

BAB II

TINJAUAN PUSTAKA

Rotary Inverted Pendulum terdiri dari kontroler, lengan/*link*, pendulum, sebuah aktuator (motor dc) dan sebuah encoder yang berputar. Kontroler membuat pendulum berdiri tegak pada perputaran lengan dengan menggerakkan lengan. Motor menyediakan kekuatan untuk memutar lengan. *Rotary encoder* mendeteksi posisi sudut pendulum dan lengan.

2.1 Penelitian Terdahulu

Penelitian *Rotary Inverted Pendulum* pernah dilakukan sebelumnya. Pada penelitian Indra, 2011, komponen *Rotary Inverted* Pendulum terdiri dari kontroler yang membuat pendulum berdiri tegak pada perputaran lengan dengan menggerakkan lengan. Motor menyediakan kekuatan untuk memutar lengan. *Rotary encoder* mendeteksi posisi sudut pendulum dan lengan.

Sistem yang dijadikan obyek penelitian adalah sistem *Rotary Inverted* Pendulum yang dapat mempertahankan kesetimbangan pada posisi ekuilibrium dengan atau tanpa gangguan. Kontrol yang digunakan adalah kontrol PID yang mampu mempertahankan posisi kesetimbangan tongkat pendulum untuk tetap tegak ke atas (*upright position*).

Pada penelitian yang dilakukan oleh Nusantoro, 2012, pengembangan dilakukan pada bagian mekaniknya. Mekanik yang semula terbuat dari kayu dan mika seperti pada Gambar 2.1, sekarang terbuat dari besi dan aluminium seperti pada Gambar 2.2.



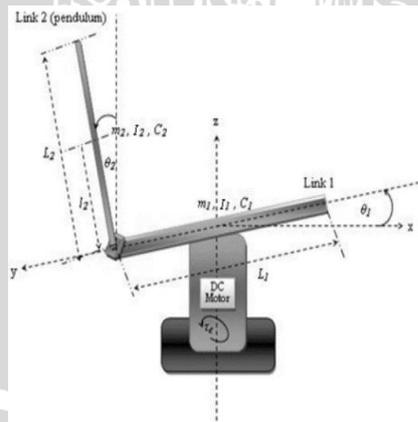
Gambar 2.1 Mekanik pada penelitian yang dilakukan Ravi Indra (Indra, 2011)



Gambar 2.2 Mekanik pada penelitian Goegoes dkk (Nusantoro, 2012)

2.2 Sistem Rotary Inverted Pendulum (RIP)

Sebuah skema pendulum terbalik ditunjukkan pada Gambar 2.3.



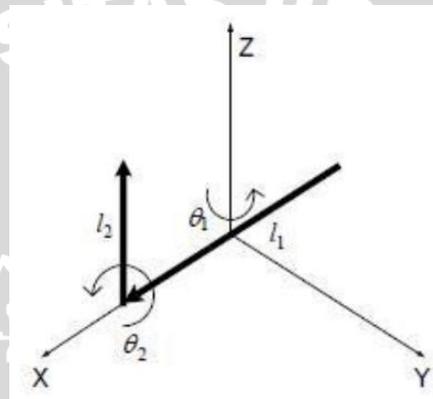
Gambar 2.3 Skema pendulum terbalik (Stepahine, 2010)

Motor DC digunakan untuk menerapkan torsi τ_e ke lengan 1. *Link* antara lengan 1 dan lengan 2 tidak digerakkan tetapi bebas untuk berputar. Kedua lengan memiliki panjang L_1 dan L_2 . Lengan memiliki massa m_1 dan m_2 yang

berlokasi masing-masing di l_1 dan l_2 , yang merupakan panjang dari sudut rotasi lengan pusat massa. Lengan memiliki momen inersia I_1 dan I_2 . Setiap rotasi sendi teredam dengan koefisien redaman C_1 dan C_2 , dimana C_1 adalah redaman disediakan oleh bantalan motor dan C_2 adalah redaman yang timbul dari kopel antara lengan 1 dan lengan 2.

2.2.1 Model Fisika

Berikut adalah Gambar 2.4 skematik dari pendulum terbalik beserta arah pergerakannya.



Gambar 2.4 Skematik dari pendulum terbalik beserta arah pergerakannya (Stephanie, 2010)

Kecepatan angular dari masing - masing *link* diberikan oleh :

$$\overline{\omega}_1 = \dot{\theta}_1 \hat{k}$$

$$\overline{\omega}_2 = -\dot{\theta}_1 \cos \theta_2 \hat{i} - \dot{\theta}_1 \sin \theta_2 \hat{j} + \dot{\theta}_2 \hat{k}$$

Kecepatan linier dari masing - masing *link* diberikan oleh :

$$\overline{v}_{1c} = \dot{\theta}_1 l_1 \hat{j}$$

$$\overline{v}_{2c} = \dot{\theta}_1 l_1 \sin \theta_2 \hat{i} + (\dot{\theta}_2 l_2 + \dot{\theta}_1 l_1 \cos \theta_2) \hat{j} - l_2 \dot{\theta}_1 \sin \theta_2 \hat{k}$$

Torsi dari Motor DC

Motor DC permanen menghasilkan gaya gerak listrik E_a . Dimana E_a berbanding lurus dengan kecepatan rotor yang dinyatakan sebagai berikut :

$$E_a = K_v \omega_\tau = K_v \dot{\theta}_1 \quad (2.1)$$

Torsi yang dihasilkan oleh motor DC berbanding lurus dengan arus rotor dinyatakan sebagai berikut :

$$\tau_e = K_\tau I_a = K_v I_a \quad (2.2)$$

Dengan K_v adalah konstanta untuk tegangan dan K_τ adalah konstanta untuk torsi. Dalam kondisi *steady state*, persamaan yang menggambarkan tegangan motor adalah :

$$V_a = R_a I_a + E_a, \text{ sehingga}$$

$$I_a = \frac{V_a}{R_a} - \frac{E_a}{R_a} \quad (2.3)$$

Substitusikan persamaan 2.3 dan persamaan 2.1 ke persamaan 2.2 ,

$$\tau_e = \frac{K_\tau V_a}{R_a} - \frac{K_\tau K_v \theta_1}{R_a} \quad (2.4)$$

dimana, V_a adalah tegangan dc yang diterapkan. Dua pertimbangan utama dalam memilih motor adalah untuk menciptakan torsi tinggi dan kecepatan tinggi. Torsi diperlukan untuk lengan putar untuk mengubah arah rotasi dengan cepat dalam rangka untuk menjaga keseimbangan pendulum. Dan kecepatan tinggi diperlukan sedemikian hingga lengan dapat bergerak lebih cepat untuk mencegah jatuhnya pendulum.

2.2.2 Model Matematika

Penurunan persamaan matematika yang menjelaskan dinamika dari sistem pendulum terbalik di dasari oleh persamaan Euler Lagrange:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial W}{\partial q_i} = Q_i \quad (2.5)$$

Dimana,

$q_i(t)$ = vektor posisi angular (rad)

$\dot{q}_i(t)$ = vektor kecepatan angular (rad/s)

Q_i = gaya luar (N)

L = Lagrangian (J)

W = energi yang hilang (J)

Dalam persamaan Euler Lagrange, L didefinisikan sebagai ;

$$L(q, \dot{q}) = E_{k \text{ total}} - E_{p \text{ total}} \quad (2.6)$$

dan $q_i = [\theta_1, \theta_2]^T$

Energi kinetik dari *link* 1 adalah :

$$E_{k \text{ link1}} = \frac{1}{2} (\bar{v}_{1c}^T m_1 \bar{v}_{1c} + \bar{\omega}_1^T I_1 \bar{\omega}_1) = \frac{1}{2} m_1 I_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_{1zz} \dot{\theta}_1^2$$

$l_1 = 0$, karena pusat massa dari lengan seimbang pada titik awal, sedemikian hingga

$$E_{k\ link1} = \frac{1}{2} I_1 \dot{\theta}_1^2$$

Energi kinetik dari link 2 dinyatakan dengan persamaan sebagai berikut:

$$\begin{aligned} E_{k\ link2} &= \frac{1}{2} (\bar{v}_{2c}^T m_2 \bar{v}_{2c} + \bar{\omega}_2^T \bar{I}_2 \bar{\omega}_2) \\ &= \frac{1}{2} \dot{\theta}_1^2 (m_2 L_1^2 + \sin^2(\theta_2)(m_2 l_2^2 + I_{2\ yy}) + I_{2\ xx} \cos^2(\theta_2)) \\ &\quad + \frac{1}{2} \dot{\theta}_1^2 (I_{2\ zz} + m_2 l_2^2) + m_2 L_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 \end{aligned}$$

Energi potensial dari link 1 dan link 2 adalah :

$$E_{p\ link1} = 0$$

$$E_{p\ link2} = m_2 g l_2 \cos \theta_2$$

Total energi yang hilang dari sistem adalah jumlahan energi yang hilang dari pendulum dan lengan atau link 1, dinyatakan oleh persamaan berikut :

$$W = \frac{1}{2} C_1 \dot{\theta}_1^2 + \frac{1}{2} C_2 \dot{\theta}_2^2 \quad (2.7)$$

Dari persamaan 2.6, maka Lagrangian didapat :

$$\begin{aligned} L &= \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + \frac{1}{2} m_2 [(L_1 \dot{\theta}_1 + l_2 \dot{\theta}_2 \cos \theta_2)^2 + (l_2 \dot{\theta}_2 \sin \theta_2)^2] \\ &\quad - m_2 g l_2 \cos \theta_2 \end{aligned} \quad (2.8)$$

Persamaan Euler-Lagrange dari masing - masing variabel adalah :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} + \frac{\partial W}{\partial \theta_1} = \tau_e$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} + \frac{\partial W}{\partial \theta_2} = 0$$

Masukkan persamaan 2.4, 2.7, dan 2.8 ke dalam persamaan Euler-Lagrange, sehingga akan didapatkan persamaan non linear dari dinamika pendulum terbalik sebagai berikut :

$$\begin{aligned} (I_1 + m_2 L_1^2) \ddot{\theta}_1 + (m_2 L_1 l_2 \cos \theta_2) \ddot{\theta}_2 + (m_2 L_1 l_2 \sin \theta_2) \dot{\theta}_2^2 \\ + C_1 \dot{\theta}_1 = \frac{K_t V_a}{R_a} - \frac{K_t K_v \dot{\theta}_1}{R_a} \end{aligned} \quad (2.9)$$

$$(m_2 L_1 l_2 \cos \theta_2) \ddot{\theta}_1 + (I_2 + m_2 l_2^2) \ddot{\theta}_2 + m_2 g l_2 \sin \theta_2 + C_2 \dot{\theta}_2 = 0 \quad (2.10)$$

Untuk melinearkan model yang telah didapatkan, digunakan pendekatan sebagai berikut :

$$\cos \theta \approx 1, \sin \theta \approx \theta, \theta^2 \approx 0$$

Sehingga persamaan 2.9 dan 2.8 menjadi :

$$h_1 \ddot{\theta}_1 + h_2 \ddot{\theta}_2 + C_1 \dot{\theta}_1 = h_5 V_a - h_6 \dot{\theta}_1$$

$$h_2 \ddot{\theta}_1 + h_3 \ddot{\theta}_2 + h_4 \dot{\theta}_2 + C_2 \dot{\theta}_1 = 0$$

Dimana didefinisikan :

$$h_1 = I_1 + m_2 L_1^2$$

$$h_2 = m_2 L_1 l_2$$

$$h_3 = I_2 + m_2 L_2^2$$

$$h_4 = m_2 g l_2$$

$$h_5 = \frac{K_t}{R_a}$$

$$h_6 = \frac{K_t K_v}{R_a}$$

Dengan eliminasi, diperoleh persamaan :

$$\ddot{\theta}_1 = \frac{-h_3(h_6 + C_1)\dot{\theta}_1 - h_2 h_4 \theta_2 + h_2 C_2 \dot{\theta}_2 + h_3 h_5 V_a}{h_1 h_3 - h_2^2} \quad 2.11$$

$$\ddot{\theta}_2 = \frac{h_2(h_6 + C_1)\dot{\theta}_1 - h_1 h_4 \theta_2 + h_1 C_2 \dot{\theta}_2 + h_2 h_5 V_a}{h_1 h_3 - h_2^2} \quad 2.12$$

Misalkan,

$$H = \begin{bmatrix} h_1 & -h_2 \\ -h_2 & h_3 \end{bmatrix}, \det[H] = h_1 h_3 - h_2^2$$

$$d_1 = \frac{h_3}{\det[H]}$$

$$d_2 = -\frac{h_2}{\det[H]}$$

$$d_3 = -\frac{h_2}{\det[H]}$$

$$d_4 = \frac{h_1}{\det[H]}$$

Sehingga persamaan 2.10 dan 2.11 dapat dinyatakan sebagai berikut :

$$\ddot{\theta}_1 = -d_1(h_6 + C_1)\dot{\theta}_1 - d_2 h_4 \theta_2 - d_2 C_2 \dot{\theta}_2 + d_1 h_5 V_a \quad (2.13)$$

$$\ddot{\theta}_2 = -d_3(h_6 + C_1)\dot{\theta}_1 - d_4 h_4 \theta_2 - d_2 C_2 \dot{\theta}_2 + d_3 h_5 V_a \quad (2.14)$$

Persamaan keadaan dan output dari pergerakan pendulum terbalik, yaitu:

$$\begin{bmatrix} \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -d_1(h_6 + C_1) & -d_2 h_4 & -d_2 C_2 \\ 0 & 0 & 0 & 1 \\ 0 & -d_3(h_6 + C_1) & -d_4 h_4 & -d_2 C_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ d_1 h_5 \\ 0 \\ d_3 h_5 \end{bmatrix} V_a \quad (2.15)$$

2.2.3 Representasi State Space dari Model Matematika

Dengan mendefinisikan variabel state sebagai berikut:

$$x(t) = [\theta_1 \ \dot{\theta}_1 \ \theta_2 \ \dot{\theta}_2]^T$$

Matriks variabel state ditunjukkan oleh :

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

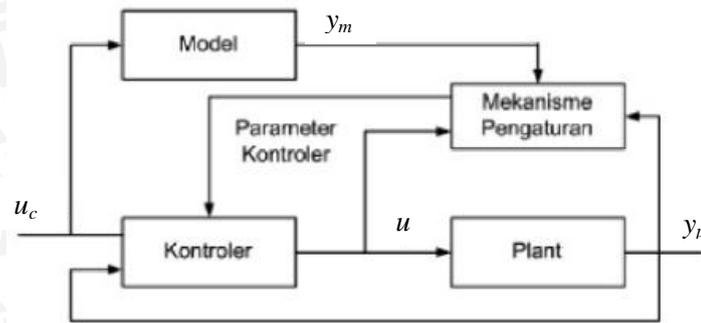
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -d_1(h_6 + C_1) & -d_2 h_4 & -d_2 C_2 \\ 0 & 0 & 0 & 1 \\ 0 & -d_3(h_6 + C_1) & -d_4 h_4 & -d_2 C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ d_1 h_5 \\ 0 \\ d_3 h_5 \end{bmatrix} V_a \quad (2.16)$$

2.3 Sistem Kontrol Adaptif Model Reference Adaptive Systems (MRAS)

Sistem kontrol adaptif adalah sistem kontrol dimana parameter-parameternya dapat diatur dan juga memiliki mekanisme untuk mengatur parameter-parameter tersebut. *Model Reference Adaptive Systems (MRAS)* merupakan salah satu kendali adaptif yang diperkenalkan pertama kali oleh Whitaker pada tahun 1958 (Butler, 1992). MRAS yang diperkenalkan oleh Whitaker mempunyai dua gagasan baru, yaitu:

1. Keluaran sistem ditentukan melalui keluaran model sistem.
2. Nilai parameter kontroler disesuaikan besarnya dengan kesalahan (error) yang terjadi antara keluaran model referensi dengan keluaran sistem yang dikendalikan.

Ide dasar MRAS adalah membuat keluaran sistem yang diatur sedemikian rupa sehingga mempunyai perilaku yang sama dengan model referensi yang diberikan. Hal ini dilakukan dengan mengubah-ubah nilai parameter kontrolernya. Parameter-parameter kontroler diatur melalui mekanisme pengaturan yang didasarkan pada *error* yang merupakan selisih antara keluaran proses dengan keluaran model referensi. Blok diagram skema *Model Reference Adaptive Systems (MRAS)* ditunjukkan pada Gambar 2.5.



Gambar 2.5 Blok diagram *Model Reference Adaptive Systems* (MRAS) (Astrom, 1995)

Dalam pengendalian ini, keluaran model referensi digunakan untuk menentukan keluaran sistem agar sesuai dengan yang diinginkan. Kontroler digunakan untuk menghasilkan sinyal kontrol u yang selanjutnya digunakan sebagai *input* bagi *plant*. *Plant* menghasilkan sinyal *output* y_p dan model referensi menghasilkan sinyal *output* y_m . *Output plant* dibandingkan dengan *output* model referensi, jika terdapat *error* di antara keduanya maka parameter pada kontroler akan berubah. Perubahan parameter kontroler dilakukan melalui hukum adaptasi. Di dalam hukum adaptasi, parameter kontroler ini akan diatur sedemikian rupa sehingga parameter kontroler dapat membenruk sinyal kontrol u yang akan menyebabkan *output plant* sesuai dengan model referensi. Apabila *plant* telah mengikuti model referensi, maka nilai parameter kontroler sudah tetap seperti yang diharapkan.

Skema sistem MRAS memiliki dua *loop*, yaitu *loop* dalam dan *loop* luar. *Loop* dalam terdiri atas *plant* dan kontroler. Sedangkan *loop* luar digunakan untuk menyesuaikan nilai parameter yang digunakan pada *loop* dalam. Setelah nilai parameter kontroler sudah tepat, maka *loop* dalam berjalan seperti pengendalian biasa.

Model reference adaptive systems bersifat adaptif karena nilai-nilai parameter kontroler dapat ditentukan secara *on-line*. Nilai-nilai parameter ini akan diperbaiki berdasarkan *error* yang terjadi antara *output* model dengan *output* sistem yang diatur.

2.3.1 Plant

Plant merupakan bagian dari sistem yang akan dikendalikan. *Plant* dinyatakan dalam fungsi alih. Fungsi alih $W_p(s)$ dengan keluaran $Y_p(t)$ terhadap sinyal kontrol $u(t)$ dapat dinyatakan dengan persamaan :

$$W_p(s) = \frac{Y_p(t)}{u(t)} = k_p \frac{\beta_p(s)}{\alpha_p(s)} \quad (2.17)$$

$\alpha_p(s)$ adalah polinomial dengan derajat n , $\beta_p(s)$ adalah polinomial berderajat m , dengan $m \leq n$. Diasumsikan bahwa koefisien $\alpha_p(s)$ dan $\beta_p(s)$ tidak diketahui, hanya n , m dan tanda penguatan k_p yang diketahui.

2.3.2 Model Referensi

Model referensi $W_m(s)$ menyatakan fungsi alih keluaran $Y_m(t)$ terhadap $r(t)$. Fungsi alih $W_m(s)$ dinyatakan sebagai berikut:

$$W_m(s) = \frac{Y_m(t)}{r(t)} = k_m \frac{\beta_m(s)}{\alpha_m(s)} \quad (2.18)$$

$\alpha_m(s)$ dan $\beta_m(s)$ adalah polinomial stabil, yaitu polinomial yang semua akarnya berada pada sebelah kiri bidang s , sedangkan k_m adalah konstanta. $\alpha_m(s)$ adalah polinomial dengan derajat n dan $\beta_m(s)$ adalah polinomial dengan derajat m , dengan $m \leq n$.

Keluaran dari model referensi menyatakan keluaran yang diinginkan dari *plant* yang dikendalikan jika sistem tersebut ditambah dengan pengendali yang tepat.

Dalam menentukan model referensi, derajat relatif model referensi harus sama dengan derajat relatif dari *plant*. Derajat relatif adalah selisih antara orde *pole* dan *zero* atau $n-m$ (Butler, 1992).

Namun perlu dicatat bahwa pada prakteknya ketika kebutuhan teoritis tidak dapat terpenuhi maka model referensi harus ditentukan dengan hati-hati dengan syarat keluaran proses sebenarnya dapat mengikuti keluaran model referensi. Jika model referensi dipilih yang memiliki respon terlalu cepat, sebagai contoh, maka sinyal kontrol dibutuhkan harus sangat besar, sehingga menyebabkan efek saturasi pada masukan atau dinamika tak termodelkan yang merusak sistem adaptif (Butler, 1992).

2.3.3 Kontroler Adaptif

Kontroler adaptif digunakan untuk menyesuaikan *output plant* dengan *output* model referensi. Struktur kontroler adaptif terdiri dari parameter kontroler θ dan sinyal *command* u_c yang dapat dinyatakan sebagai berikut:

$$u = \theta u_c \quad (2.19)$$

Sinyal kontrol u yang dihasilkan oleh kontroler digunakan sebagai *input* bagi *plant*, sehingga *output plant* sesuai dengan model referensi.

2.3.4 Hukum Adaptasi

Proses adaptasi terjadi melalui adanya perubahan nilai pada parameter kontroler. Proses ini berlangsung selama terjadi *error* antara keluaran *plant* $y_p(t)$ dan *output* model referensi $y_m(t)$. Proses adaptasi diatur melalui suatu cara yang disebut dengan hukum adaptasi (Narendra, 1989).

Dalam hukum adaptasi, parameter kontroler akan diubah sedemikian rupa sehingga *error* antara $y_p(t)$ dan $y_m(t)$ sama dengan nol. Pengaturan parameter kontroler akan menghasilkan sinyal kontrol u sebagai masukan bagi *plant*, sehingga *output* $y_p(t)$ akan mendekati $y_m(t)$ pada saat t menuju tak hingga. Hal tersebut dapat dituliskan sebagai berikut:

$$\lim_{t \rightarrow \infty} |y_p(t) - y_m(t)| = 0 \quad (2.20)$$

Pada MRAS, *error* antara y_p dan y_m digunakan secara langsung untuk mengubah nilai parameter kontroler. Proses adaptasi pada MRAS ini disebut dengan adaptasi langsung.

Salah satu hukum adaptasi yang digunakan adalah MIT *Rule* (aturan MIT). MIT *Rule* merupakan pendekatan orisinal pada MRAS. Nama MIT dipakai karena metode tersebut dikembangkan di Laboratorium Instrumentasi (sekarang Laboratorium Draper) di MIT.

Untuk mengetengahkan aturan MIT, dimisalkan sebuah sistem *loop* tertutup dimana kontrolernya mempunyai sebuah parameter θ yang dapat diatur. Respon *loop* tertutup yang diinginkan ditentukan dengan suatu model yang memiliki *output* y_m . Misalnya e sebagai *error* antara *output* sistem *loop* tertutup y dan *output* model y_m . Satu kemungkinan untuk mengatur parameter dengan meminimalisasi *loss function* (fungsi kerugian), J :

$$J(\theta) = \frac{1}{2} e^2 \quad (2.21)$$

Untuk mendapatkan nilai J kecil, merupakan hal yang beralasan untuk mengubah parameter dalam arah negatif gradient dari J , yaitu:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (2.22)$$

$$e = y_p - y_m = kG(s)\theta u_c - k_o G(s)\theta u_c \quad (2.23)$$

$$\frac{\partial e}{\partial \theta} = kG(s)\theta u_c = \frac{k}{k_o} y_m \quad (2.24)$$

MIT Rule memberikan hukum adaptasi berikut:

$$\frac{d\theta}{dt} = -\gamma e \frac{k}{k_o} y_m \quad (2.25)$$

Dari persamaan diatas dapat dirubah menjadi domain-s untuk mendapatkan θ ,

$$s\theta = -\gamma e \frac{k}{k_o} y_m \quad (2.26)$$

$$\theta = -\frac{1}{s} \gamma e \frac{k}{k_o} y_m = -\frac{1}{s} \gamma e y_m \quad (2.27)$$

Turunan parsial $\partial e/\partial \theta$ yang disebut dengan turunan sensitivitas sistem, menunjukkan bagaimana *error* dipengaruhi oleh parameter yang dapat diatur. Jika diasumsikan bahwa parameter berubah lebih lambat daripada variabel yang lain di dalam sistem, sehingga turunan $\partial e/\partial \theta$ dapat dievaluasi di bawah asumsi bahwa θ adalah konstan.

2.3.5 Persamaan Error

Pada pengendalian dengan MRAS, proses adaptasi akan berhenti apabila keluaran *plant* sama dengan keluaran model referensi. Dengan kata lain *error* sama dengan 0. Sedangkan persamaan *error* antara model referensi dengan *plant* adalah:

$$e(t) = y_p(t) - y_m(t) \quad (2.28)$$

2.4 Rotary Encoder

Rotary encoder adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. Rotary encoder umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah.

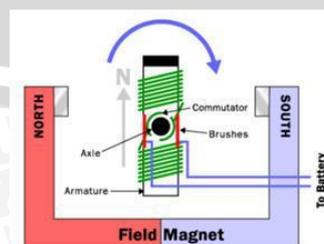
Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh rotary encoder untuk diteruskan oleh rangkaian kendali.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau divais berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi rotary encoder tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi rotary encoder tersebut.

2.5 Motor DC

Motor dc memiliki konstruksi yang sama dengan motor AC, perbedaannya terletak pada sikat dan cincin belah (komulator). Saat siklus pertama, arus mengalir dari kutub positif ke negatif. Aliran arus yang melewati bagian kabel yang berada didekat kutub N magnet akan menimbulkan gaya Lorentz ke bawah.

Sementara itu aliran arus yang melewati kabel yang berada di dekat kutub S magnet akan menyebabkan gaya Lorentz ke atas. Kedua perpaduan gaya Lorentz tersebut akan menyebabkan kawat berputar. Pada siklus berikutnya terjadi hal yang serupa seperti pada siklus sebelumnya. Motor DC ditunjukkan dalam gambar berikut :



Gambar 2.6 Gambar melintang motor DC

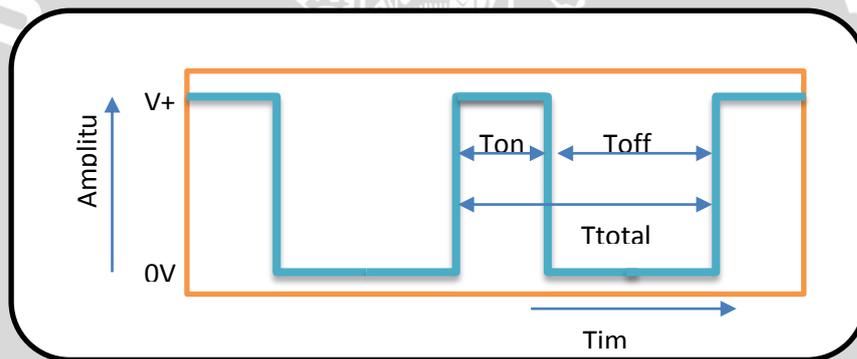
(<http://fahmizaleeits.wordpress.com>)

Apabila arus terus-menerus dialirkan, maka kawat akan berputar secara terus menerus pula. Pada aplikasi sesungguhnya, kawat adalah sebuah rotor yang akan dikopel dengan sebuah as dan akan memutar as tersebut terus menerus seiring perputaran motor.

2.6 PWM (Pulse Width Modulation)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM secara umum dapat dilihat dalam gambar berikut



Gambar 2.7 Sinyal PWM secara umum

$$T_{total} = T_{on} + T_{off}$$

$$\%D = \frac{T_{on}}{T_{total}} \times 100\% = \dots (\%) \tag{2.29}$$

dimana :

T_{on} = Periode logika tinggi

T_{off} = Periode logika rendah

D = Dutycycle (lamanya pulsa tinggi dalam satu periode)

$$V_{out} = \frac{T_{on}}{T_{total}} \times V_{in} = \dots (v) \tag{2.30}$$

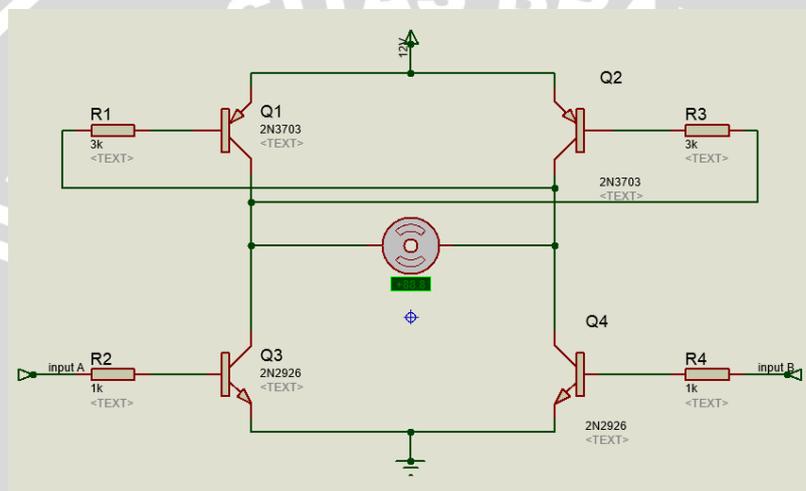


Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut :

$$f_{on} = \frac{f_{clk}1/10}{N.256} = \dots \text{ (Hz)} \quad (2.31)$$

2.7 Driver Motor H-Bridge

Rangkaian driver motor DC ini disebut dengan h-bridge dikarenakan konfigurasi/susunan transistornya seperti membentuk huruf H. Transistor – transistor ini digunakan sebagai switching sehingga nantinya motor dapat berputar searah jarum jam (*clockwise*) dan berlawanan arah jarum jam (*counterclockwise*).



Gambar 2.8 Rangkaian H-Bridge Driver

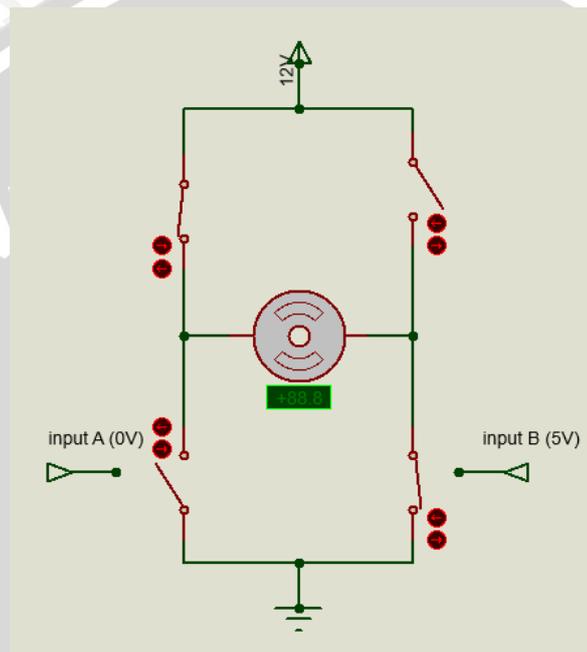
Cara kerja rangkaian h-bridge sebagai berikut :

1. A = B = '0'

Karena input A dan B mempunyai logika yg sama '0' (0V), maka kedua transistor TIP31 (Q_1 & Q_2) tidak akan mendapat picuan pada basisnya sehingga transistor bersifat cut-off atau transistor bersifat seperti saklar yg terbuka. Dari rangkaian diatas terlihat pula bahwa kedua TIP 32 (Q_3 & Q_4) bergantung pada TIP31 dimana basis kedua TIP32 terhubung pada kolektor TIP 31. Jadi, apabila tidak ada arus yg mengalir pada kolektor TIP 31 maka basis TIP 32 juga tidak akan terpicu akibatnya motor tidak akan berputar atau berhenti.

2. A = '0'; B = '1'

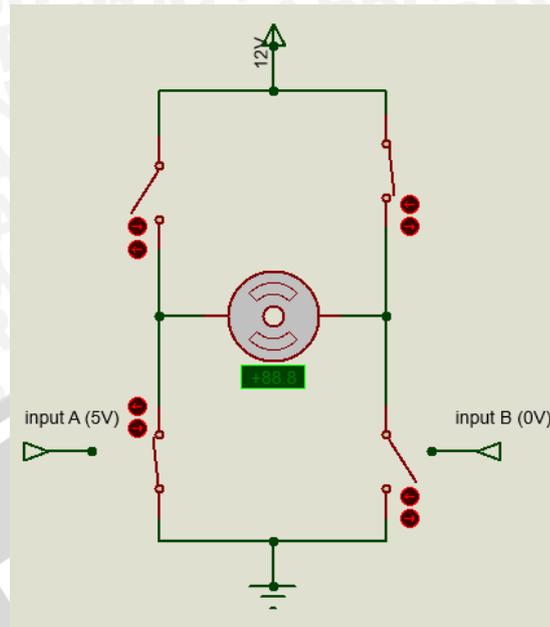
Saat input A diberi logika '0' (0V) dan input B diberi logika '1' (5V) maka Q_2 akan saturasi sedangkan Q_1 tetap cut-off. Karena Q_2 bersifat saturasi atau seperti saklar yang tertutup maka basis Q_3 akan mendapat picuan sehingga Q_3 juga bersifat saturasi. Akibatnya arus akan mengalir dengan urutan seperti berikut : $V_s - Q_3 - \text{motor} - Q_1 - \text{ground}$, sehingga motor akan berputar searah jarum jam.



Gambar 2.9 Rangkaian H-Bridge saat A = '0'; B = '1'

3. A = 1; B = 0

Saat input A diberi logika '1' (5V) dan input B diberi logika '0' (0V) maka Q_1 akan saturasi sedangkan Q_2 cut-off. Akibatnya Q_4 juga akan menjadi saturasi karena basis Q_4 mendapat picuan dari Q_1 . Sehingga arus akan mengalir dengan urutan seperti berikut : $V_s - Q_4 - \text{motor} - Q_1 - \text{ground}$ dan motor akan berputar berlawanan arah jarum jam.



Gambar 2.10 Rangkaian H-Bridge saat A = '1'; B = '0'

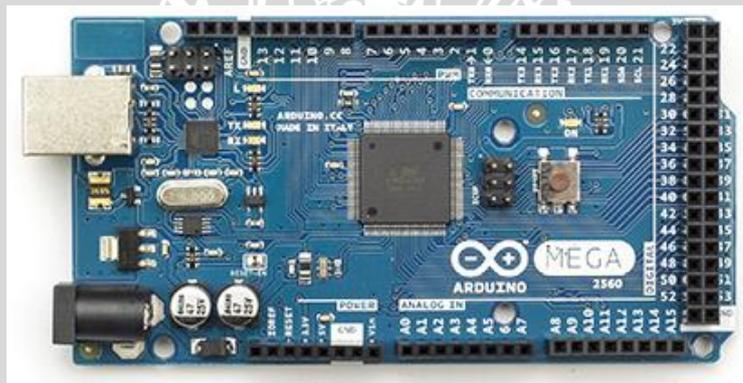
4. A = B = '1'

Jika kedua input diberi logika '1' secara bersamaan maka akan mengakibatkan semua transistor dalam kondisi saturasi. Secara logika motor tidak akan berputar karena tidak ada beda potensial pada ujung – ujung konektornya. Namun hal ini akan menyebabkan timbulnya panas yang berlebihan pada semua transistor sehingga dapat menyebabkan kerusakan. Oleh karena itu hal ini harus dihindari.

2.8 Arduino Mega 2560

Arduino Mega merupakan sebuah mikrokontroler berdasarkan ATmega2560, memiliki 54 digital input / output pin (yang 14 dapat digunakan sebagai output PWM), input analog 16, 4 UART (*port serial hardware*), osilator kristal 16 MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Arduino juga dapat didukung melalui koneksi USB dari komputer, dari baterai 9V, atau dari catu daya. Arduino dapat dikontrol dari komputer atau diprogram oleh komputer dan kemudian terputus dan memungkinkan untuk bekerja secara independen. Arduino Mega telah kompatibel dengan sebagian shield yang dirancang untuk Arduino Duemilanove atau Diecimila. Berikut adalah karakteristik dari Arduino Mega 2560:

Microcontroller	Atmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



Gambar 2.11 Arduino Mega 2560
(<http://arduino.cc>)

Perangkat lunak untuk pemrograman Arduino mudah dan juga tersedia secara bebas untuk Windows, Mac, dan komputer LINUX.

2.8.1 Daya

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Eksternal (non-USB) daya dapat berasal baik dari adaptor AC-DC atau baterai. Papan arduino dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. Jika daya kurang dari 7V, namun, 5V pin dapat menyediakan daya kurang dari lima volt dan papan arduino

mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak papan. Rentang yang dianjurkan adalah 7 sampai 12 volt. Pin listrik adalah sebagai berikut:

1. VIN. Tegangan input ke papan Arduino ketika menggunakan sumber daya eksternal (sebagai lawan 5 volt dari koneksi USB atau regulasi daya lain). Anda dapat menyediakan tegangan melalui pin ini, atau jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
2. 5V. Merupakan pin output 5V diatur dari regulator di papan arduino. Papan arduino dapat diaktifkan dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN dari papan arduino (7-12V). Menyediakan tegangan melalui 5V atau 3.3V pin melewati regulator, dan dapat merusak papan arduino. Kami tidak menyarankan itu.
3. 3V3. Sebuah pasokan 3,3 volt dihasilkan oleh regulator didalam papan arduino. Arus maksimumnya adalah 50 mA.
4. GND. Pin tanah.
5. IOREF. Pin ini pada papan Arduino memberikan tegangan referensi seperti mikrokontroler. Sebuah perisai dikonfigurasi dengan benar dapat membaca pin tegangan IOREF dan pilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

2.8.2 Memory

ATmega2560 memiliki 256 KB dari flash memory untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan EEPROM *library*).

2.8.3 Input dan Output

Masing-masing dari 54 digital pin pada Mega dapat digunakan sebagai input atau output, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Fungsi tersebut beroperasi pada 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up* internal (terputus secara *default*) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

1. Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pins 0 dan 1 juga terhubung ke pin yang sesuai dari ATmega16U2 USB-to-TTL Serial keping.
2. Eksternal Interupsi: 2 (interrupt 0), 3 (menggangu 1), 18 (menggangu 5), 19 (menggangu 4), 20 (menggangu 3), dan 21 (menggangu 2). Pin ini dapat dikonfigurasi untuk memicu interrupt pada nilai yang rendah, naik atau jatuh, atau perubahan nilai. Lihat *attachInterrupt()* fungsi untuk rincian.
3. PWM: pin 2 sampai 13 dan 44 sampai 46. Menyediakan 8-bit PWM output dengan fungsi *analogWrite()*.
4. SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan SPI *library*. Pin SPI juga pecah pada header ICSP, yang secara fisik kompatibel dengan Uno, Duemilanove dan Diecimila.
5. LED: 13 Ada *built-in* LED terhubung ke pin digital 13 Ketika pin memiliki nilai TINGGI, LED menyala, ketika pin memiliki nilai RENDAH, maka mati.
6. TWI: 20 (SDA) dan 21 (SCL). Dukungan komunikasi TWI menggunakan *Wire Library*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Duemilanove atau Diecimila.

Arduino Mega2560 memiliki 16 input analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* mereka mengukur dari tanah sampai 5 volt, meskipun memungkinkan untuk mengubah batas atas menggunakan pin AREF dan fungsi *analogReference()*. Ada beberapa pin lainnya di papan arduino sebagai berikut :

1. AREF. Referensi tegangan untuk input analog. Digunakan dengan *analogReference()*.
2. Reset. Digunakan untuk me-reset mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset untuk melindungi dari sesuatu yang menghalangi di papan arduino.

2.8.4. Komunikasi

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega2560

menyediakan empat UART hardware untuk TTL (5V) komunikasi serial. Sebuah ATmega16U2 (ATmega 8U2 pada revisi 1 dan revisi 2 papan arduino) pada salah satu papan USB dan menyediakan *port com* virtual untuk perangkat lunak pada komputer (Windows akan memerlukan file inf, tapi OSX dan Linux akan mengakui papan arduino sebagai *port COM* secara otomatis). Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data tekstual sederhana yang akan dikirim ke dan dari papan. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui *Chip* ATmega8U2 / ATmega16U2 dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *SoftwareSerial library* memungkinkan untuk komunikasi serial pada salah satu pin digital pada Mega2560.

ATmega2560 juga mendukung TWI dan SPI komunikasi. Perangkat lunak Arduino termasuk *Wire library* untuk menyederhanakan penggunaan bus TWI; lihat dokumentasi untuk rincian. Untuk komunikasi SPI, menggunakan *SPI library*.

2.8.5. Pemrograman

Arduino mega dapat diprogram dengan software Arduino. ATmega2560 pada Arduino mega datang *preburned* dengan *bootloader* yang memungkinkan Anda dapat meng-*upload* kode baru tanpa menggunakan programmer hardware eksternal. Komunikasi ini menggunakan protokol STK500 asli (referensi, file header C). Anda juga dapat memotong *bootloader* dan program mikrokontroler melalui ICSP (In-Circuit Serial Programming) Header menggunakan Arduino ISP atau serupa.

ATmega16U2 (atau 8U2 dalam papan Rev1 dan rev2) sumber kode *firmware* tersedia di *repositori* Arduino. ATmega16U2 / 8U2 diisikan dengan *bootloader* DFU, yang dapat diaktifkan oleh:

- Pada papan Rev 1: menghubungkan jumper solder di bagian belakang papan (dekat peta Italia) dan kemudian reset 8U2.
- Pada papan Rev2: ada resistor yang menarik 8U2 / 16U2 HWB baris ke ground, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU. Kemudian dapat menggunakan FLIP software Atmel (Windows) atau

programmer DFU (Mac OS X dan Linux) untuk memuat firmware baru. Dapat juga menggunakan header ISP dengan programmer eksternal (Timpa *bootloader* DFU).

2.8.6. Otomatis (*Software*) Atur Ulang

Daripada membutuhkan *pers* fisik dari tombol reset sebelum *upload*, Arduino Mega2560 dirancang dengan cara yang memungkinkan untuk diatur ulang oleh perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu garis kontrol aliran hardware (DTR) dari ATmega8U2 terhubung ke garis *reset* dari ATmega2560 melalui 100 nanofarad kapasitor. Bila saluran ini menegaskan (diambil rendah), garis *reset* turun cukup lama untuk me-*reset chip*. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan Anda untuk meng-*upload* kode dengan hanya menekan tombol *upload* di lingkungan Arduino. Ini berarti bahwa *bootloader* dapat memiliki batas waktu yang lebih pendek, seperti penurunan DTR yang dapat terkoordinasi dengan dimulainya *upload*.

Pengaturan ini memiliki implikasi lain. Ketika Mega2560 terhubung ke salah satu komputer yang menjalankan Mac OS X atau Linux, akan me-*reset* setiap kali sambungan dibuat dari software (melalui USB). Untuk berikutnya setengah detik atau lebih, *bootloader* sedang berjalan pada Mega2560. Sementara itu diprogram untuk mengabaikan data yang cacat (yaitu selain *upload* kode baru), ia akan memotong beberapa byte pertama dari data yang dikirim ke papan setelah sambungan dibuka. Jika berjalan *sketch* di papan menerima konfigurasi satu kali atau data lain ketika pertama kali dijalankan, pastikan bahwa perangkat lunak dengan yang berkomunikasi menunggu satu detik setelah membuka koneksi dan sebelum mengirim data.

Mega2560 mengandung jejak yang dapat dipotong untuk menonaktifkan *auto-reset*. Bantalan di kedua sisi jejak dapat disolder bersama untuk mengaktifkan kembali itu. Ini berlabel "RESET-EN". Anda juga mungkin dapat menonaktifkan *auto-reset* dengan menghubungkan 110ohm resistor dari 5V ke garis reset.

2.8.7. Perlindungan USB Arus Berlebih

Arduino Mega2560 memiliki *polyfuse reset* yang melindungi *port* USB komputer Anda dari *shorts* dan arus lebih. Meskipun kebanyakan komputer memberikan perlindungan internal mereka sendiri, sekering memberikan lapisan tambahan perlindungan. Jika lebih dari 500 mA diterapkan ke port USB, sekering secara otomatis akan mematahkan sambungan atau overload akan dihapus.

2.8.8. Karakteristik Fisik dan Kompatibilitas Pelindung

Panjang maksimum dan lebar PCB Mega2560 4 dan 2,1 inci masing-masing, dengan konektor USB dan jack listrik yang melampaui dimensi versi berikutnya. Tiga lubang sekrup memungkinkan papan harus terpasang ke permukaan. Perhatikan bahwa jarak antara pin digital 7 dan 8 adalah 160 mil (0.16"), bukan kelipatan genap dari jarak 100 mil dari pin lainnya.

Mega2560 dirancang agar kompatibel dengan sebagian pelindung yang dirancang untuk Uno, Diecimila atau Duemilanove. Pin Digital 0-13 (dan berdekatan antara AREF dan pin GND), analog input 0 sampai 5, header listrik, dan *header* ICSP semua di lokasi setara. Selanjutnya UART utama (*port* serial) terletak di pin yang sama (0 dan 1), seperti interupsi eksternal 0 dan 1 (pin 2 dan 3 masing-masing). SPI tersedia melalui *header* ICSP pada Mega2560 dan Duemilanove / Diecimila. Harap dicatat I2C yang tidak terletak pada pin yang sama pada Mega (20 dan 21) sama halnya pada Duemilanove / Diecimila (input analog 4 dan 5).