untuk memperkecil *error*. Kontroler *proporsional integral* (PI) adalah penggabungan antara kontrol *proporsional* (P) dengan kontrol *integral* (I). Kontrol PI sendiri berfungsi untuk mempercepat output dan mengurangi *offset*. Sehingga dengan menggunakan kontrol PI dalam kontrol kecepatan motor DC dapat memberi keuntungan dalam memperbaiki respon transien, mengurangi *error steady state* dan memberi efek redaman.

Pada desain sistem kontrol yang baik harus memenuhi persyaratan-persyaratan tertentu yang telah ditetapkan. Persyaratan yang harus dipenuhi sistem kontrol disebut sebagai indeks untuk kerja (*performance index*). Indeks ini berkaitan dengan ketelitian, kestabilan dan waktu *output* mencapai *steady state*. Terdapat banyak metode dalam perancangan sistem kontrol. Terdapat beberapa cara untuk mendapatkan nilai-nilai dalam memenuhi kontroler *proporsional* (P) dan kontroler *integral* (I). Salah satu metode perancangan sistem kontrol adalah menggunakan metode *pole placement*. Dari hasil desain kriteria performansi yang diinginkan, ditentukan lokasi pole yang diperlukan. Kestabilan suatu sistem kontrol lup tertutup, baik waktu kontinu maupun waktu diskrit ditentukan oleh letak *pole*. Meskipun suatu sistem sudah stabil, belum tentu *pole* dari sistem tersebut sesuai dengan yang diinginkan karena hal ini menetukan tingkat kecepatan

mencapai *steady state* (Nugroho, 2013). Dalam perancangan motor DC, terdapat pula metode optimum. Salah satunya metode optimum adalah *symmetrical optimum*. Metode *symmetrical optimum* (SO) diusulkan oleh Kessler pada tahun 1958 dan dimodifikasi lebih lanjut oleh Voda dan Landau pada tahun 1955 untuk memastikan bahwa kontroler dapat mencapai fasa maksimum dalam sistem loop tertutup. Metode *symmetrical optimum* dirancang untuk memaksimalkan *phase margin* dari sistem kontrol yang telah dirancang (Martin Macaba, 2012).

Penempatan *pole* pada metode *pole placement* masih bersifat *trial* dan *error*. Oleh karena itu dengan kombinasi *pole placement* dan *symmetrical optimum* akan dimungkinkan perancangan sistem kontrol pada motor DC mendapatkan hasil seperti yang diinginkan.

1.2 Rumusan Masalah

1. Bagaimana cara menentukan parameter PI menggunakan metode *symmetrical optimum*.
2. Bagaimana respon sistem terhadap kombinasi kontroler *pole placement* dan *symmetrical optimum*.

1.3 Batasan Masalah

Untuk menekankan pada objek pembahasan yang ada, maka penelitian ini diberikan batasan masalah sebagai berikut :

1. Motor DC yang digunakan adalah motor DC RF-130CH dengan *operating range* 2 - 6V, arus 0.036 A.
2. *Range* kecepatan motor yang diatur adalah 4000 rpm, 5000 rpm, 6000 rpm, 7000 rpm dan 8000 rpm.
3. Pembahasan dititik beratkan pada analisa *respon transient* pada plant.
4. Kontroler yang digunakan adalah Mikrokontroler Arduino Uno R3.
5. Kinerja *driver* dan elektronika tidak dibahas mendalam.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk merancang kontroler pada motor DC dalam kombinasi *Pole Placement* dan *Symmetrical Optimum*.

1.5 Sistematika Penulisan

Agar penyusunan laporan skripsi ini dapat mencapai sasaran dan tidak menyimpang dari judul yang telah ditentukan, maka diperlukan sistematika pembahasan yang jelas. Pembahasan dalam skripsi ini secara garis besar adalah sebagai berikut:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Tinjauan Pustaka

Menjelaskan teori dasar yang berisi penjelasan tentang teori Motor DC, Kontroler, Kontroler PI, rangkaian Arduino Uno, *Pole Placement*, *Rotary Encoder* dan *Symmetrical Optimum*.

BAB III Metodologi Penelitian

Menjelaskan tentang metodologi penelitian yang terdiri atas studi literatur, perancangan alat, pembuatan alat, pengujian alat, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan dan Pembuatan Alat

Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

BAB V Pengujian dan Analisis

Menjelaskan tentang pengujian alat dan analisa yang meliputi pengujian bagian blok sistem secara keseluruhan.

BAB IV Penutup

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perealisasikan dan pengujian alat sesuai dengan tujuan dan rumusan masalah, serta pemberian saran untuk pengembangan.

UNIVERSITAS BRAWIJAYA

BAB II
TINJAUAN PUSTAKA

2.1 Kontroler

Sistem pengontrolan dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengontrolan adalah harus stabil. Disamping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K., 1997).

Kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Hal itu disebabkan oleh tidak dapat dirubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya. Perubahan perilaku sistem hanya dapat dilakukan melalui penambahan kontroler.

Salah satu fungsi kontroler adalah mengurangi sinyal *error*, dan sinyal *error* adalah perbedaan nilai *setpoint* dengan nilai *ouput plant*. *Setpoint* adalah nilai referensi atau nilai yang diinginkan, sedangkan *ouput plant* adalah nilai aktual yang terukur pada *output plant*. Semakin kecil nilai sinyal *error* maka kinerja sistem kontrol dinilai semakin baik.

Prinsip kerja kontroler adalah membandingkan nilai *output plant* dengan nilai *setpoint*, menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata K., 1997).

2.1.1 Kontroler *Proporsional* (P)

Kontroler *proporsional* (P) adalah sebuah kontroler yang memiliki karakteristik mempercepat output. Hubungan antara output kontroler u(t) dan sinyal *error* e(t) ditunjukkan dalam persamaan berikut :

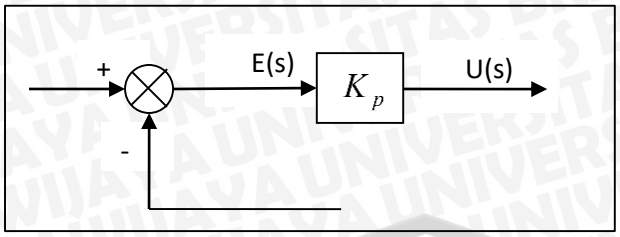
$$u(t)=K_p\, e(t) \qquad (2.1)$$

atau dalam fungsi alih

$$\frac{U(s)}{E(s)} = Kp$$

dimana K_p adalah penguatan.

Diagram blok kontroler *proporsional* (P) ditunjukkan dalam gambar 2.1.



Gambar 2.1 Diagram blok kontroler proporsional (P)
Sumber : Ogata, K. (2010)

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler *proporsional* pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K., 2010).

2.1.2 Kontroler *Integral* (I)

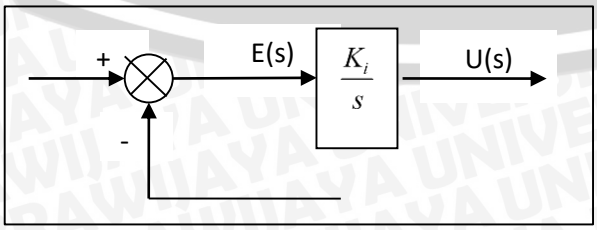
Kontroler *integral* (I) memiliki kemampuan untuk mengurangi *offset* yang diakibatkan oleh kontroler *proporsional* (P). Output kontroler u(t) diubah dengan laju yang sebanding dengan *error* e(t) (Ogata K., 2010). Persamaan kontroler *integral* (I) ditunjukkan dalam persamaan 2.2

$$\frac{du(t)}{dt} = K_i\, e(t)$$

$$u(t) = K_i \int_0^t e(t)\, dt$$

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \qquad (2.2)$$

yang merupakan fungsi alih kontroler *integral* (I), dengan K_i adalah konstanta integral yang dapat diubah nilainya. Jika e(t) bernilai nol, makan nilai m(t) tetap konstan. Aksi kontrol *integral* (I) biasa disebut dengan kontrol reset. Diagram blok kontroler *integral* (I) ditunjukkan pada gambar 2.2.



Gambar 2.2 Diagram Blok Kontroler Integral (I)
Sumber : Ogata K. 1997

2.1.3 Kontroler *Proporsional Integral* (PI)

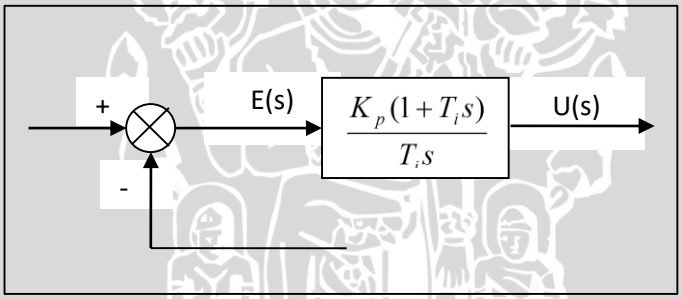
Kontroler *proporsional integral* (PI) memiliki kemampuan untuk mempercepat output dan mengurang *offset*. Persamaan kontroler *proporsional integral* (PI) adalah

$$u(t) = K_p\, e(t) + \frac{K_p}{T_i}\int_0^t e(t)\, dt \qquad (2.3)$$

Adapun fungsi alihnya adalah

$$\frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i\, s}\right)$$
$$= \frac{K_p\,(1 + T_i\, s)}{T_i\, s}$$

dengan K_p penguatan proporsional dan T_i disebut waktu integral, yang keduanya dapat ditentukan. Waktu integral mengatur aksi kontrol internal sedangkan perubahan nilai K_p berakibat pada pada bagian aksi kontrol proporsional maupun integral. Diagram blok kontroler *proporsional integral* (PI) ditunjukkan pada gambar 2.3.



Gambar 2.3 Diagram Blok Kontroler Proporsional Integral (PI)
Sumber : Ogata, K. (2010)

2.2 Motor DC RF 130-CH

Motor DC merupakan motor listrik yang sangat sering digunakan sebagai elemen kontrol akhir dalam sistem kontrol posisi dan kecepatan. Prinsip kerja motor DC sesuai dengan hukum Lorenz, apabila arus dialirkan melalui kumparan jangkar dari mesin DC dan kumparan medannya diberi penguatan, maka akan timbul Gaya Lorentz pada tiap sisi kumparan jangkar tersebut (Soemarwanto, 2010).

Besar gaya Lorentz yang ditimbulkan motor DC adalah

$$F = B.I.l \text{ dyne} \qquad (2.4)$$

dengan:

B = kerapatan medan magnetik (Gauss)

I = Arus Listrik (Ampere)

l = panjang konduktor (cm)

2.2.1 Prinsip Dasar Motor DC

Keuntungan utama motor DC adalah sebagai kontrol kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur.

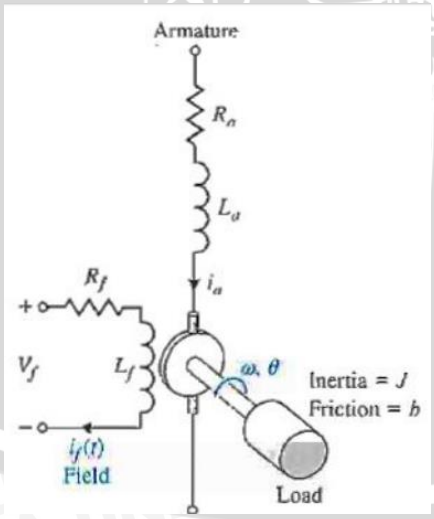
1. Tegangan rotor – meningkatkan tegangan rotor akan meningkatkan kecepatan.
2. Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik.

2.2.3 Fungsi Alih Motor DC

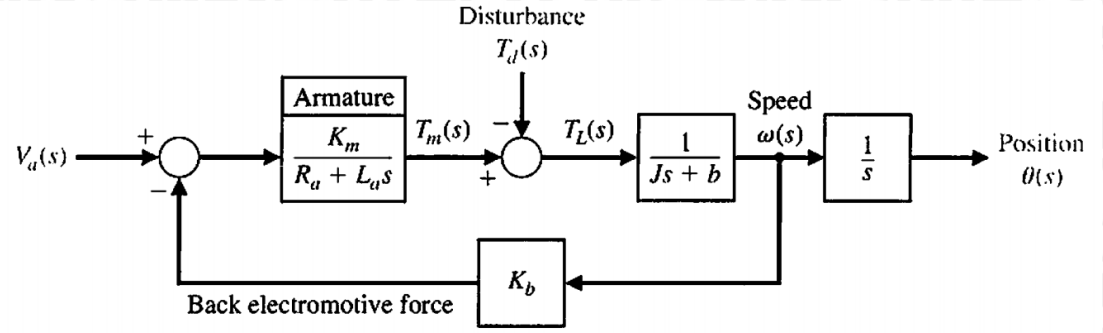
Motor *Direct Current* (DC) merupakan aktuator yang banyak digunakan dalam teknologi kontrol. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan dalam satu lilitan disebut *armature* (jangkar). *Armature* adalah sebuatan untuk komponen yang berputar di antara medan magnet.

Rangkaian ekivalen motor DC magnet permanen dapat dilihat pada Gambar 2.4.



Gambar 2.4 Rangkaian ekivalen motor DC magnet permanen
Sumber: Dorf, R. C. & Bishop, R. H. (2011)

Pada motor DC magnet permanen, arus kumparan medan I_f dibuat konstan. Diagram blok pengontrolan kecepatan motor DC magnet permanen dapat dilihat dalam Gambar 2.5.



Gambar 2.5 Diagram blok pengontrolan kecepatan motor DC magnet permanen
Sumber: Dorf, R. C. & Bishop, R. H. (2011)

Apabila *disturbance* (gangguan) $T_d(s)=0$, maka fungsi alih motor DC adalah

$$\frac{\omega(s)}{V_a\,(s)} = \frac{K_m}{(R_a + L_a s)(\mathrm{Js} + \mathrm{b}) + (K_b K_m)}$$

dengan:

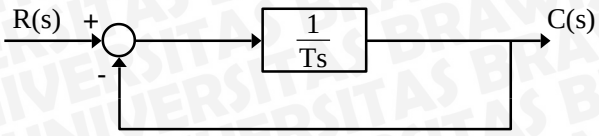
K_m = konstanta motor
K_b = konstanta *back electromotive force*
R_a = resistansi *armature*
L_a = induktansi *armature*
J = inersia
b = gesekan

Salah satu jenis motor DC adalah motor DC magnet permanen RF 130-CH seperti dalam Gambar 2.6.

Gambar 2.6 Motor DC magnet permanen RF 130-CH

2.3 Output Sistem Orde Satu

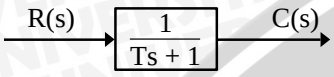
Diagram blok sistem orde satu dapat dilihat dalam Gambar 2.7 yang mempunyai fungsi alih sebagai berikut



Gambar 2.7 Sistem orde satu

$$\frac{C(s)}{R(s)} = \frac{1/Ts}{1+1/Ts} \qquad (2.5)$$

$$= \frac{1}{Ts+1}$$



2.3.1 Output Unit Step Pada Sistem Orde Satu

Jika sistem orde satu diberi masukan unit step r(t)= 1, dari Transformasi Laplace fungsi ramp adalah R(s) = $1/s$. Keluaran sistem adalah sebagai berikut

$$C(s) = \frac{1}{Ts+1} R(s)$$

$$C(s) = \frac{1}{Ts+1}\frac{1}{s}$$

$$= \frac{1}{s(Ts+1)} \qquad (2.6)$$

$$c(t) = \mathcal{L}^{-1}[C(s)]$$

$$c(t) = 1 - e^{-t/T} \quad (t \geq 0) \qquad (2.7)$$

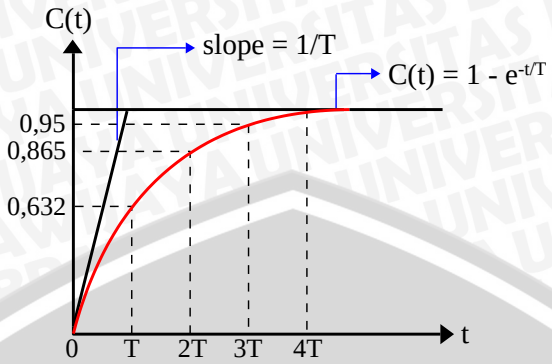
Persamaan keluaran tersebut menyatakan bahwa keluaran c(t) mula-mula nol kemudian akhirnya menjadi satu. Salah satu karakteristik penting output eksponensial c(t) tersebut adalah bahwa pada t = T, maka $c(t) = 1 - e^{-1} \cong 0.632 = 2/3$

T = *time constant* / konstanta waktu sistem

= $2/3$ harga akhir

Konstanta waktu T yang lebih kecil akan mempercepat output sistem. Karakteristik penting lainnya pada kurva output eksponensial adalah kemiringan garis singgung / gradien pada t = 0 adalah $1/T$, karena

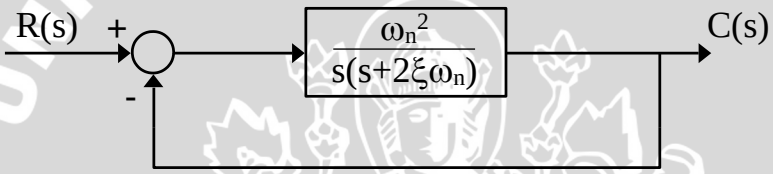
$$\frac{dc}{dt} = \frac{1}{T} e^{-t/T}\Big|_{t=0} = \frac{1}{T}$$



Gambar 2.8 Output unit step sistem orde satu

2.4 Output Sistem Orde Dua

Sistem orde dua dengan fungsi alihnya adalah sebagai berikut:



Gambar 2.9 Sistem orde dua

$$\frac{C(s)}{R(s)} = \frac{\frac{\omega_n^2}{s(s+2\xi\omega_n)}}{1+\frac{\omega_n^2}{s(s+2\xi\omega_n)}}$$

$$= \frac{\omega_n^2}{s(s+2\xi\omega_n)+\omega_n^2}$$

$$= \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} \qquad (2.8)$$

Akar-akar penyebut fungsi alih atau persamaan karakteristik adalah

$$s_{12} = \frac{-2\xi\omega_n \pm \sqrt{(2\xi\omega_n)^2-4\omega_n^2}}{2}$$

$$s_{12} = -\xi\omega_n \pm \omega_n\sqrt{\xi^2-1}$$

$$= -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2}$$

$$s_{12} = -\xi\omega_n \pm j\omega_d \qquad (2.9)$$

dimana

ξ = rasio peredaman sistem (*damping ratio*)

ω_n = frekuensi natural/alamiah tak teredam
ω_d = frekuensi natural/alamiah teredam

Kelakuan dinamik sistem orde dua dapat digambarkan dalam suku dua parameter ξ dan ω_n . Jika $(0 < \xi < 1)$, maka pole loop tertutup merupakan konjugat kompleks dan berada pada bidang s sebelah kiri. Dalam hal ini, sistem dikatakan dalam peredaman dan tanggapan peralihan berosilasi. Jika $(\xi = 1)$, maka sistem dikatakan teredam kritis. Sistem terlalu teredam berhubungan dengan $(\xi > 1)$. Tanggapan peralihan sistem teredam kritis dan sistem terlalu teredam tidak berosilasi. Jika ξ=0, tanggapan peralihan tidak muncul.

Pada sistem orde dua seperti terlihat dalam Gambar 2.13, berdasarkan output sistem dengan masukan unit step akan terdapat tiga keadaan yang berbeda yaitu keadaan teredam $(0 < \xi < 1)$, teredam kritis $(\xi = 1)$, dan sistem terlalu teredam $(\xi > 1)$.

2.4.1 Keadaan Kurang Teredam / Underdamped $(0 < \xi < 1)$

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \qquad (2.10)$$

Jika sistem diberi input berupa unit step atau $R(s) = \frac{1}{s}$, maka:

$$C(s) = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \qquad (2.11)$$

Dari Tabel Transformasi Laplace didapatkan

$$c(t) = 1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin(\omega_n \sqrt{1-\xi^2}\, t + \phi)$$

$$\phi = \arctan\frac{\sqrt{1-\xi^2}}{\xi}$$

Jika $\omega_d = \omega_n\sqrt{1-\xi^2}$; maka

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_d t + \arctan\frac{\sqrt{1-\xi^2}}{\xi}\right) \quad t \geq 0 \qquad (2.12)$$

Output sistem tersebut juga bisa diperoleh dengan menggunakan Transformasi Laplace balik jika C(s) ditulis dalam bentuk berikut:

$$C(s) = \frac{1}{s} - \frac{s + 2\xi\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2} \qquad (2.13)$$

$$= \frac{1}{s} - \frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2} - \frac{\xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}$$

$$\mathscr{L}^{-1}\left[\frac{s+\xi\omega_n}{(s+\xi\omega_n)^2+\omega_d^2}\right] = e^{-\xi\omega_n t}\cos\omega_d t$$

$$\mathscr{L}^{-1}\left[\frac{\xi\omega_n}{(s+\xi\omega_n)^2+\omega_d^2}\right] = e^{-\xi\omega_n t}\sin\omega_d t \qquad (2.14)$$

oleh karena itu, transformasi laplace balik dari persamaan

$$C(s) = \frac{\omega_n^2}{(s^2+2\xi\omega_n s+\omega_n^2)s}$$

diperoleh sebagai

$$\mathscr{L}^{-1}[C(s)] = c(t)$$

$$c(t) = 1 - e^{-\xi\omega_n t}\left(\cos\omega_d t + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d t\right)$$

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}}\sin\left(\omega_d t + \arctan\frac{\sqrt{1-\xi^2}}{\xi}\right) \quad t \geq 0 \qquad (2.15)$$

Sinyal kesalahan / error adalah e(t) = r(t) – c(t), dimana

r(t) = 1

dan

$$c(t) = 1 - e^{-\xi\omega_n t}\left(\cos\omega_d t + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d t\right)$$

sehingga

$$e(t) = e^{-\xi\omega_n t}\left(\cos\omega_d t + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d t\right) \quad t \geq 0$$

$$\text{jika } \xi = 0 \quad \Rightarrow \quad c(t) = 1 - \cos\omega_n t$$

2.4.2 Teredam Kritis / Critically Damped $(\xi = 1)$

Dalam hal ini apabila dua pole $C(s)/R(s)$ hampir sama, maka sistem dapat didekati dengan bentuk teredam kritis. Jika input berupa unit step dimana R(s) = 1/s maka:

$$C(s) = \frac{\omega_n^2}{(s+\omega_n)^2 s} \qquad (2.16)$$

$$c(t) = 1 - e^{-\omega_n t}(1+\omega_n t) \qquad t \geq 0$$

2.4.3 Terlalu Teredam / Overdamped $(\xi > 1)$

Dalam hal ini pole $C(s)/R(s)$ adalah bilangan nyata / real negatif yang tidak sama.

Jika input berupa unit step dimana R(s) = 1/s dan C(s) dapat ditulis dengan :

$$C(s) = \frac{\omega_n^2}{(s+\omega_n)^2 s} \qquad (2.17)$$

$$C(s) = \frac{\omega_n^2}{\left(s+\xi\omega_n+\omega_n\sqrt{\xi^2-1}\right)\left(s+\xi\omega_n-\omega_n\sqrt{\xi^2-1}\right)s}$$

$$c(t) = 1 + \frac{\omega_n}{2\sqrt{\xi^2-1}}\left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2}\right) \qquad t \geq 0$$

dengan $s_1 = \left(\xi+\sqrt{\xi^2-1}\right)\omega_n$

$$s_2 = \left(\xi-\sqrt{\xi^2-1}\right)\omega_n \qquad (2.18)$$

Tanggapan c(t) terdiri dari dua suku eksponensial menurun.

2.5 Tahapan Peralihan

Sistem dengan tenaga tidak dapat memberikan tanggapan seketika dan akan menujukkan tanggapan peralihan walaupun diberi masukan ataupun gangguan. Karakteristik unjuk kerja sistem kontrol yang diinginkan dicirikan oleh suku tanggapan peralihan terhadap masukan unit step karena hal itu mudah dilakukan dan cukup drastis. Jika tanggapan terhadap masukan unit step diketahui, secara matematis dapat dihitung tanggapan untuk masukan yang lain.

Tanggapan peralihan sistem kontrol selalu menujukkan osilasi teredam sebelum mencapai keadaan mantapnya, hal ini juga menunjukkan bahwa sistem tersebut mempunyai rasio peredaman $(0 < \xi < 1)$ yang juga berarti bahwa sistem tersebut merupakan sistem yang kurang teredam / underdamped.

Tanggapan peralihan sistem kontrol terhadap masukan unit step umumnya dikelompokkan sebagai berikut:

1) Delay Time / Waktu Tunda, t_d

   Waktu yang dibutuhkan oleh outputs untuk mencapai setengah harga akhir pada saat lonjakan pertama

2) Rise Time / Waktu Naik, t_r

Waktu yang dibutuhkan oleh outputs agar bertambah dari 10% menjadi 90% dari nilai akhir

3) Peak Time / Waktu Puncak, t_p

Waktu yang dibutuhkan oleh outputs untuk mencapai puncak pertama lonjakan (maksimum)

4) Maximum Overshoot / Lonjakan Maksimum, M_p

Merupakan nilai puncak kurva outputs diukur dari satu

$$Mp = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

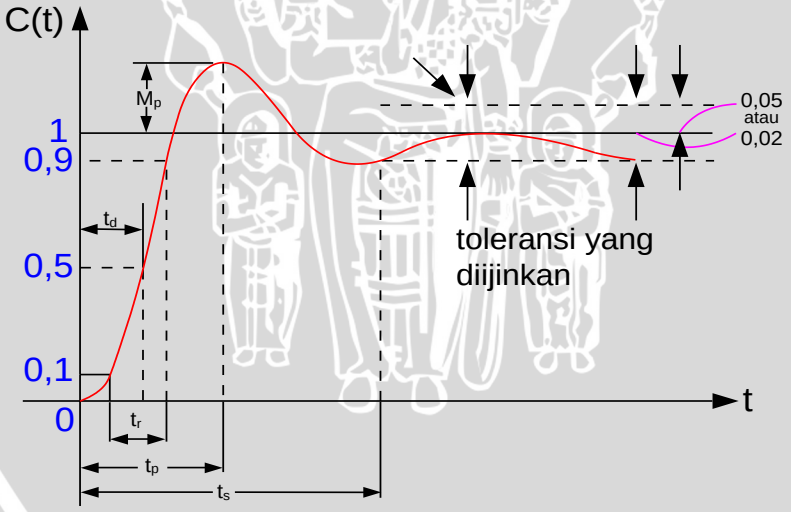
dengan $c(t_p)$ = nilai outputs pada saat lonjakan maksimum.

$c(\infty)$ = nilai outputs pada saat keadaan mantap.

5) Settling Time / Waktu Turun, t_s

Waktu yang dibutuhkan oleh outputs untuk mencapai harga tertentu dan tetap dalam range nilai akhir (biasanya 5% atau 2%)

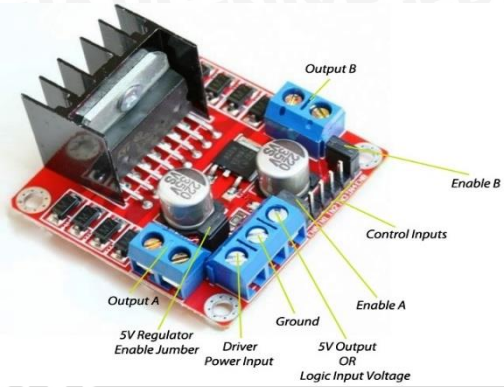
Output unit step sistem orde dua ditunjukkan oleh gambar 2.10



Gambar 2.10 Output Unit Step Sistem Orde Dua

2.6 *Driver* Motor L298N

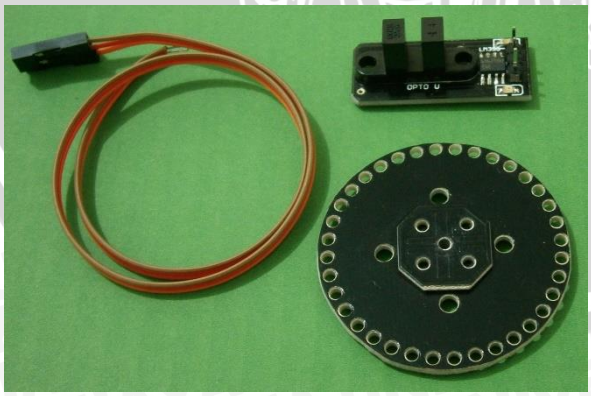
Driver motor L298N seperti ditunjukkan Gambar 2.6 digunakan untuk mengendalikan putaran motor DC. Modul ini dihubungkan dengan output dari mikrokontroler Arduino Uno.



Gambar 2.11 Driver Motor LN298N
(Sumber: electrosome.com)

2.7 Rotary Encoder

Rotary encoder adalah *elektromekanik* yang dapat mendeteksi atau memonitor gerakan dan posisi. *Rotary encoder* biasanya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat dijadikan gerakan, posisi dan arah. Rotary encoder adalah sensor rotari atau putaran yang terdiri dari dua bagian, yaitu satu rangkaian sensor optocoupler dan piringan derajat yang terdiri dari beberapa lubang yang adalah *trigger* dari sensor optocoupler.



Gambar 2.12 Rotary Encoder
(Sumber: depokinstruments.com)

2.8 Arduino Uno

Arduino Uno adalah *board* mikrokontroler berbasi Atmega328. Arduino Uno memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz *osilator* kristal, koneksi USB, *jack power*, ICSP *header* dan tombol *reset*. Supaya mikrokontroler dapat digunakan, cukup hanya menggubungkan *board* arduino uno ke komputer dengan menggunakan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.

2.8.1 Catu Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (non-USB) daya dapat datang dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya *plug* pusat-positif 2.1mm ke dalam board colokan listrik. Lead dari baterai dapat dimasukkan ke dalam *header* pin Gnd dan Vin dari konektor *Power*.

Board dapat beroperasi pada pasokan daya dari 6 – 20 volt. Jika diberikan kurang dari 7V, bagaimanapun pin 5V dapat menyuplai kurang dari 5 volt dan *board* mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7-12 volt. Pin catu daya adalah sebagai berikut :

- VIN = Tegangan input ke *board* Arduino ketika menggunakan sumber daya eksternal (sebagai lawan 5v dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini. Atau jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- 5V = Catu daya diatur untuk daya mikrokontroler dan komponen lainnya di *board*. Hal ini dapat terjadi baik dari VIN melalui regulator on-board atau diberikan oleh USB.
- 3.3V = Volt pasokan yang dihasilkan regulator on-board. Menarik arus maksimum adalah 50 mA.
- GND

2.8.2 Memory

Arduino Uno ini memiliki 32 KB dengan 0,5 KB digunakan untuk *loading file*. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM.

2.8.3 Input dan Output

Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output, menggunakan fungsi pinMode(), digitalWrite() dan digitalRead(). Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal dari 20-50 KΩ. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik, tapi jatuh atau perubahan nilai.
- PWM: 3, 5, 6, 9, 10 dan 11. Menyediakan 8-bit output PWM dengan analogWrite () fungsi.
- SPI: 10 (SS), 11(mosi), 12(MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: Terdapat built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH dan OFF.

Uno memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara *default* sistem mengukur dari GND sampai 5 volt.

- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI.
- Aref: Referensi tegangan atau input analog. Digunakan dengan *analogReference* ().
- Reset.

2.8.4 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui USB dan muncul sebagai

com port virtual untuk perangkat lunak pada komputer. *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun pada windows, file. Inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memunginkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

2.9 *Symmetrical Optimum*

Symmetrical Optimum (SO) adalah metode desain yang didasarkan pada optimasi. Tuning parameter dengan *symmetrical optimum* pertama dikenalkan oleh Kessler pada 1985. Metode ini memaksimalkan *phase margin* dari sistem kontrol dan mengarahkannya ke fasa yang simetri dan karakterisitik amplitudo. Persamaan fungsi alih *symmetrical optimum* ditunjukkan pada persamaan (2.15).

$$F_{ol}(s) = \underbrace{Gc\left(\frac{T_i s + 1}{T_i s}\right)}_{Kontroler\ PI} \underbrace{\frac{G_m}{T_m s + 1}}_{Plant} \frac{1}{T_{mn} s} \tag{2.19}$$

Dengan :

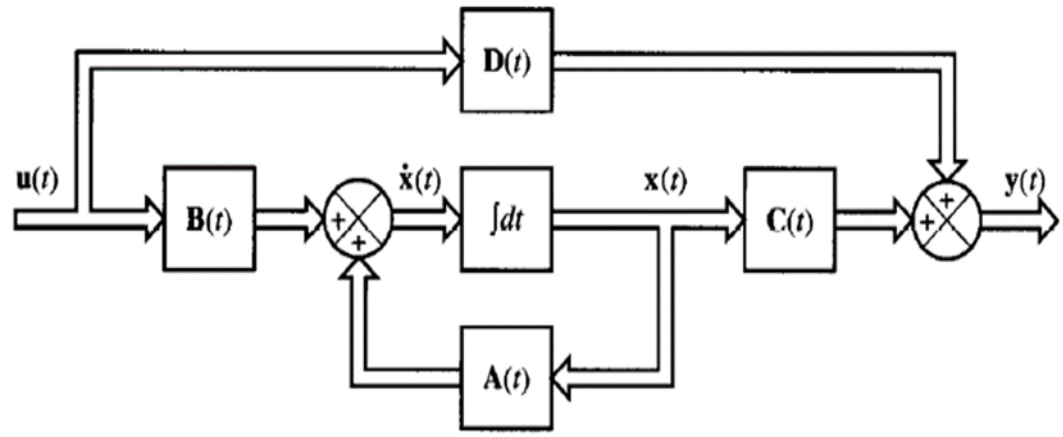
F_{ol} = Fungsi *open loop*

G_m = Nilai *gain* motor

T_m = *Time konstan* motor terkecil

2.10 Konsep Ruang Keadaan (*State Space*)

Proses pemodelan sistem akan selalu dilakukan dalam mendesain sistem kontrol. Model dapat berupa diagram blok, diagram aliran sinyal, maupun dalam bentuk persamaan matematis. Model matematis sebuah sistem didefinisikan sebagai kumpulan informasis dalam bentuk persamaan matematis yang mewakili sistem fisik yang ditinjau. Terdapat suatu metode untuk memodelkan sistem dengan *multi input* atau *multi output*, yang dikenal dengan metode ruang keadaan (*state space*). Model ruang keadaan merupakan sekumpulan persamaan diferensial orde satu yang berhubungan dan ditulis dalam notasi matriks vektor. Bentuk umum ruang keadaan (*state space*) sebuah sistem digambarkan dalam bentuk persamaan *state* dalam Gambar 2.13.



Gambar 2.13 Representasi sistem dalam bentuk persamaan state
Sumber: Ogata, K. (2010)

Persamaan *state*:

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad (2.20)$$

Persamaan *output*:

$$y(t) = Cx(t) + Du(t) \qquad (2.21)$$

dengan:

x(t) = vektor state (n-vektor)
u(t) = vektor kontrol (m-vektor)
y(t) = vektor *output* (p-vektor)
A = matriks keadaan orde n x n
B = matriks *input* orde n x m
C = matriks *output* orde p x n
D = matriks transmisi langsung orde p x n

2.11 Pole Placement

Pole placement adalah suatu metode desain kontrol dimana menentukan letak dari *pole* sistem loop tertutup pada bidang kompleks dengan mengatur *gain kontroler*. Pada pendekatan secara konvensional, kita mengasumsikan bahwa pengaruh respon *pole* loop tertutup tak domain dapat diabaikan. Pada pendekatan dengan penempatan *pole* (*pole placement*), pendekatan penempatan *pole* yang diberikan menetapkan semua loop tertutup. Terdapat persyaratan pada bagian sistem untuk *pole-pole* loop tertutup yang akan ditempatkan pada lokasi yang dipilih secara sembarang yaitu bahwa sistem harus berupa keadaan lengkap yang dapat dikontrol.

Keterkontrolan keadaan secara sempurna dapat dilihat dari sistem waktu

diskrit. Sistem diskrit yang dinyatakan oleh " *x((k + 1)T) = Gx(kT) + Hu(kT)* "

Dimana :

x(kT) = vektor keadaan (vektor n dimensi)
u(kT) = sinyal kontrol
G = matriks nonsinguler n xn
H = matriks n x 1
T = perioda cacah

Kondisi keterkontrolan sempurna diturunkan berdasarkan kenyataan bahwa jika suatu sistem terkontrol sempurna, maka ada sinyal kontrol kontinyu sepotong-sepotong yang akan memindahkan setiap keadaan awal ke titik asal dalam sejumlah perioda cacah yang terhingga.

Observability atau keteramatan sistem linier dapat dinyatakan oleh persamaan berikut:

$$\dot{x}(t) = Ax(t)$$

$$y = Cx(t)$$

dimana :

x = vektor keadaan (vektor n-dimensi)
y = vektor keluaran (vektor m-dimensi)
A = matriks n x n
C = matriks m x n

Keteramatan sempurna dari sistem waktu diskrit. Tinjau sistem yang dinyatakan oleh :

$$x((k+1)T) = Gx(kT)$$

$$y(kT) = Cx(kT)$$

dimana:

x(kT) = vektor keadaan (vektor n-dimensi)
y(kT) = vektor keluaran (vektor m-dimensi)
G = matriks n x n
C = matriks m x n
T = periode cacah

Persamaan ruang keadaan waktu berbentuk :

$\dot{x}(t) = Ax(t) + Bu(t)$ (2.22)

Dan sinyal kontrol diberikan oleh :

$u = -kx$ (2.23)

Dan subtitusi persamaan (2.22) ke persamaan (2.23) sehingga didapatkan

$\dot{x}(t) = (A - BK)x(t)$ (2.24)

Solusi dari persamaan (2.24)

$$x(t) = e^{(A-BK)t}x(0)$$

Kestabilan dari sistem (2.24) ditentukan oleh nilai eigen dari (A-BK) , artinya jika matriks K dapat dipilih secara tepat, maka bagian riil dari nilai eigen matriks (A- BK) terletak di sebelah kiri sumbu imajiner bidang s, hal itu berarti, untuk semua $x(0) \neq 0$, $x(t)=0$ untuk $t \rightarrow \infty$. Syarat perlu dan cukup untuk penempatan pole dinyatakan dalam teorema berikut. :

Teorema [Ogata Katsuhiko, 1997]: "Jika diberikan suatu sistem, maka syarat perlu dan cukup untuk penempatan sembarang pole yang diinginkan adalah bahwa sistem tersebut terkontrol secara lengkap".

BAB III
METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Dalam menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

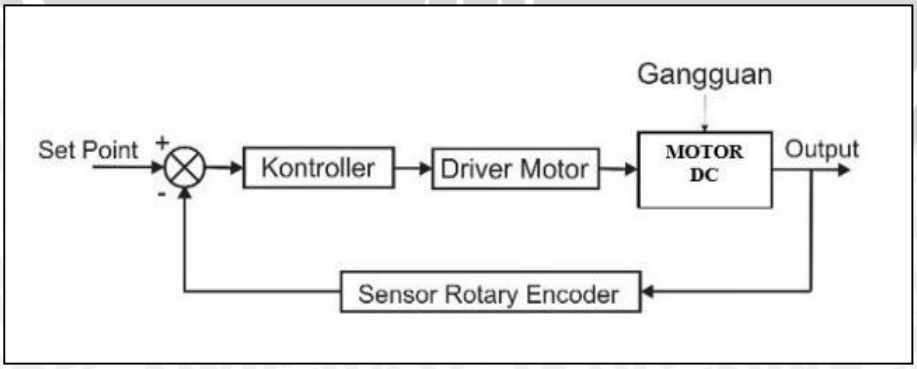
1. Perancangan blok diagram sistem
2. Spesifikasi desain
3. Karakteristik setiap blok

   Karakteristik setiap blok dilakukan untuk mempermudah analisis sistem. Karakteristik dibagi menjadi beberapa bagian :

   1. Karakteristik motor DC.
   2. Karakteristik driver motor.
4. Pembuatan perangkat keras
5. Perancangan algoritma

3.1 Perancangan Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana. Diagram blok sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem Loop Tertutup

Keterangan :

1. *Setpoint* sistem berupa putaran motor yang memiliki lima nilai yaitu 4000 rpm, 5000 rpm, 6000 rpm, 7000 rpm dan 8000 rpm.
2. Kontroler yang digunakan adalah kontorler *Proporsional Integral* (PI) menggunakan perangkat keras Arduino Uno.
3. *Driver* motor menggunakan L298N .
4. Aktuator yang digunakan adalah motor DC RF 130-CH.
5. Sensor *Rotary Encoder* sebagai pembaca putara motor digunakan untuk *feedback* sistem.

3.2 Spesifikasi Desain

Desain yang diinginkan pada perancangan kombinasi *pole placement* dan *symmetrical optimum* mempunyai spesifikasi yaitu:

1. Kontroler yang digunakan adalah kontroler *Proporsional Integral* (PI) menggunakan metode *symmetrical optimum.*
2. *Setpoint* sistem berupa putaan motor yang memiliki lima nilai yaitu 4000 rpm, 5000 rpm, 6000 rpm, 7000 rpm dan 8000 rpm.
3. *Settling time* 1 detik.
4. Pembacaan kecepatan menggunakan sensor *rotary encoder*.

3.3 Karakteristik Motor DC

Karakteristik motor DC dilakukan untuk mengetahui karakter atau gain motor DC. Hasil diperoleh dengan mengamati kecepatan motor DC terhadap perubahan tegangan motor DC.

Peralatan yang digunakan dalam mencari karakteristik dari motor DC terdiri atas:

1. *Power Supply*,
2. Motor DC
3. Kabel penghubung.

Langkah pengujian meliputi:

1. Menghubungkan tegangan *output power supply* dengan motor DC.
2. Mengatur tegangan *output power suplly* dari 0 V sampai 6 V sebagai tegangan sumber motor DC.
3. Menggunakan tachometer digital untuk mengetahui nilai putaran motor.

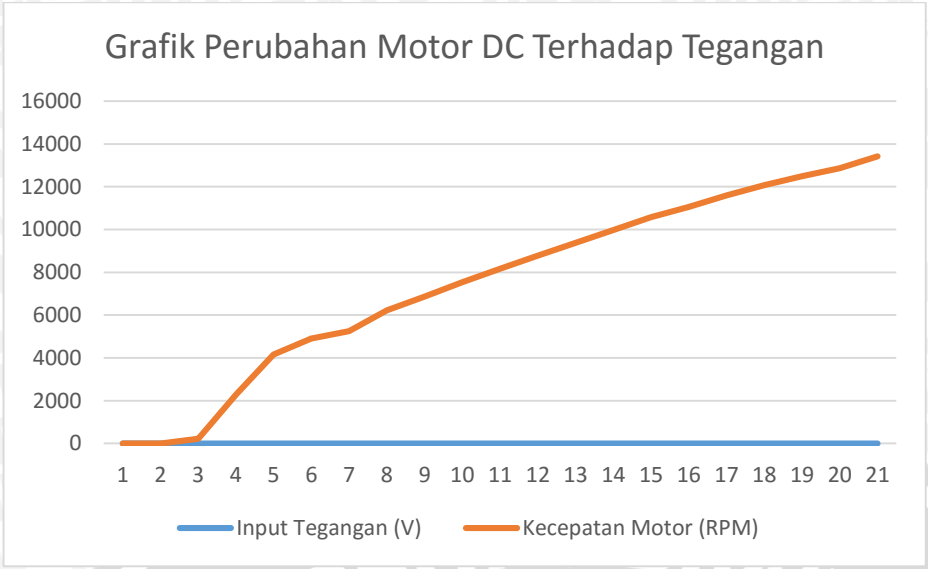
4. Mengamatai dan mencatat hasil pengukuran putaran motor.

Data hasil pengujian tegangan input terhadap kecepatan motor DC ditunjukkan dalam Tabel 3.1.

Tabel 3.1 Data pengujian kecepatan motor DC terhadap tegangan

Input Tegangan (V)	Kecepatan Motor (RPM)
0	0
0,159	0
1,045	225
1,52	2287
1,873	4162
2,168	4912
2,355	5250
2,803	6225
3,08	6862
3,385	7537
3,7	8175
3,966	8775
4,26	9375
4,55	9975
4,84	10575
5,1	11062
5,35	11587
5,6	12075
5,82	12487
6	12862
6,35	13425

Grafik hasil pengujian tegangan input terhadap kecepatan motor DC ditunjukkan dalam Gambar 3.2.



Gambar 3.2 Grafik perubahan kecepatan motor DC terhadap tegangan

3.4 Karakteristik Driver Motor

Pengujian karakteristik driver motor bertujuan untuk mengetahui masukan dutycycle PWM dari mikrokontroler dengan besarnya nilai perubahan Vrms.

Peralatan yang digunakan dalam mencari karakteristik driver motor terdiri atas:

1. Motor DC
2. Kabel
3. Arduino Uno
4. Power Supply
5. Voltmeter

Langkah pengujian meliputi :

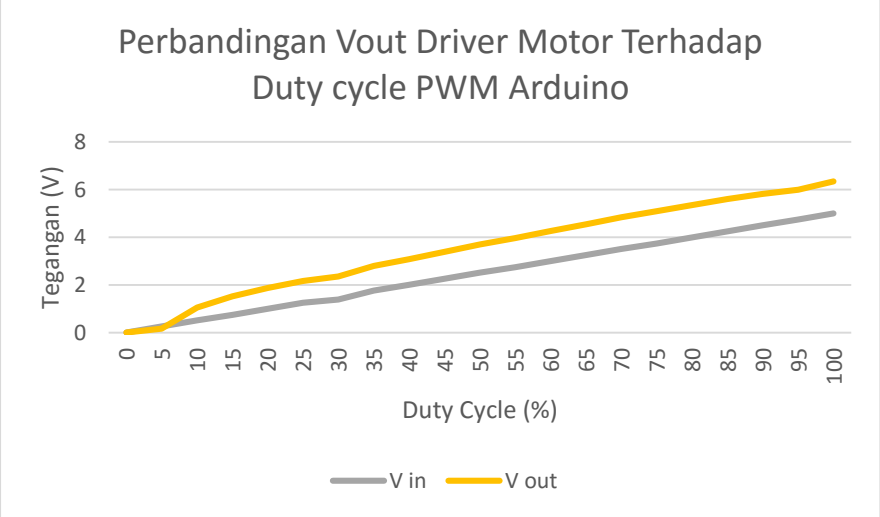
1. Menghubungkan catu motor driver ke *power supply* dan menghubungkan motor ke *driver*.
2. Menghubungkan input tegangan driver motor dengan pin output PWM di Arduino Uno.
3. Menghubungkan output tegangan driver motor dengan multimeter.
4. Mengatur duty cycle sinyal PWM pada arduino uno dengan nilai 0-100%.

Data hasil pengujian driver motor ditunjukkan pada Tabel 3.2.

Tabel 3.2 Data pengujian driver motor

Duty Cycle (%)	Vin	Vout
0	0	0
5	0,25	0,159
10	0,5	1,045
15	0,74	1,52
20	1	1,873
25	1,25	2,168
30	1,39	2,355
35	1,76	2,803
40	2	3,08
45	2,25	3,385
50	2,5	3,7
55	2,74	3,966
60	3	4,26
65	3,25	4,55
70	3,5	4,84
75	3,74	5,1
80	4	5,35
85	4,25	5,6
90	4,5	5,82
95	4,7	6
100	5	6,35

Grafik hasil pengujian *driver* motor ditunjukkan dalam Gambar 3.3.



Gambar 3.3 Grafik perubahan tegangan keluaran driver motor

3.5 Penentuan Fungsi Alih Motor DC

Pengujian identifikasi motor DC dilakukan untuk mendapatkan sebuah fungsi alih. Identifikasi tersebut menggunakan aplikasi Matlab. Fungsi alih sendiri nantinya akan digunakan untuk mengetahui karakteristik dari motor DC yang akan digunakan.

3.5.1 Menentukan Parameter Motor DC

Data yang diperoleh dari pengujian fisik motor DC adalah sebagai berikut :

$V = 1\ volt$

$i = 0{,}05\ A$

$\omega = 3{,}03\ rad/s$

$R = 2{,}8\ \Omega$

$L = 0{,}0057\ H$

$T_m = 0{,}36\ s$

Kemudian hasil dari pengujian fisik motor DC tersebut akan didapatkan parameter-parameter sebagai berikut :

e = 0,86 *volt*

K = 0,2831

b = 0,0047

J = 0,0017

3.5.2 Menentukan Fungsi Alih Motor DC

Setelah mendapatkan parameter-parameter dari motor DC, berikutnya akan ditentukan fungsi alih dengan memasukkan parameter-parameter kedalam simulasi menggunakan MATLAB. Nilai-nilai yang akan dimasukkan kedalam program matlab adalah sebagai berikut:

R = 2,8 Ω

L = 0,0057 *H*

K_t = 0,3200 *Nm/A*

K_e = 0,2831 *Vs/rad*

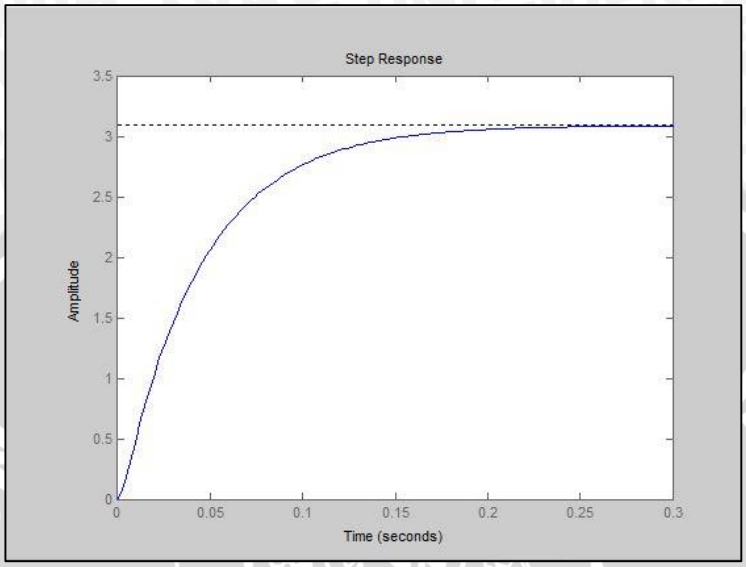
b = 0,0047 *Nms*

J = 0,0017 kgm^2

Data diatas dimasukkan kedalam program matlab sehingga didapatkan fungsi alih motor sebagai berikut ini.

$$Fm(s) = \frac{3{,}347 \times 10^4}{s^2 + 494s + 1{,}084 \times 10^4} \quad \text{.....(3.1)}$$

Dengan memberikan masukan sinyal unit *step* pada persamaan (3.1) maka didapatkan grafik respon plant tanpa kontroler ditunjukkan pada gambar (3.4)



Gambar 3.4 Respon Fungsi Alih Motor DC

3.6 Perancangan Kontroler Proporsional Integral

Dalam kawasan waktu kontroler PI dapat dinotasikan dengan persamaan

$$c(t) = Kp(e(t) + \frac{1}{T_i}\int_0^t e(t)dt \quad \text{.....(3.2)}$$

Dimana c(t) adalah keluaran kontroler, Kp adalah gain proporsional, Ti adalah waktu konstan integral atau *reset time* dan e(t) adalah *error* yang terjadi. Dari persamaan diatas dapat diubah menjadi kawasan frekuensi dengan Transformasi Laplace sehingga menjadi persamaan berikut

$$C(s) = Kp\left(1 + \frac{1}{Ti\ s}\right)E(s) \quad \text{.....(3.3)}$$

Pada persamaan (3.3) belum bisa dimasukkan kedalam arduino, maka dari itu persamaan kontinyu harus diubah kedalam bentuk diskrit melalui Transformasi Z. dalam Transformasi Z dibutuhkan waktu sampling (Ts). Digunakan metode *Billinear Transform* sehingga nilai notasi s pada Laplace setara dengan persamaan (3.4).

$$s = \frac{2}{Ts}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \quad \text{.....(3.4)}$$

Lalu persamaan (3.4) disubstitusikan kedalam persamaan (3.3) menjadi persamaan (3.5).

$$C(z) = \left[Kp + \frac{KpTs(1+z^{-1})}{2Ti(1-z^{-1})}\right] E(z) \quad \text{.....(3.5)}$$

Berikutnya adalah memodifikasi persamaan agar dapat disederhanakan. Kedua ruas pada persamaan (3.5) dikalikan dengan (1-z^{-1}).

$$C(z)(1-z^{-1}) = Kp\,E(z)(1-z^{-1}) + \frac{Kp\,Ts}{2Ti} E(z)(1+z^{-1}) \quad \text{.....(3.6)}$$

Kemudian pada persamaan (3.6) disusun kembali menjadi keluaran kontroler dan diubah menjadi persamaan beda sehingga didapatkan persamaan (3.7).

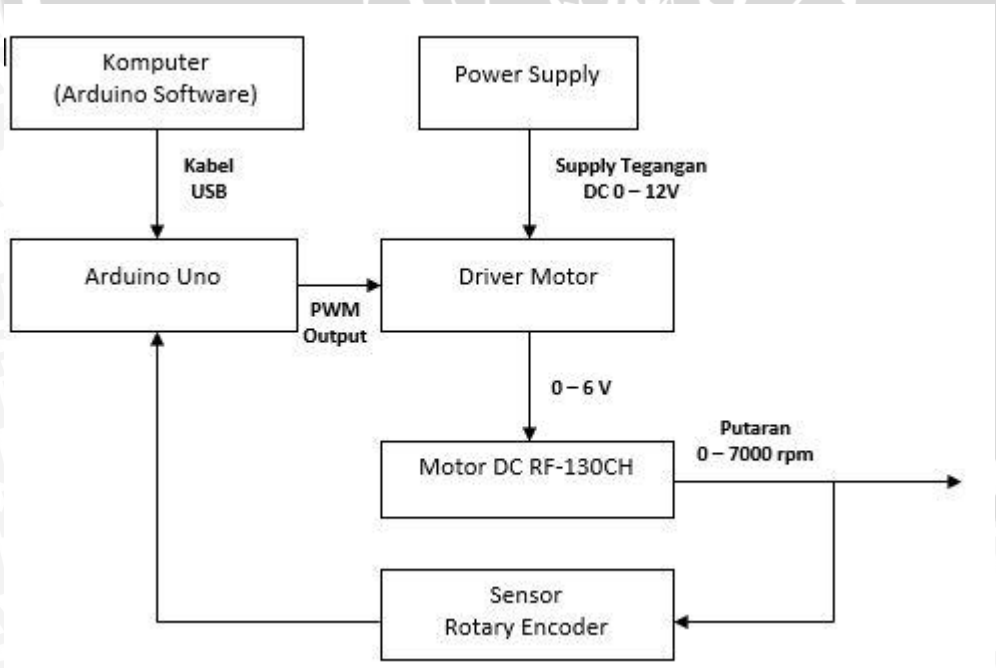
$$C(k) = C(k-1) + Kp\left(E(k) - E(k-1)\right) + \frac{Kp\,Ts}{2Ti}(E(k) + E(k-1)) \quad \text{.....(3.7)}$$

Pada persamaan (3.7), k-1 adalah kondisi sebelumnya. Persamaan diatas lalu dimasukkan kedalam program pada arduino.

3.7 Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya sistem beserta programnya. Hal ini dimaksudkan agar rpm kecepatan motor DC sesuai dengan setpoint yang diinginkan dan sistem dapat bekerja dengan baik sesuai yang direncanakan.

1. Skema pembuatan perangkat keras (Gambar 3.5)



Gambar 3.5 Skema Perangkat Keras

2. Penentuan modul elektronik yang digunakan meliputi :

- Power Supply Unit sebagai catu daya driver motor memiliki range 0 – 12 Volt.

Gambar 3.6 Power supply unit (PSU)

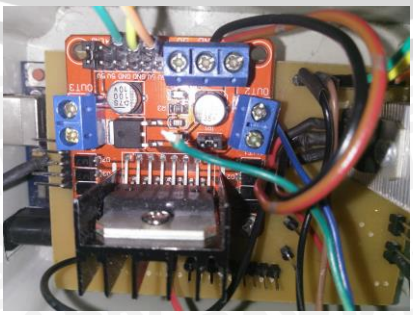
- Mikrokontroler Arduino Uno R3 sebagai perangkat kontroler.

Gambar 3.7 Arduino Uno R3

- Sensor *rotary encoder* sebagai *feedback*.

Gambar 3.8 Sensor Rotary Encoder

- Driver Motor.



Gambar 3.9 Driver Motor

- Komputer atau PC yang sudah terinstall *software* arduino.
- Motor DC RF-130CH

3.8 Penentuan Konstanta Kp dan Ki dengan Metode Symmetrical Optimum

Dalam menentukan parameter – parameter dalam kontroler PI menggunakan metode *symmetrical optimum*. Tuning paramter dengan *symmetrical optimum* pertama dikemukakan oleh Kessler pada 1958. Metode ini memaksimalkan *phase margin* dari sistem kontrol dan mengarahkan ke fasa yang simetris dan karakteristik amplitudo. Dengan memodifikasi persamaan fungsi alih *open loop* sistem menjadi

$$F_{ol}(s) = \underbrace{Gc\left(\frac{T_i s + 1}{T_i s}\right)}_{Kontroler\ PI} \underbrace{\frac{G_m}{T_m s + 1}}_{Plant} \frac{1}{T_{mn} s}$$

Dalam membentuk persamaan *plant* diatas maka fungsi alih motor perlu diubah. Gm adalah nilai gain dari motor yang telah dimodifikasi dan Tm adalah time konstan motor terkecil. Pada persamaan (3.1) harus diubah terlebih dahulu ke bentuk

$$F(s) = \frac{G}{(T_1 s + 1)(T_2 s + 1)} \quad \text{.....(3.8)}$$

Mulanya bagian penyebut diakarkan menjadi

$$F(s) = \frac{3{,}347 \times 10^4}{s^2 + 494s + 1{,}084 \times 10^4}$$

$$F(s) = \frac{3{,}347 \times 10^4}{(s + 471)(s + 23)}$$

$$F(s) = \frac{3{,}347 \times 10^4}{(471)(23)(\frac{1}{471}s + 1)(\frac{1}{23}s + 1)}$$

$$F(s) = \frac{3{,}0896}{(\frac{1}{471}s + 1)(\frac{1}{23}s + 1)} \quad \text{.....(3.9)}$$

$$T_1 = \frac{1}{471} = 2{,}1231 \times 10^{-3}$$

$$T_2 = \frac{1}{23} = 0{,}04348$$

$$T_1 < T_2$$

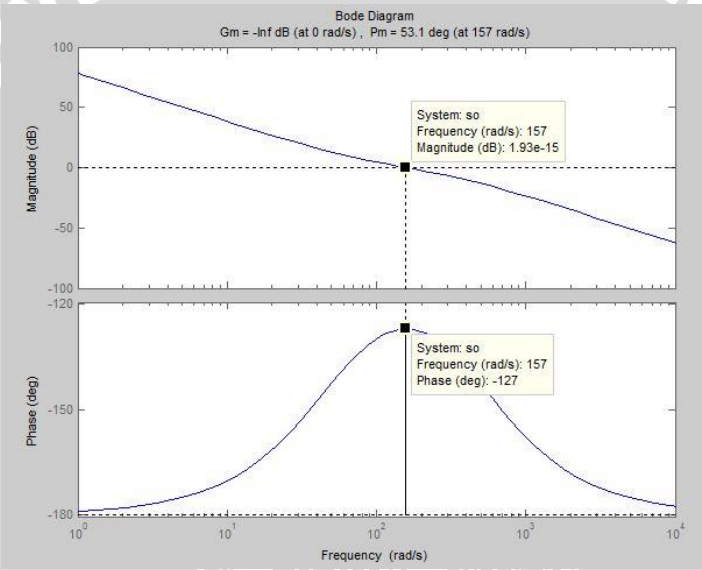
Maka nilai Gm adalah 3,0896 dan Tm= T_1. Nilai Tmn ditetapkan samadengan 0,001. Dengan menentukan mengubah nilai faktor redaman (D) maka dapat ditentukan nilai Gc dan Ti melalui persamaan berikut

$$Gc = \frac{1}{a\ Gm}\ \frac{Tmn}{Tm} \quad \text{.....(3.10)}$$

$$T_i = a^2\ Tma > 1 \quad \text{.....(3.11)}$$

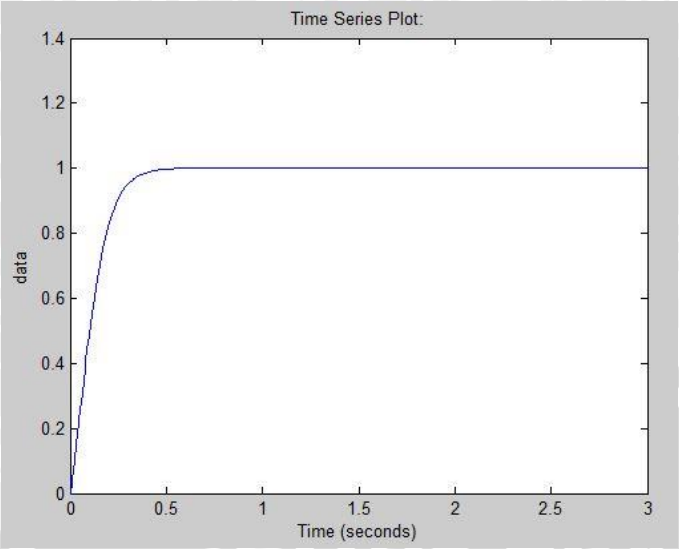
$$a = 2D + 1 \quad \text{.....(3.12)}$$

Nilai-nilai dari variabel diatas dimasukkan kedalam Matlab dan ditampilkan dalam bodeplot untuk mengetahui bahwa *frequency crossover* pada margin terletak pada fasa maksimumnya.



Gambar 3.10 Hasil Bodeplot dengan metode Symmetrical Optimum

Dengan nilai D=1 didapatkan Gcw=0,0508 dan Tcw=0,0191. Sehingga nilai Kp=0,0508 dan Ki= 2,6595. Nilai Kp dan Ki tersebut disimulasikan pada *toolbox* Simulink untuk mengetahui respon plant dengan kontroler PI jika masukan berupa sinyal unit *step* maka didapatkan grafik sebagai berikut.



Gambar 3.11 Respon keluaran plant dengan kontroler

Besarnya nilai settling time setelah diberi kontroler adalah 0,5s dan mampu mencapai *setpoint*.

Setelah nilai Kp dan Ki didapatkan, didapatkanlah fungsi keseluruhan dari PI kontroler dapat dilihat pada persamaan 3.13.

$$F(s) = \frac{s^3 + 494s^2 + 12540s + 89013}{s^3 + 494s^2 + 10840s} \quad \text{.....(3.13)}$$

Setelah didapatkan fungsi keseluruhan dari PI kontroler, didapatkan nilai pole sebesar -471 dan -23. Nilai *pole* ini yang nantinya akan digunakan dalam perancangan kontroler *pole placement*.

3.9 Perancangan Algoritma

Dalam merancang perangkat lunak, hal pertama yang dilakukan adalah mengetahui karakteristik motor DC. Setelah itu, dilakukan penghitungan untuk menentukan algoritma kontroler dengan *Output Feedback Control*. Setelah didapatkan algoritma, kemudian dibuat program yang akan digunakan dalam mikrokontroler dengan software program arduino. Perancangan algoritma tersebut melalui beberapa tahap :

1. Mencari matriks A,B,C dan D.
2. Mencari matriks K (matriks umpan balik).
3. Mencari matriks M (pra filter).
4. Pembuatan simulasi pada software matlab.
5. Penyusunan flowhart yang algoritmanya diterapkan pada mikrokontroler.
6. Implementasi pada alat.

3.10 Desain Kontroler Struktur *Output Feedback Control*

Dalam mendesain kontroler struktur *Output Feedback Control* diperlukan nilai dari matriks A,B,C,D,K dan M. Dimana matriks A adalah matriks keadaan orde n x n, B adalah matriks masukan orde n x m, C adalah matriks keluaran orde p x n, D adalah matriks transimisi langsung orde p x n, K adalah matriks umpan balik dan M adalah matriks pra filter. Langkah – langkah yang dilakukan untuk mencari matriks – matriks tersebut adalah sebagai berikut :

1. Fungsi alih dari motor DC yang didapatkan dari pemodelan digunakan pertama – tama untuk mencari matriks A,B,C dan D dengan menggunakan *state space*.

$$\frac{Y(s)}{U(s)} = \frac{33470}{s^2 + 494s + 10840}$$

Fungsi alih dari motor DC tersebut dimasukkan kedalam matlab dengan menggunakan persamaan yaitu :

```
>> num=[33470];
>> den=[1 494 (1.084*(10^4))];
>> [A,B,C,D]=tf2ss(num,den)
```

Setelah itu didapatkan matriks:

$$A = \begin{bmatrix} -494 & 10840 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [1 \quad 33470]$$

$$D = [0]$$

2. Setelah diketahui matriks A, B, C dan D dari plant selanjutnya adalah menentukan pole dari kontroler untuk mendapatkan nilai matriks umpan balik K. Matriks umpan balik K bisa didapatkan dengan menggunakan persamaan pada matlab yaitu:

```
K=place(A,B,p);
```

dimana:

A = matriks A, matriks orde 2x2

B = matriks B, matriks orde 2x1

p = pole kontroler yang nilainya bebas, matriks orde 2x1

3. Setelah mendapatkan nilai matriks K, langkah selanjutnya adalah mencari matriks *pra filter* M menggunakan persamaan pada matlab yaitu:

```
m=C*inv(-Ad)*B
```

dimana:

m = matriks M

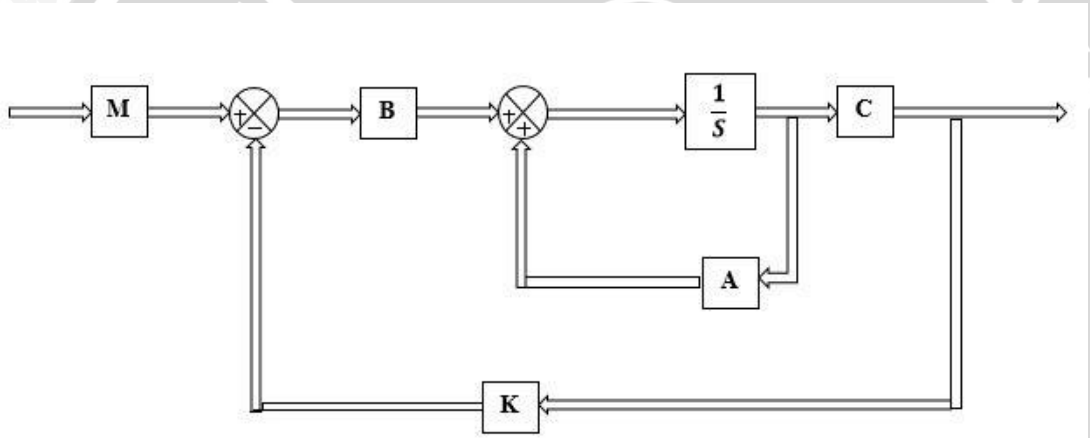
C = matriks C

Ad = matriks hasil dari (A-BK)x(t)

B = matriks B, matriks orde 2x1

3.11 Diagram Blok Perancangan Desain

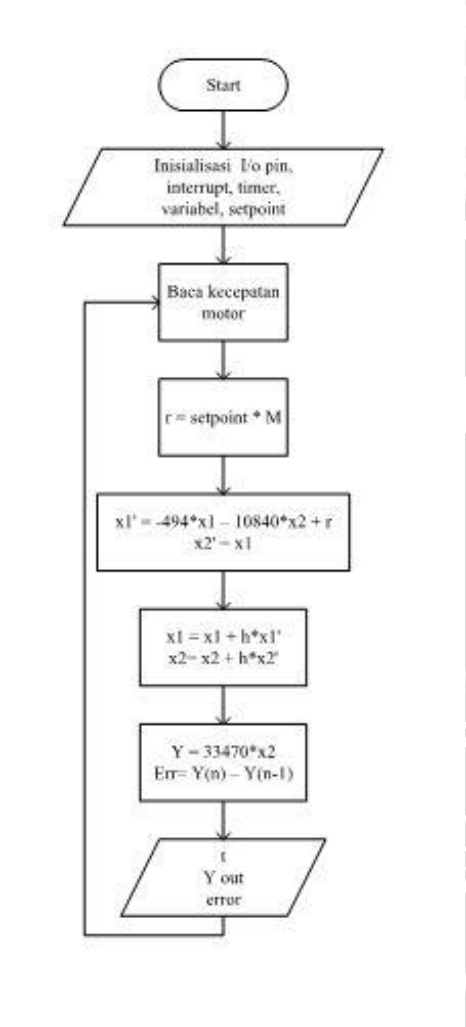
Setelah dilakukan perancangan, didapatkan blok perancangan desain ditunjukkan pada gambar 3.12.



Gambar 3.12 Diagram Blok Perancangan Desain

3.12 *Flowchart* Program

Flowchart program merupakan alur proses program yang dilakukan oleh kontroler pada saat implementasi. *Flowchart* program ditunjukkan pada Gambar 3.13.



Gambar 3.13 Flowchart program

BAB IV
HASIL DAN PEMBAHASAN

Hasil dan pembahasan dilakukan dengan melakukan simulasi dan menganalisis sistem serta melakukan pengujian keseluruhan sistem. Simulasi dan analisis sistem dilakukan untuk mengetahui apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan skema pembuatan. Pengujian keseluruhan sistem dilakukan untuk menguji apakah parameter yang sudah didapatkan dapat diaplikasikan pada alat. Pengujian dibagi menjadi bebarapa bagian, yaitu:

1. Simulasi *Pole Placement* menggunakan MATLAB.
2. Pengujian keseluruhan sistem.

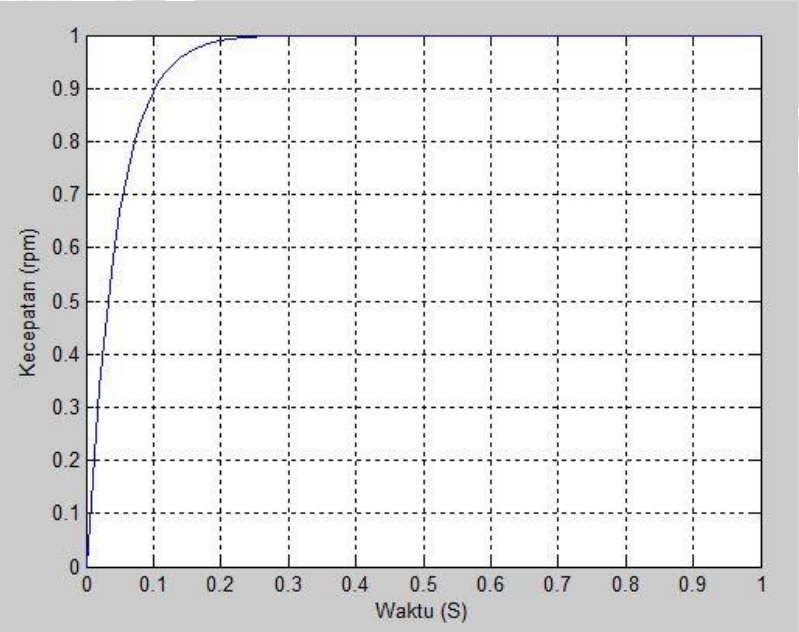
4.1 Simulasi *Pole Placement* Menggunakan MATLAB

Simulasi *pole placemenet* pada matlab diperlukan agar dapat mengetahui hasil dari *state space* fungsi alih motor yang telah diaplikasikan kedalam metode *pole placement.* Simulasi menggunakan software matlab dengan sintaks:

```
A=[-494   -10840; 1 0];
B=[1;0];
C=[0 33470];
D=[0];

t=[0: 0.1:5];
figure(1);
sys1=ss(A,B,C,D);
y=step(sys1,t);
plot (t,y), grid;% Gambar sistem tanpa pole

p=[ -471; -23];
K=place(A,B,p)

Ad=A-B*K;
t=[0:0.01:1];

y1=step(Ad,B,C,D,1,t);
figure (3);
plot(t,y1), grid % gambar untuk sistem dengan pole


[a,b]=ss2tf(Ad,B,C,D);
%figure(3)
%t=(0:0.1:7)
y2=step(a,b,t);
```

```
m=C*inv(-Ad)*B
r=6000
e=0
yc=(r+e)*y2/m
figure (2)
plot(t,yc), grid
```

A, B, C dan D merupakan matriks yang didapatkan dari perubahan fungsi alih motor kedalam *state space*. Hasil dari simulasi *pole placement* pada matlab ditunjukkan pada Gambar 4.1.

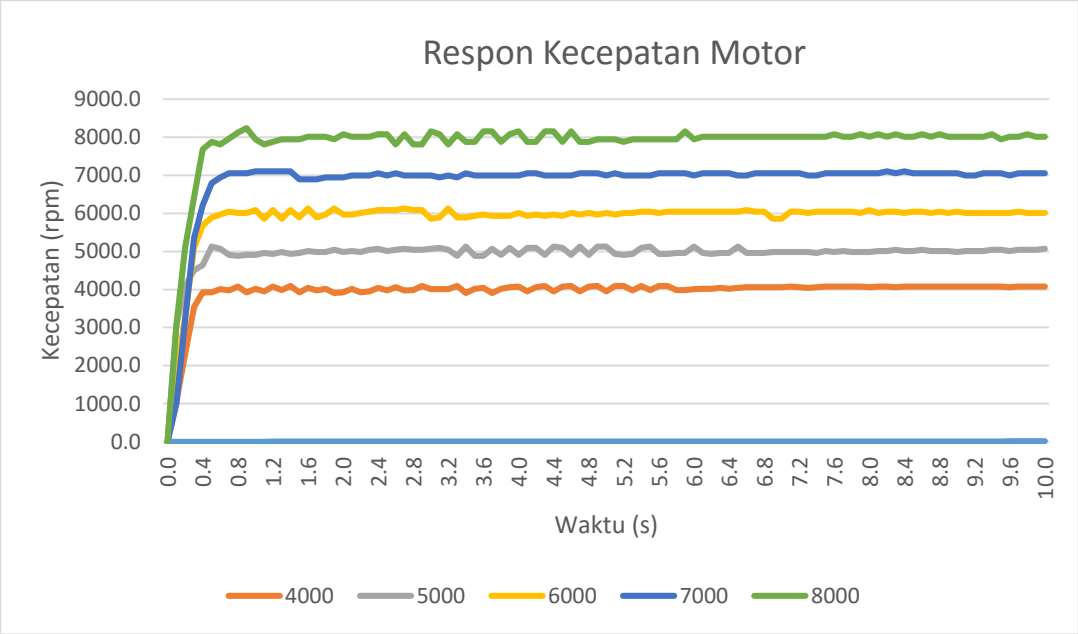


Gambar 4.1 Hasil Simulasi Pole Placement Matlab

4.2 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem bertujuan untuk mengetahui respon motor ketika diberikan metode *pole placement*. Data dengan nilai *pole* -471, -23 dan dengan nilai M = 3.0896 akan diimplementasikan pada alat. Implementasi dengan mengubah – ubah setpoint ini dilakukan supaya kita mengetahui bagaimana respon *output* kecepatan sistem apabila diberikan dengan nilai yang berubah – ubah.

Setpoint diberikan pada kecepatan 4000, 5000, 6000, 7000 dan 8000. Grafik hasil dari pengujian keseluruhan sistem tanpa gangguan ditunjukkan pada Gambar 4.2.



Gambar 4.2 Respon Kecepatan Motor

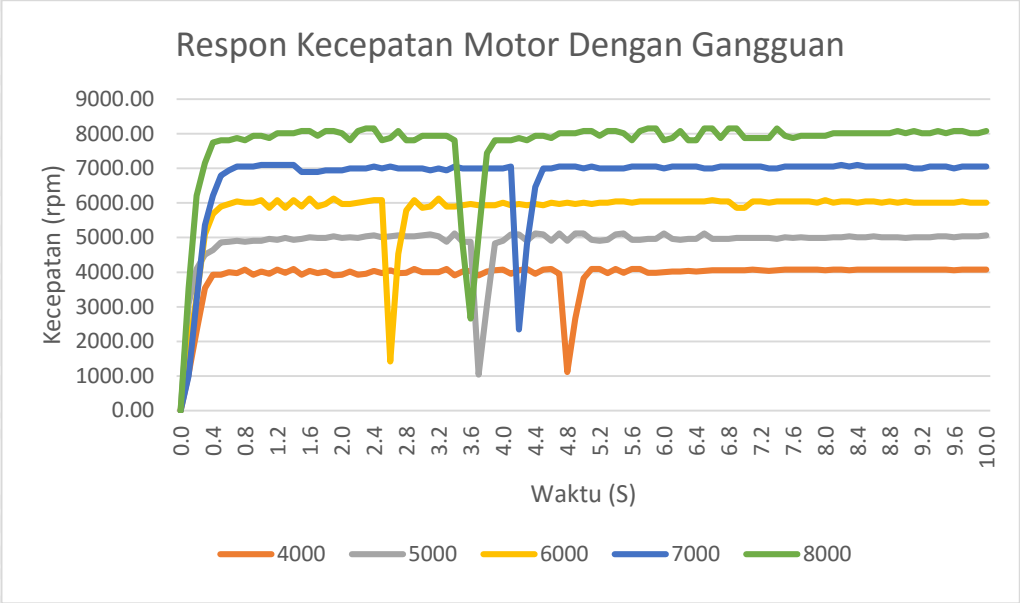
Dari grafik *output* respon kecepatan motor DC dapat dilihat bahwa kontroler *pole placement* mampu mengendalikan kecepatan motor pada setiap *setpoint*. Nilai e*rror steady state* pada *setpoint* 4000 sampai 8000 sebesar 1,9% sampai 2,28%. Untuk hasil pada Gambar 4.2 dirangkum pada tabel 5.1

Tabel 4.1 Rekapitulasi Analisis Grafik Respon Kecepatan Motor

Setpoint	ts	td	tr	tp	mp
4000	0,8 s	0,2 s	0,3 s	0,5 s	5,80%
5000	1 s	0,2 s	0,4 s	0,6 s	2,60%
6000	0,6 s	0,3 s	0,5 s	1 s	2,50%
7000	0,7 s	0,3 s	0,4 s	1 s	1,45%
8000	1,1 s	0,2 s	0,6 s	0,9 s	3,02%

4.3 Pengujian Keseluruhan Sistem Dengan Gangguan

Pengujian keseluruhan sistem dengan gangguan bertujuan untuk mengetahui bagaimana respon motor ketika diberikan gangguan ketika sedang bekerja. Setpoint diberikan pada kecepatan 4000, 5000, 6000, 7000 dan 8000. Grafik hasil pengujian keseluruhan sistem dengan gangguan dapat dilihat pada Gambar 4.3.

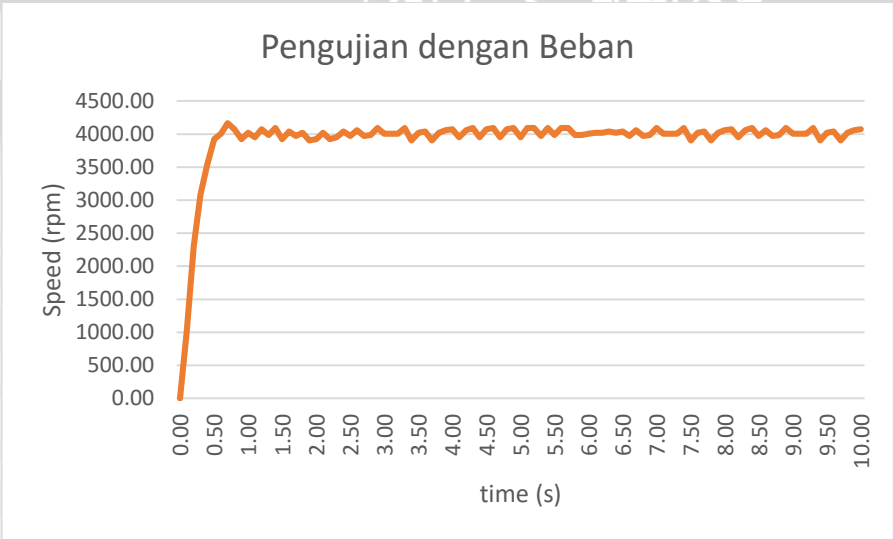


Gambar 4.3 Respon Kecepatan Motor Dengan Gangguan

Dari grafik respon kecepatan motor dengan gangguan dapat dilihat, bahwa waktu yang dibutuhkan sistem untuk mencapai *steady state* adalah 0,3 dan 0,4 detik tanpa *overshoot* dengan nilai *error steady state* sebesar 0,5% sampai 2,44%.

4.4 Pengujian Sistem dengan beban cakram padat

Salah satu aplikasi dari motor DC tipe RF-130CH adalah sebagai pemutar disk pada CD/DVD player. Beban berupa cakram padat dengan ukuran diameter 120 mm dan berat 15-20 mg. Hasil pengujian dengan setpoint 4000 rpm diberikan oleh gambar 4.4.



Gambar 4. 4 Hasil Pengujian dengan beban Disk

Berdasarkan grafik respon yang diberikan maka dapat diketahui waktu yang dibutuhkan sistem untuk mencapai *steady state* adalah 1.1 detik dengan *overshoot* 4.1 % serta nilai *error steady state* sebesar 2,28%.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Parameter PI yang didapatkan menggunakan metode *symmetrical optimum* digunakan untuk mendapatkan nilai pole dari motor DC. Langkah awal yang dilakukan dengan mengubah nilai fungsi alih dari motor DC ke bentuk $F(s) = \frac{G}{(T_1 s+1)(T_2 s+1)}$ yang dimana akan didapatkan nilai Gm. Variabel-variabel tersebut dimasukkan kedalam matlab dan ditampilkan dalam *bodeplot* untuk mengetahui *frequency crossover* pada margin terletak pada fasa maksimumnya. Dengan metode *symmetrical optimum*, didapatkan nilai Kp = 0,0508 dan nilai Ki = 2,6595.
2. Simulasi output motor DC setelah diberi kontroler *pole placement* mempunyai *settling time* 0,3 detik tanpa *error steady state* dan tanpa *overshoot*. Hasil dari implementasi tanpa gangguan mempunyai *settling time* sebesar 0,6 detik sampai 1 detik dengan *error steady state* sebesar 1,9% sampai 2,28%. Sedangkan implementasi dengan gangguan, waktu yang dibutuhkan untuk mencapai *steady state* kembali setelah diberi gangguan bernilai 0,3 detik sampai 0,4 detik tanpa *overshoot* dan *error steady state* sebesar 0,5% sampai 2,44%.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Motor DC yang digunakan dapat diganti dengan yang lain seperti BLDC motor atau motor DC *shunt*.
2. Rotary encoder yang digunakan bisa diganti dengan hall sensor untuk lebih baik dalam sensing kecepatan.
3. Sinyal PWM yang berasal dari Arduino lebih baik menggunakan *mode* 16-bit daripada 8-bit untuk memperkecil *error steady state* karena lebih teliti dalam pengaturan *dutycycle*.

UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Ogata, K. 1997. *Teknik Kontrol Automatik*. terjemahan: Edi Laksono Ir. Jakarta : Penerbit Erlangga.

Ogata, K. 2010. *Modern Control Engineering*, Pearson Education, Inc,, publishing as Prentice Hall, One Lake Street, Upper Saddle River, New Jersey 07458. Fifth edition.

Soemarwanto. 2010. *Dasar Konversi Energi Elektrik.* Malang

Levine, William S. 1996. *The Control Handbook.* US :CRC Press and IEEE Press.

Faisol. M. Arif. 2015. *Sistem Kontrol Kecepatan Motor DC D-6759 Berbasih Arduino Mega 2560*. Malang : Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.

Nugroho, Ari. 2013. *Penstabilan Sistem Melalui Umpan Balik Keadaan Proporsional-Integral*. Semarang : Skripsi. Fakultas Sains dan Matematika. Universitas Diponegoro.

Mizera, Roman. 2005. *Modification Of Symmetric Optimum Methode*. Seminar, Instruments and Control. XXX : 319-320.

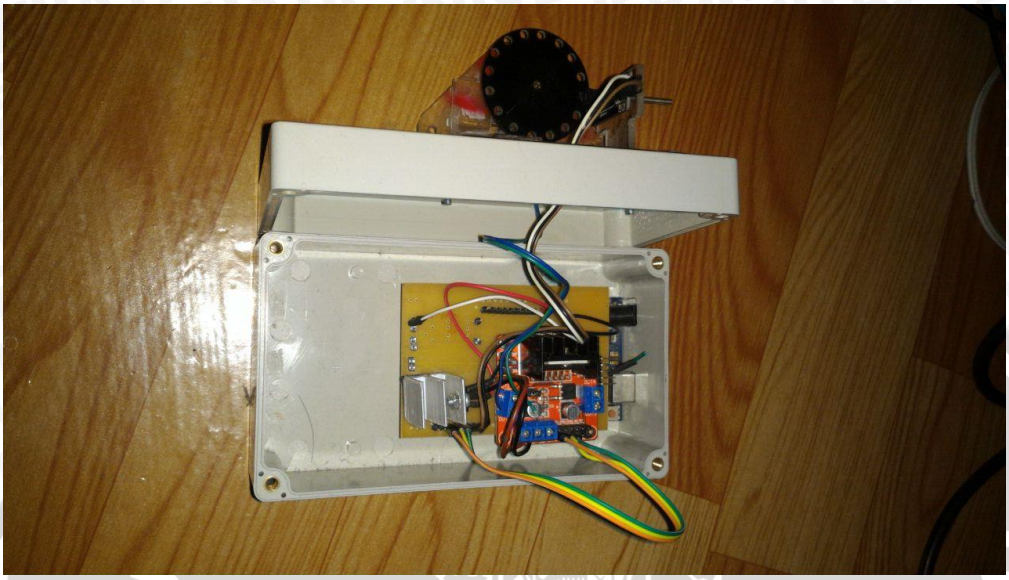
Prayogo, R. 2014. *Aplikasi Kontrol Optimal LQG untuk Pengontrolan Water Level Steam Drum Boiler*. Malang : Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.

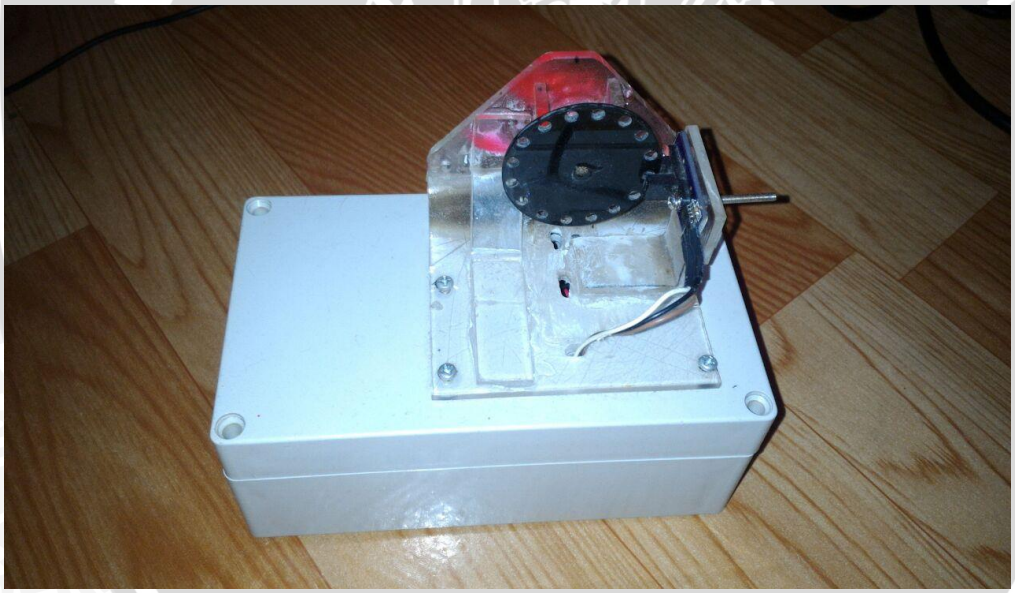
Amalia, Zakiyah. 2016. *Desain Kontroler Struktur Output Feedback Control Dengan Pole Placement Pada Pengontrolan Tegangan Output Generator DC*. Malang : Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.

LAMPIRAN I

FOTO ALAT

FOTO ALAT





LAMPIRAN II

LISTING PROGRAM

PROGRAM UTAMA

```
#define sensor 8

int a=0, Sampling, Value1, Value2, Rpm; // Read RPM Motor

double err, t, x1=0, x1_new, x2=0, x2_new, r, y; // U/ Pole Placement
double M = 0.3296, setpoint = 4000, h = 0.1, duty;


void setup() {
 // put your setup code here, to run once:
 noInterrupts();

 //Untuk Motor use L298N
 pinMode(6, OUTPUT);//enA ==>PWM
 pinMode(5, OUTPUT);//in1
 pinMode(4, OUTPUT);//in2

 digitalWrite(5, HIGH);
 digitalWrite(4, LOW);


 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);
 TCCR1B=(0<<ICNC1) | (1<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) |
(1<<CS11) | (1<<CS10);

 attachInterrupt(0, BacaRpm, RISING);
 Serial.begin(9600); // serial monitor

 interrupts();
```

```
}

void loop() {
 // put your main code here, to run repeatedly:
  r = setpoint * M;
  err = y - Rpm;

  x1_new = - 494 * x1 - 10833 * x2 + r;
  x2_new = x1;

  x1 = x1 + x1_new * h;
  x2 = x2 + x2_new * h;

  y = 33470 * x2;

  duty = map (y, 0, 12847, 0, 255);
  analogWrite(6,duty);
  serialdisplay();
}

 void BacaRpm(){

   if (a == 0){
      Value1 = TCNT1;
      a = 1;
    }
   else{

      Value2 = TCNT1;
      if (Value2 > Value1){
       Sampling = Value2 - Value1;
        }
      else if (Value1 > Value2){
```

```
        Sampling = (65535 - Value1) + Value2;
        }
      a=0;
      Rpm = 250000/Sampling; //clock timer1 diset 250KHz
      Rpm = Rpm * 60 / 16;
    }
  }

void serialdisplay(){
      Serial.print(millis()); // t dalam ms
      Serial.print("\t");
      Serial.println(Rpm); // kecepatan aktual motor
}
```

LAMPIRAN III

DATASHEET MOTOR

RF-130CH

MABUCHI MOTOR

Precious metal-brush motors

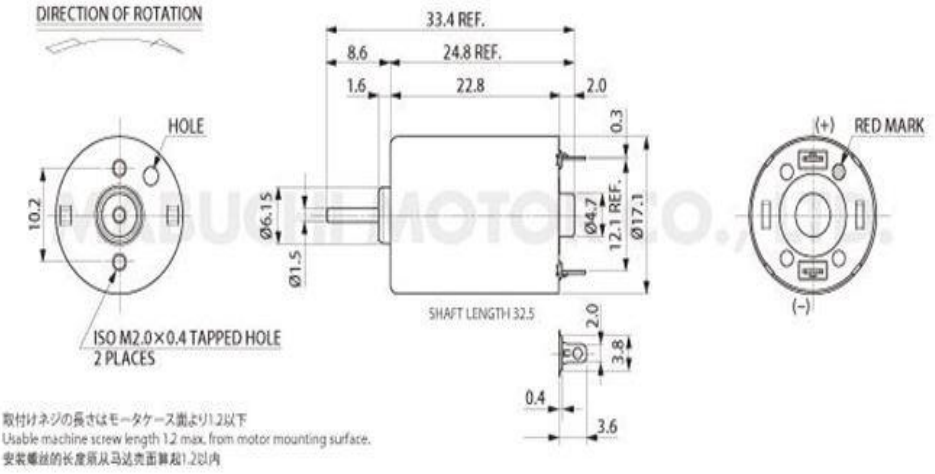
OUTPUT:0.03W-2.5W (APPROX)

WEIGHT: 18g (APPROX)

Typical Applications : Home Appliances

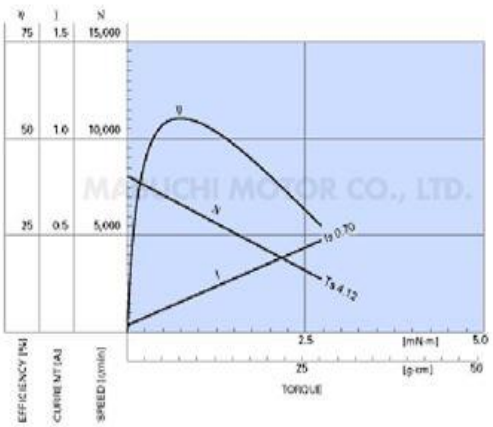
*By clicking the "MODEL", you can display the Performance Chart Simulation.

MODEL		VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
		OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			V	r/min	A	r/min	A	mN·m	g·cm	W	mN·m	g·cm	A
RF-130CH	12250	2 - 6	6	8100	0.036	6600	0.16	0.76	7.8	0.53	4.12	42	0.70



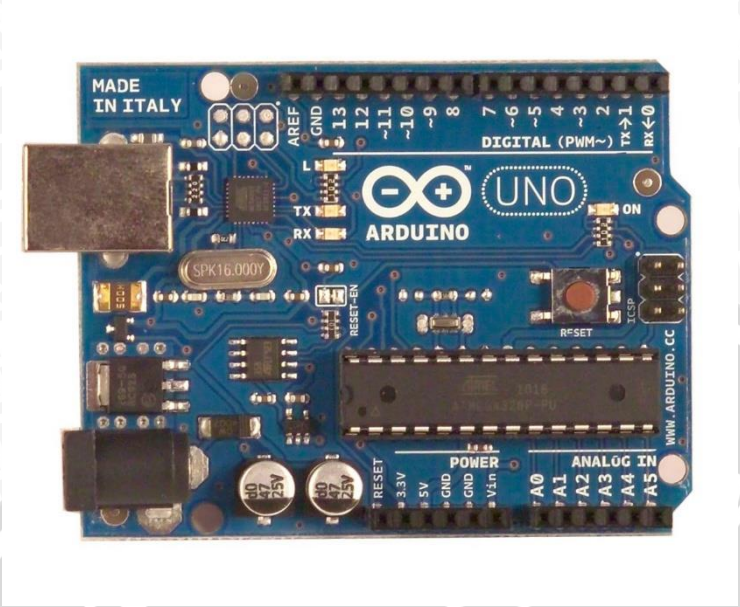
UNIT: MILLIMETERS

RF-130CH-12250 6.0V



LAMPIRAN IV

DATASHEET ARDUINO UNO



Technical Specifications

Terms & Conditions

Enviromental Policies

half sqm of green via Impatto Zero®

Arduino UNO

The Arduino Uno is a microcontrol er board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscil ator, a

USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to

support the microcontrol er; simply connect it to a computer with a USB cable or power it with a AC-to-DC

adapter or battery to get started. The Uno differs from al preceding boards in that it does not use the FTDI

USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

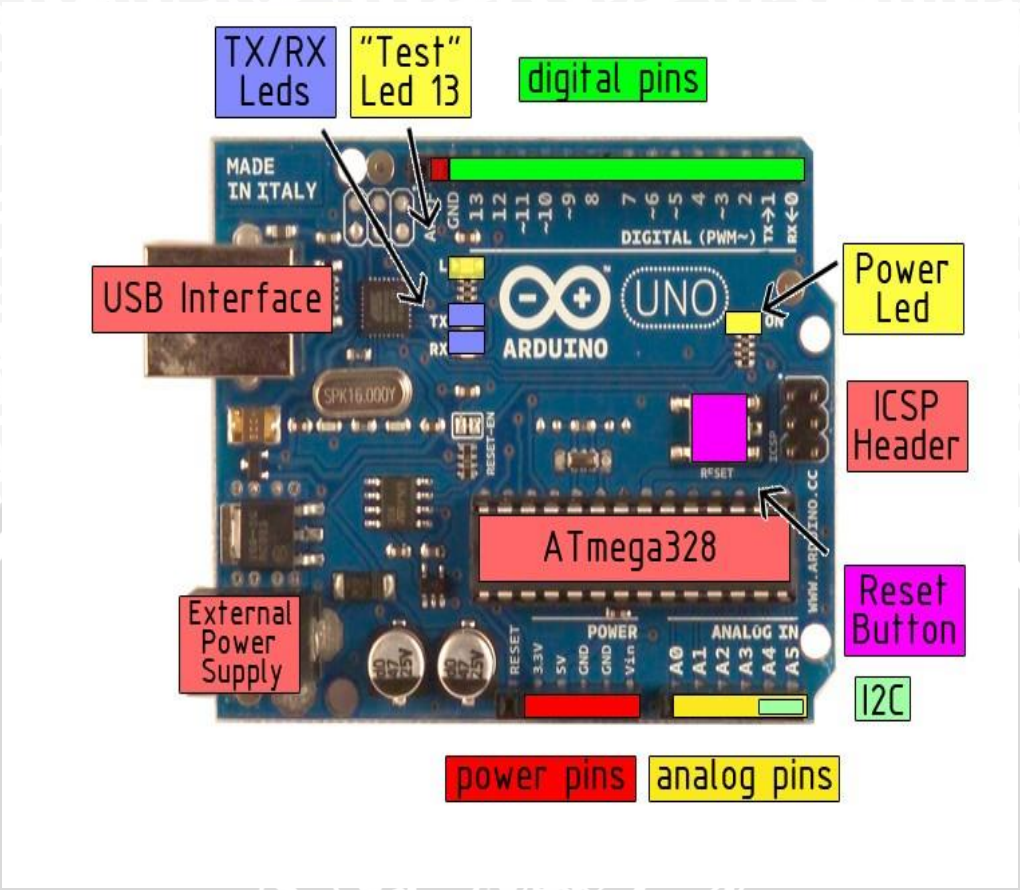
"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version

1.0 wil be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB

Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions,

see the index of Arduino boards.

the board



Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Microcontrol er

ATmega328

Operating Voltage

5V

Input Voltage (recommended) 7-12V

Input Voltage (limits)

6-20V

Digital I/O Pins

14 (of which 6 provide PWM output)

Analog Input Pins

6	
DC Current per I/O Pin	
40 mA	
DC Current for 3.3V Pin	
50 mA	
32 KB of which 0.5 KB used by	
Flash Memory	
bootloader	
SRAM	
2 KB	
EEPROM	
1 KB	
Clock Speed	
16 MHz	

Input and Output

The Arduino Uno can be powered via the USB connection or with an external power supply. The power

source is selected automatical y.

External (non-USB) power can come either from an AC-to-DC adapter (wal -wart) or battery. The adapter

can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a

battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V

pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage

regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as fol ows:

•

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to

5 volts from the USB connection or other regulated power source). You can supply voltage through

this pin, or, if supplying voltage via the power jack, access it through this pin.

•

5V. The regulated power supply used to power the microcontrol er and other components on the

board. This can come either from VIN via an on-board regulator, or be supplied by USB or another

regulated 5V supply.

•

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

•

GND. Ground pins.

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It

has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and

digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pul -up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have

specialized functions:

•

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are

connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .

- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or fal ing edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additional y, some pins have specialized functionality:

- **I2C: 4 (SDA) and 5 (SCL).** Support I2C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

-

AREF. Reference voltage for the analog inputs. Used with analogReference().

•

Reset. Bring this line LOW to reset the microcontrol er. Typical y used to add a reset button to

shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other

microcontrol ers. The ATmega328 provides UART TTL (5V) serial communication, which is available on

digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB

and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB

COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which al ows simple textual data to be sent to and from the

Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-

serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library al ows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire

library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/

ATmega328" from the **Tools > Board** menu (according to the microcontrol er on your board). For details,

see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that al ows you to upload new code

to it without the use of an external hardware programmer. It communicates using the original STK500

protocol (reference, C header files).

You can also bypass the bootloader and program the microcontrol er through the ICSP (In-Circuit Serial

Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader,

which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and

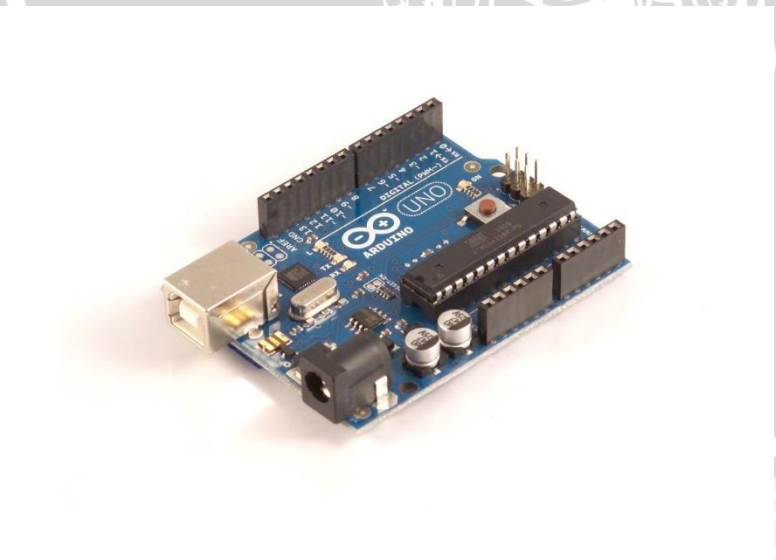
then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer

(overwriting the DFU bootloader).

Automatic (Software) Reset

USB Overcurrent Protection

Physical Characteristics



Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a

way that al ows it to be reset by software running on a connected computer. One of the hardware flow control

lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad

capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The

Arduino software uses this capability to al ow you to upload code by simply pressing the upload button in the

Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR

can be wel -coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or

Linux, it resets each time a connection is made to it from software (via USB). For the fol owing half-second or

so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything

besides an upload of new code), it will intercept the first few bytes of data sent to the board after a

connection is opened. If a sketch running on the board receives one-time configuration or other data when it

first starts, make sure that the software with which it communicates waits a second after opening the

connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can

be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset

by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and

overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer

of protection. If more than 500 mA is applied to the USB port, the fuse will automatical y break the connection

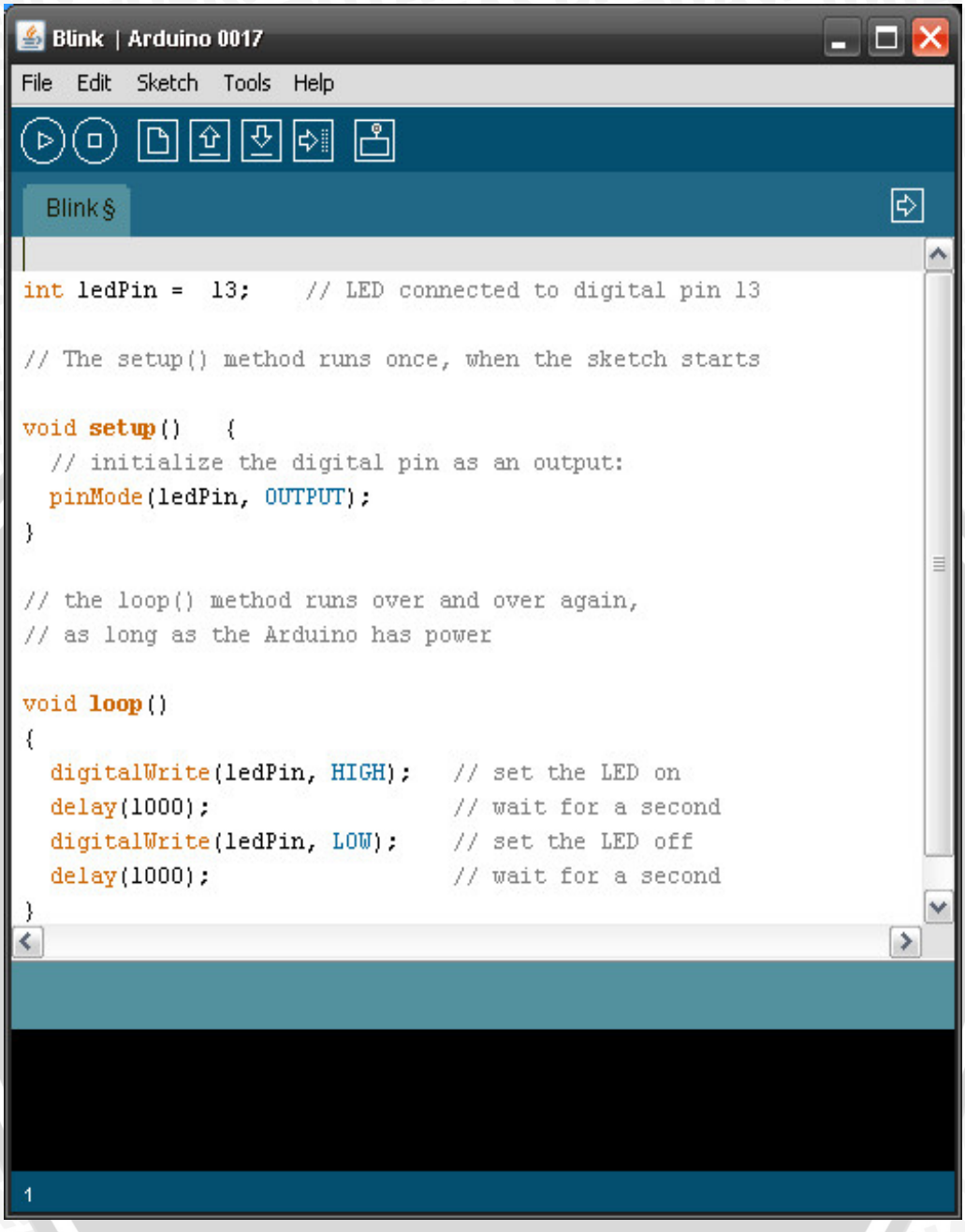
until the short or overload is removed.

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector

and power jack extending beyond the former dimension. Three screw holes al ow the board to be attached to

a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple

of the 100 mil spacing of the other pins.



Arduino can sense the environment by receiving input from a variety of sensors and can affect its

surroundings by control ing lights, motors, and other actuators. The microcontrol er on the board is

programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can

communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'l have to fol ow different instructions for your personal

OS. Check on the Arduino site for the latest instructions.
http://arduino.cc/en/Guide/HomePage

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Now you're actual y ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the wel known programming "hel o world", select

File>Sketchbook>

Arduino-0017>Examples>

Digital>Blink

Once you have your skecth you'll see something very close to the screenshot on the right.

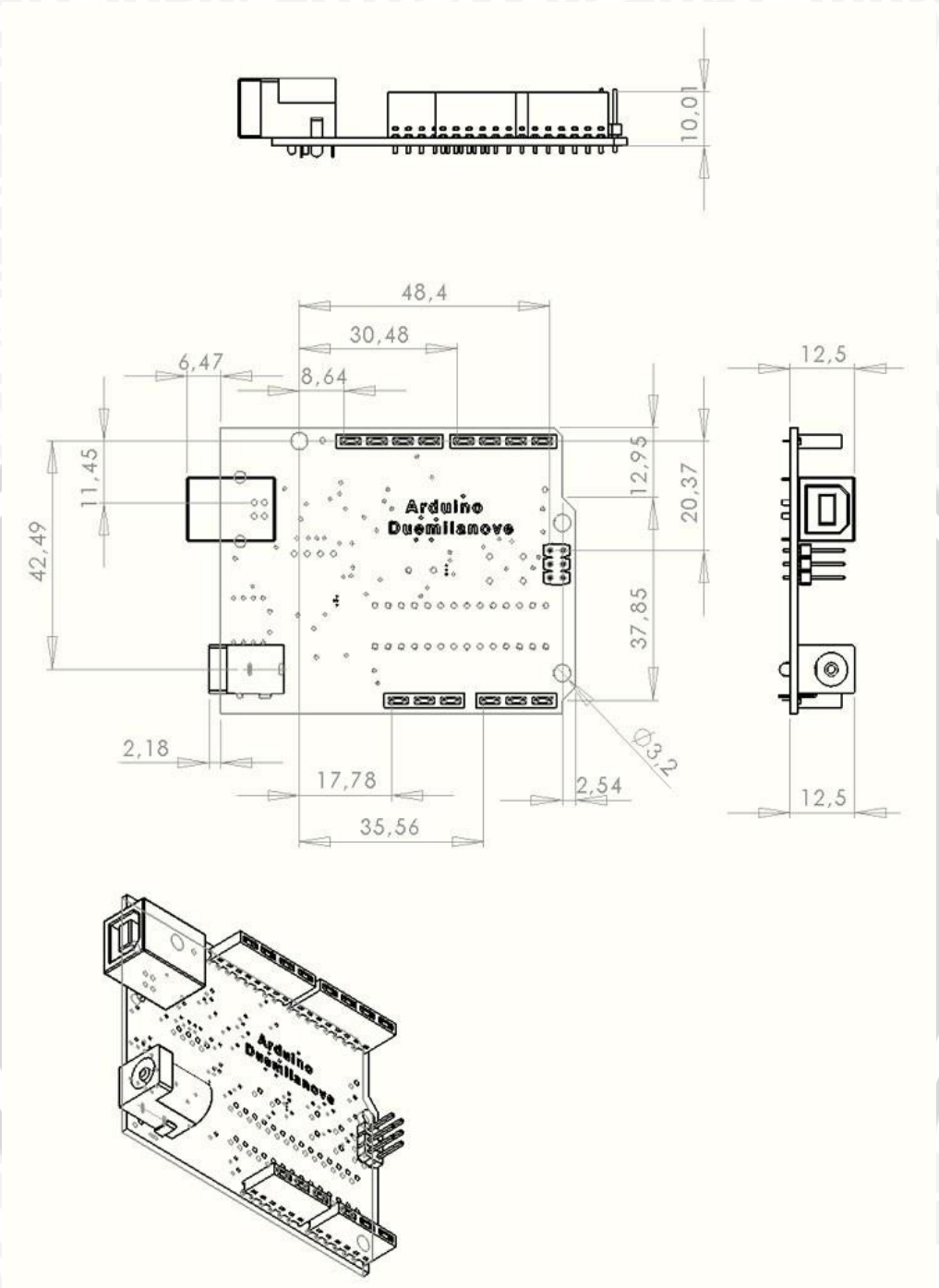
In **Tools>Board** select

Now you have to go to

Tools>SerialPort

and select the right serial port, the one arduino is attached to.

Dimensioned Drawing



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The

producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing,

or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from

Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems

necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the

producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER

WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the

products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other

services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth

above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino・ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause

severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the

operation of nuclear facilities and weapons systems. Arduino・ products are neither designed nor intended for use in military or aerospace applications or

environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino・ products which is solely

at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its

products and any use of Arduino・ products in Customer's applications, notwithstanding any applications-related information or support that may be

provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages,

liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this

terms and conditions or (i) the gross negligence or willful misconduct by the Customer.

3. **Consequential Damages Waiver**

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or

exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the

possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or

characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is

subject to change without notice. Do not finalize a design with this information.

The producer of Arduino・ has joined the Impatto Zero®

policy of LifeGate.it. For each Arduino board produced is

created / looked after half squared Km of Costa Rica's

forest's.