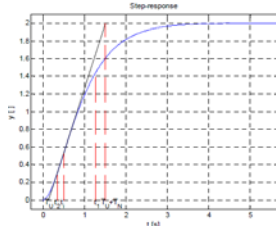


Gambar 2 Diagram Blok Sistem

B. Identifikasi Strejc [3]

Metode Strejc merupakan metode untuk identifikasi *plant* dengan cara menarik garis singgung pada titik belok respon step *plant* dalam domain waktu. Penarikan garis ini ditunjukkan pada Gambar 3.



Gambar 3. Penarikan garis pada Metode Strejc

Pada metode strejc penarikan garis berdasar pada titik belok (inflection point) dari grafik respons. Dengan ini didapatkan nilai  $T_u$  dan  $T_N$ .  $T_u$  merupakan waktu tunda sistem dan  $T_N$  merupakan waktu saat respons melewati waktu tunda hingga mencapai batas penarikan garis terhadap titik belok respons. Nilai  $T_u$  dan  $T_N$  ini berpengaruh terhadap orde pant dan untuk memodelkan respon *plant* berdasarkan tabel Strejc. Dengan diketahui nilai  $T_u$  dan  $T_N$ , maka dapat dicari nilai  $\tau$ ,

$$\tau = \frac{T_u}{T_N} \tag{1}$$

Untuk  $\tau \geq 0$  sistem didekati dengan orde ke  $n$  dengan persamaan

$$G_{ST2}(s) = \frac{K}{(\tau_{ST}s + 1)^n} \tag{2}$$

Berdasarkan nilai  $\tau$  dapat ditentukan nilai  $y_t$  berdasarkan nilai yang ada pada Tabel 2. Dengan nilai  $y_t$ , maka nilai  $t_i$  dapat ditemukan dengan melihat respon sistem. Jika nilai  $t_i$  sudah diketahui maka dapat ditemukan nilai  $\tau_{ST}$ .

$$\tau_{ST} = \frac{t_i}{n - 1} \tag{3}$$

Tabel 1. Parameter identifikasi Metode Strejc

n	$\tau$	$y_t$
2	0.104	0.264
3	0.218	0.327
4	0.319	0.359
5	0.41	0.371
6	0.493	0.384
7	0.57	0.394
8	0.642	0.401
9	0.709	0.407
10	0.773	0.413

C. Generalized Predictive Control (GPC)[4][5]

GPC yang telah diperkenalkan oleh Clarke *et al* [4] telah banyak digunakan pada proses dalam industri. Hal ini disebabkan karena kemampuan pengaturan yang stabil untuk sistem dengan parameter, *dead time*, dan orde model yang berubah ubah.

Ketika memperhatikan titik operasi, *plant* non linier umumnya menggunakan model linier pada daerah tertentu. Salah satu permodelan yang umum digunakan adalah model CARMA (Controlled Auto-Regresive Moving Average),

$$A(q^{-1}) = B(q^{-1})u(t - 1) + C(q^{-1})\xi(t) \tag{4}$$

dengan  $u(t)$  *input* kontrol,  $y(t)$  variabel yang terukur atau *output*, dan  $\xi(t)$  gangguan. Dimana A dan B adalah polinomial

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_naq^{-na} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_nbq^{-nb} \\ C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_ncq^{-nc} \end{aligned}$$

Sinyal *output* prediksi  $\hat{y}(k + d + 1 + n_c)$  dituliskan sebagai penjumlahan dari *free response* dan *force response*.

$$\hat{y} = \hat{y}_{force} + \hat{y}_{free} \tag{5}$$

*Free Response* merupakan respon hasil pengaruh dari *input* lampau, *output* yang akan datang dengan asumsi aksi kontrol dari waktu saat ini. *Force response* merupakan respon hasil pengaruh *input* saat ini dan yang akan datang terhadap *output* yang akan datang. Persamaan *force response* adalah sebagai berikut :

$$\begin{aligned} \hat{y}_{force}(k + d + 1 + n_c|k) &= \frac{B(q^{-1})}{A(q^{-1})}u(k + 1 + n_c|k) \\ &= G(q^{-1})u(k + 1 + n_c|k) \end{aligned} \tag{6}$$

atau

$$\hat{y}_{force} = \mathbf{G}u_f$$

di mana

$$u_f = [u(k|k), u(k + 1|k), \dots, u(k + n_{c2} - 1|k)]^T,$$

$G(q^{-1})$  adalah fungsi alih pulsa dari proses tanpa dead time fisik.

Variabel manipulasi untuk saat ini dan akan datang disebut  $u(k + i), i \geq 0$  didesain dengan waktu diskrit, dan bisa ditulis  $\Delta u(k + i|k)$ . Untuk lebih mudahnya penulisan  $\Delta u(k + i|k)$  hanya ditulis  $\Delta u(k + i)$ .  $\mathbf{G}$  merupakan koefisien respon unit step

$$\mathbf{G} = \begin{bmatrix} g_{nc1+1} & g_{nc1} & \dots & g_{nc1-n_u+2} \\ g_{nc1+2} & g_{nc1+1} & \dots & g_{nc1-n_u+1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{nc2+1} & g_{nc1} & \dots & g_{nc2-n_u+2} \end{bmatrix}$$

Berdasarkan strategi *receding horizon* hanya *input* pertama  $u(k)$  digunakan dan dihitung dan diulang ke langkah kontrol selanjutnya,

$$\mathbf{u}(k) = [1, 0, \dots, 0][\mathbf{G}^T \mathbf{G} + \lambda_u \mathbf{I}]^{-1} \mathbf{G}^T (\mathbf{y}_r - \hat{y}_{free}) \tag{7}$$

dimana vektor  $[1, 0, \dots, 0]$  memiliki  $n_u - 1$  nilai nol.

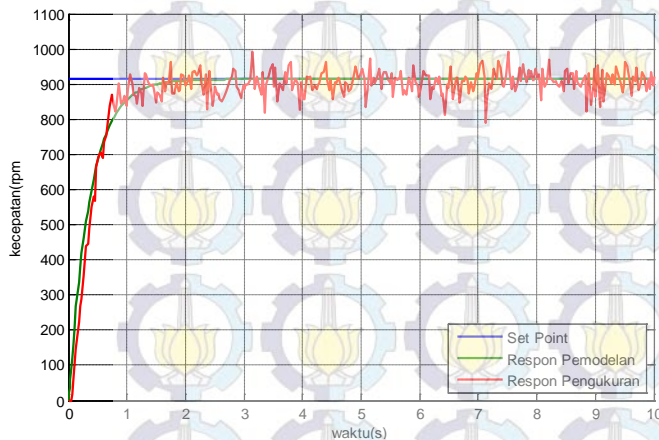
III. IDENTIFIKASI DAN PERANCANGAN KONTROLER

A. Identifikasi Plant

Identifikasi pada ICE dan motor DC dilakukan pada sistem open loop dengan identifikasi statis, yaitu dengan memberikan *input* berupa sinyal step. Pada ICE, sinyal step berupa injakan pedal yang langsung diarahkan kebatas maksimum pedal. Sedangkan pada motor DC dilakukan secara manual dengan memutar autotrafo. Pengambilan data ini dilakukan dengan menghubungkan tachogenerator dengan arduino yang telah terhubung ke software Simulink MATLAB.

B. Identifikasi ICE

Dalam perancangan kontroler, dilakukan identifikasi *plant* pada saat beban nominal. Permodelan dengan beban nominal dilakukan memberikan beban berupa rem elektromagnetik yang diberikan tegangan 60V. Grafik respon ICE dengan beban nominal dapat dilihat pada Gambar 4.



Gambar 4. Hasil Identifikasi dan Pemodelan ICE Beban Nominal

Dari grafik tersebut ditemukan parameter permodelan Strejc seperti berikut : nilai K didapatkan dari

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{914}{914} = 1$$

Selanjutnya dicari nilai  $T_U$  dan  $T_N$  yang didapatkan dengan menarik garis miring yang memotong titik belok respon saat transien. dari nilai  $T_U$  dan  $T_N$  didapatkan nilai  $\tau$ .

$$\tau = \frac{T_U}{T_N} = \frac{0,06133}{0,5843} = 0,1049$$

Setelah nilai  $\tau$  diketahui, dengan melihat Tabel 1., maka ditemukan nilai  $y_i$ . Nilai tersebut kemudian dikalikan dengan nilai  $Y_{ss}$ , dan ditemukan suatu nilai.

$$y_i = 0,264(914) = 241,296$$

Dengan melihat grafik respon sistem yang ditunjukkan pada Gambar 4, maka ditemukan  $t_i$  saat  $y_i$

$$t_i = 0,1848$$

Kemudian dengan menggunakan persamaan 3, didapatkan nilai  $\tau_{ST}$

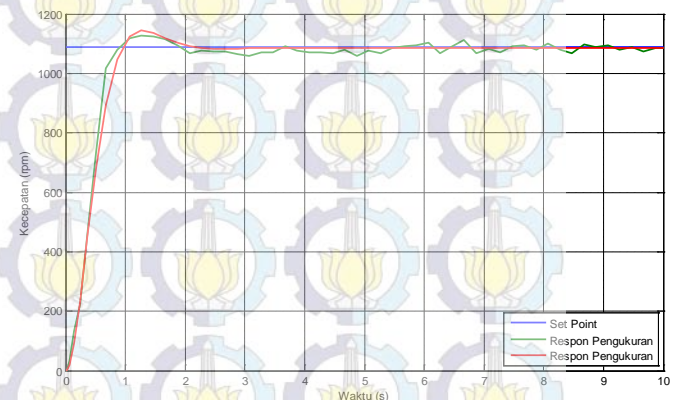
$$\tau_{ST} = 0,1848$$

Dari perhitungan di atas didapatkan fungsi alih ICE dengan beban nominal sebagai berikut.

$$G(s) = \frac{1}{(0,1848s + 1)^2}$$

C. Identifikasi Motor DC

Proses identifikasi motor DC melalui tahapan yang hampir sama dengan motor bakar. Motor DC diberikan tegangan 75V dc dari autotrafo. Grafik respon motor DC ditampilkan pada Gambar 5. Dari gambar tersebut terlihat bahwa terdapat *overshoot* pada respon sistem, sehingga dengan respon demikian permodelan Strejc tidak bisa dilakukan. Oleh karena itu dilakukan permodelan dengan pendekatan analitik sebagai berikut :



Gambar 5 Respon Kecepatan Motor DC

Menentukan besar overshoot maksimum (persen) dengan dengan nilai  $c(t_p) = 1126$ , dan  $c(\infty) = 1090$

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% = 5,5\%$$

Selanjutnya adalah mencari nilai  $\omega_d$  yang merupakan frekuensi alami teredam dengan persamaan berikut

$$\omega_d = \frac{\pi}{t_p} = \frac{3,14}{1,287} = 2,44$$

Untuk menemukan nilai koefisien peredaman ( $\xi$ ), digunakan persamaan:

$$M_p = e^{-(\xi/\sqrt{1-\xi^2})\pi} = 0,678$$

Setelah mendapatkan nilai  $\xi$ , maka dapat ditemukan nilai  $\omega_n$  dengan persamaan :

$$\omega_d = \omega_n \sqrt{1 - \xi^2} \rightarrow \omega_n = 3,318$$

Respon sistem menunjukkan bahwa sistem tergolong redaman kurang atau *under damped* ( $0 < \xi < 1$ ), maka fungsi alih sistem dapat dituliskan sebagai berikut

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

$$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$$

D. Model Referensi

Model ini didesain dengan tidak memiliki *error steady state* dan memiliki *settling time* yang lebih cepat dibanding ICE. Dari fungsi alih ICE yang memiliki *settling time* ( $\pm 5\%$ ) sebesar 1,2. Maka ditentukan *settling time* model referensi

lebih cepat yaitu 0,6s. Model referensi dirancang dengan model orde satu dengan nilai  $\tau$

$$T_s(\pm 5\%) = 3\tau$$

$$\tau = 0,2$$

Sehingga didapatkan model referensi sebagai berikut

$$G(s) = \frac{1}{0,2s + 1}$$

**E. Perancangan Kontroler**

Pada penelitian ini, *Generalized Predictive Control* (GPC) digunakan untuk mengatur motor DC. Pengaturan motor DC ini bertujuan untuk membantu motor bakar dalam meningkatkan akselerasi. Hal ini disebabkan karena respon motor bakar lambat.

Kontroler ini disusun berdasarkan kinerja motor DC. Dari model matematika motor DC, maka dilakukan permodelan ARMA (*AutoRegressive Moving Average*). Bentuk ARMA didapatkan dari penyusunan blok identifikasi ARMAX pada simulink Matlab. Ditentukan nilai orde sistem adalah 2 dan orde *input* adalah 2 dan *time sampling* sebesar 0,1, dengan dijalankan selama 100s maka didapatkan hasil permodelan ARMA sebagai berikut:

$$\frac{Y(z)}{U(z)} = \frac{0,047215z + 0,0406226}{z^2 - 1,548z + 0,63763}$$

$$y(k) = 1,548y(k - 1) - 0,63763y(k - 2) + 0,047215u(k - 1) - 0,0406226u(k - 2)$$

Dengan

$$A(q^{-1}) = a_1q^{-1} + a_2q^{-2}$$

$$= 0,047215q^{-1} + 0,0406226q^{-2}$$

$$B(q^{-1}) = 1 - b_1q^{-1} + b_2q^{-2}$$

$$= 1 - 1,548q^{-1} + 0,63763q^{-2}$$

Setelah model ARMA ditemukan, langkah selanjutnya yaitu menentukan panjang *prediction horizon*. Penentuan ini harus disesuaikan dengan respon yang diharapkan. Apabila panjang horizon terlalu pendek akan menimbulkan overshoot, sedangkan apabila terlalu panjang respon sistem akan lambat walaupun tidak terdapat overshoot. Dari model referensi yang telah ditentukan sebelumnya, diharapkan respon lebih cepat dari pada sebelumnya, sehingga ditentukan panjang *prediction horizon* sebesar 3.

$$y(k) = b_1y(k - 1) + b_2y(k - 2) + a_1u(k - 1) + a_2u(k - 2)$$

$$y(k + 1) = b_1y(k) + b_2y(k - 1) + a_1u(k) + a_2u(k - 1)$$

$$y(k + 2) = (b_1^2 + b_2)y(k) + b_1b_2y(k - 1) + a_1u(k + 1) + (b_1a_1 + a_2)u(k) + b_1a_2u(k - 1)$$

$$y(k + 3) = (b_1^3 + 2b_1b_2)y(k) + (b_1^2b_2 + b_2^2)y(k - 1) + a_1u(k + 2) + (b_1a_1 + a_2)u(k + 1) + (b_1^2a_1 + b_2a_1 + b_1a_2)u(k) + (b_1^2b_2 + b_2^2)u(k - 1)$$

Dari ketiga persamaan diatas maka dapat dibentuk matriks sebagai berikut

$$\hat{y} = \hat{y}_{force} + \hat{y}_{free}$$

$$\begin{bmatrix} \hat{y}(k + 1) \\ \hat{y}(k + 2) \\ \hat{y}(k + 3) \end{bmatrix} = G \begin{bmatrix} u(k) \\ u(k + 1) \\ u(k + 2) \end{bmatrix} + \begin{bmatrix} \hat{y}_{free}(k + 1) \\ \hat{y}_{free}(k + 2) \\ \hat{y}_{free}(k + 3) \end{bmatrix}$$

$$= \begin{bmatrix} g_1 & 0 & 0 \\ g_2 & g_1 & 0 \\ g_3 & g_2 & g_1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k + 1) \\ u(k + 2) \end{bmatrix} + \begin{bmatrix} \hat{y}_{free}(k + 1) \\ \hat{y}_{free}(k + 2) \\ \hat{y}_{free}(k + 3) \end{bmatrix}$$

Dengan

$$\begin{bmatrix} g_1 & 0 & 0 \\ g_2 & g_1 & 0 \\ g_3 & g_2 & g_1 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 \\ b_1a_1 + a_2 & a_1 & 0 \\ b_1^2a_1 + b_2a_1 + b_1a_2 & b_1a_1 + a_2 & a_1 \end{bmatrix};$$

$$\begin{bmatrix} \hat{y}_{free}(k + 1) \\ \hat{y}_{free}(k + 2) \\ \hat{y}_{free}(k + 3) \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & a_2 \\ b_1^2 + b_2 & b_1b_2 & b_1a_2 \\ b_1^3 + 2b_1b_2 & b_1^2b_2 + b_2^2 & b_1^2b_2 + b_2^2 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k - 1) \\ u(k - 1) \end{bmatrix}$$

Sinyal kontrol didapatkan dengan

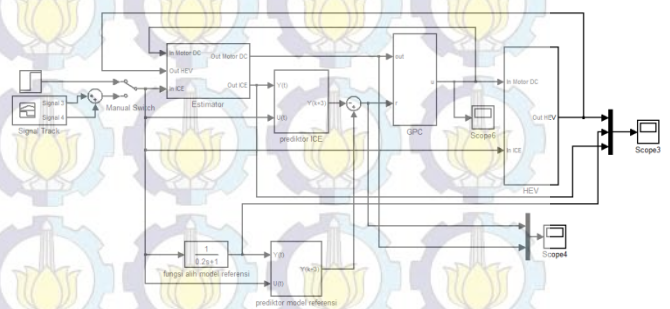
$$u(k) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [G^T G + \lambda_u I]^{-1} G^T (y_r - \hat{y}_{free})$$

Dengan

$$y_r = \begin{bmatrix} y_{r+1} \\ y_{r+2} \\ y_{r+3} \end{bmatrix}$$

**IV. SIMULASI, IMPLEMENTASI, DAN ANALISA**

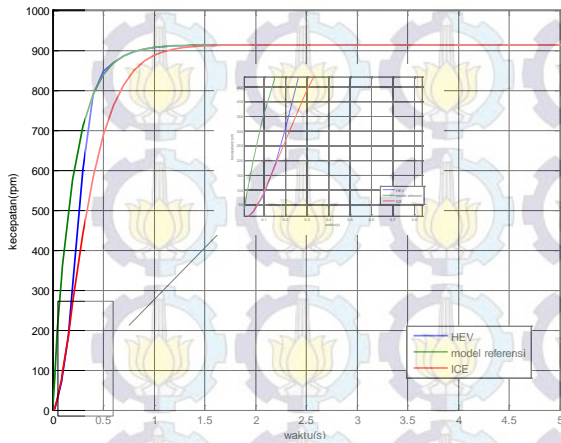
Sistem pengaturan traksi pada HEV melibatkan motor DC sebagai penggerak sekunder. Oleh karena itu motor DC akan bekerja ketika terdapat error antara kecepatan ICE dan model referensi. Blok diagram simulasi ditunjukkan pada Gambar 6.



**Gambar 6.** Blok Diagram Simulasi dengan Simulink

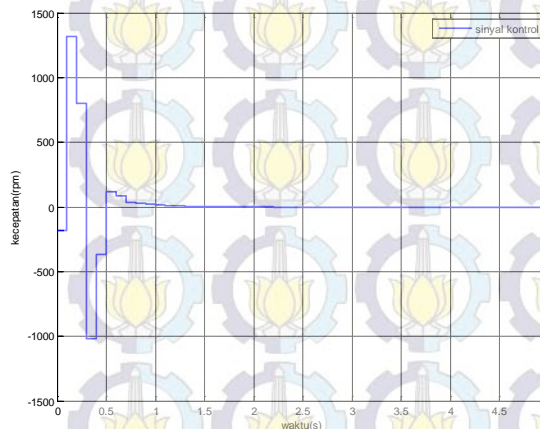
Respon HEV dengan kontroler GPC 3 prediksi kedepan ditunjukkan pada Gambar 7. Gambar tersebut menunjukkan bahwa kontrol GPC yang dirancang sudah dapat melaksanakan aksi kontrol setelah menjalani *satu time sampling* (0.1s). Setelah melalui 0,1s, kontroler bekerja mendekati model referensi. Walau tertinggal diawal, respon *plant* yang telah diberi kontroler dapat mendahului model

referensi setelah 0,4s dan mempunyai *rise time* sebesar 0,353s.



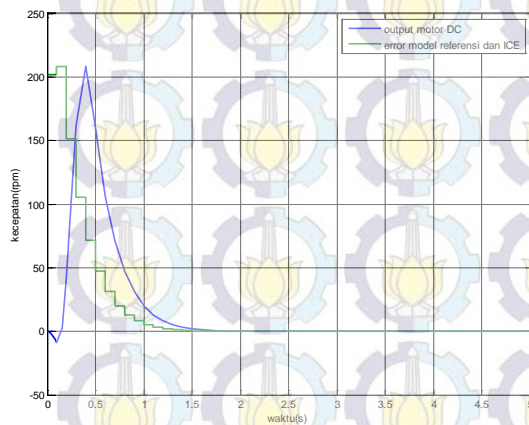
**Gambar 7.** Respon HEV dengan kontroler GPC 3 prediksi kedepan dengan *input* sinyal step

Dari kontroler GPC tersebut menghasilkan sinyal kontrol berbentuk diskrit seperti yang ditunjukkan pada Gambar 8.



**Gambar 8** Sinyal kontrol GPC 3 prediksi kedepan dengan *input* sinyal step

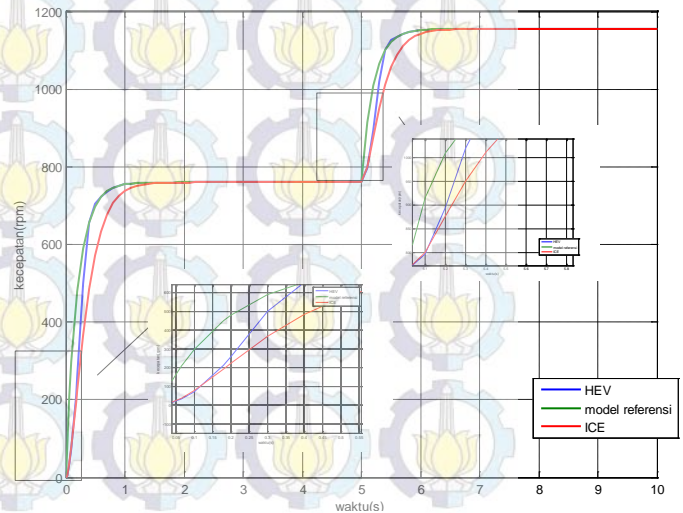
Sinyal kontrol menunjukkan bahwa kontroler memberikan *input* kecepatan sebesar 1288rpm pada motor DC saat 0,1s. Saat 0,3s kontroler menghasilkan sinyal kontrol sebesar -968,43rpm. Hal ini menunjukkan bahwa kontroler bekerja secara maksimal setelah dapat melakukan prediksi. *Output* motor DC ditunjukkan pada Gambar 9.



**Gambar 9** *Output* Motor DC dengan kontroler GPC 3 prediksi kedepan dengan *input* sinyal step

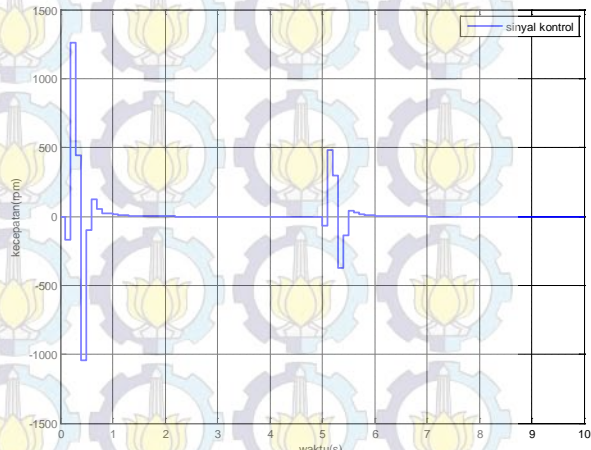
Gambar tersebut menunjukkan bahwa motor DC dapat mengikuti sinyal referensi dan motor DC akan berhenti saat kecepatan ICE sama dengan model referensi. Kecepatan maksimal yang dihasilkan motor DC sebesar 206,9s.

Saat diberikan sinyal uji berupa sinyal step dengan perubahan amplitudo, respon sistem ditunjukkan pada Gambar 10. Gambar tersebut menunjukkan bahwa sistem dapat mengikuti model referensi. Sama halnya ketika diberi sinyal uji step, setelah melalui 0,1s, kontroler bekerja mendekati model referensi. Walau tertinggal diawal, respon *plant* yang telah diberi kontroler dapat mendahului model referensi setelah 0,4s dan memiliki *rise time* sebesar 0,34s. Untuk kenaikan amplitudo *plant* yang telah diberi kontroler dapat mendahului model referensi setelah 0,4s dan memiliki *rise time* sebesar 0,35s sama seperti saat step pertama.



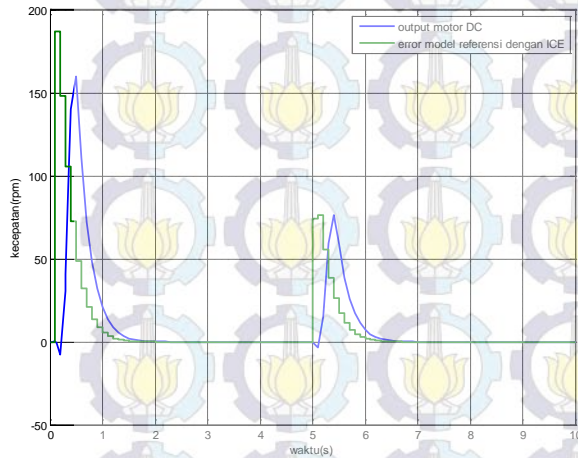
**Gambar 10** Respon HEV dengan kontroler GPC 3 prediksi kedepan dengan perubahan amplitudo *input* sinyal step

Seperti halnya ketika diberikan sinyal uji step, saat diberikan sinyal step dengan amplitudo yang berbeda, sinyal kontrol akan menyesuaikan dengan kenaikan sinyal step. Sinyal kontrol mengalami kenaikan saat 0,1s setelah perubahan sinyal masukan dan bernilai 1071rpm. Saat 0,3s sinyal kontrol turun dan bernilai -805,26rpm. Sinyal kontrol ditunjukkan pada Gambar 11.



**Gambar 11** Sinyal kontrol GPC 3 prediksi kedepan dengan perubahan amplitudo *input* sinyal step

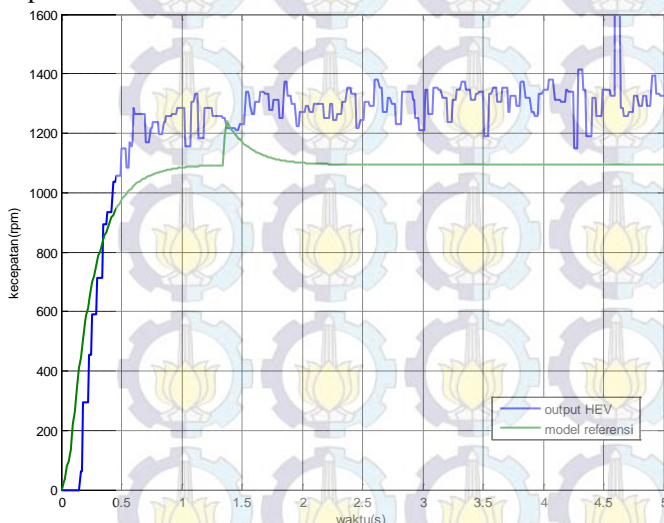
Hal ini menunjukkan bahwa kontroler membutuhkan waktu untuk menyesuaikan nilainya dengan model referensi. *Output* motor DC menunjukkan bahwa motor DC membantu ICE saat keadaan akselerasi. *Output* motor DC ini ditunjukkan pada Gambar 12



Gambar 12 *Output* Motor DC dengan kontroler GPC 3 prediksi kedepan dengan perubahan amplitudo *input* sinyal step

Setelah mensimulasikan kontroler, langkah selanjutnya adalah mengimplementasikan kontroler pada simulator HEV. Implementasi dilakukan dengan menghubungkan potensiometer dan tachometer pada mikrokontroler arduino. Mikrokontroler megolah data dan ditampilkan pada Simulink Matlab. Selanjutnya matlab akan mengirimkan sinyal kontrol melalui komunikasi serial arduino dan arduino akan mengeluarkan sinyal kontrol berupa *output* PWM. *Output* PWM kemudian dihubungkan dengan driver motor DC. Driver ini menggerakkan motor DC berdasarkan sinyal kontrol yang dikehendaki.

Pengujian dilakukan dengan memberikan kontroler GPC dengan prediksi 3 langkah ke depan. Respon hasil implementasi ditunjukkan pada Gambar 13. gambar tersebut menunjukkan bahwa *output* HEV dapat mengikuti model referensi pada saat transien. respon HEV memiliki *rise time* sebesar 0,29s dan memiliki error sebesar 11,5% terhadap setpoint.



Gambar 13. Implementasi 3 prediksi kedepan

## V. KESIMPULAN

Dari hasil simulasi maka dapat disimpulkan bahwa saat *plant* motor DC pada HEV diberikan kontroler GPC tanpa informasi masa depan saat start, mulai bekerja dengan baik setelah mencapai 0,1s (satu *time sampling*). Dan mulai mengikuti model referensi saat 0,4s. Saat diberikan *input* berupa sinyal step dengan perubahan amplitudo, respon HEV sama dengan saat diberikan *input* sinyal step, yaitu kontroler bekerja dengan baik setelah mencapai 0,1s dan mengikuti model referensi saat 0,4 s. Metode GPC dapat memperbaiki respon dengan peningkatan *rise time* dari 0,624s ke 0,35s. Hal ini membuktikan bahwa kontroler GPC sesuai untuk mempercepat respon *transien* dari suatu *plant*

Kontroler GPC cocok digunakan untuk *plant* yang memiliki pengetahuan terhadap *output* yang akan datang, karena sinyal kontrol yang dihasilkan dapat langsung mengikuti model referensi saat mulai start.

## DAFTAR PUSTAKA

- [1] Sriwidodo, Wahyu. "Perancangan dan Implementasi Pengendalian Traksi Berbasis Neuro Fuzzy untuk Simulator Kendaraan Hybrid pada Mode Akselerasi". *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [2] Satrio, Galih. "Desain dan Implementasi Kontroler Optimal Berbasis Neuro Fuzzy untuk Pengendalian Simulator Hybrid Electric Vehicle", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [3] Ing. Pavel Jakoubek, "Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods", *Konference Studenske Tvuurci-Cinnosti*, 2009.
- [4] Clarke, D.W., Mohtadi, C., Tuffs, P.S., "Generalized Predictive Control-Part I. The Basic Algorithm" *Automatica*, Vol. 23, No. 2, pp. 137-148 Pergamon Journals Ltd, 1987.
- [5] Haber, Robert., Bars, Ruth., Schmitz, Ulrich., "Predictive Control in Process Engineering", Weinheim, Germany: WILEY-VCH Verlag GmbH & Co. KgaA, 2011