

# Pengaturan Kecepatan pada Simulator *Parallel Hybrid Electric Vehicle* (PHEV) Menggunakan *Linear Quadratic Regulator* (LQR) Berdasarkan *Particle Swarm Optimization* (PSO)

Widhayaka Aji Candra, Rusdhianto Effendie A.K, dan Ali Fatoni

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: rusdhi@elect-eng.its.ac.id, fatoni@ee.its.ac.id

**Abstrak**—*Hybrid Electric Vehicle* (HEV) merupakan suatu jenis kendaraan hibrida yang menggabungkan sistem daya gerak *Internal Combustion Engine* (ICE) dengan sistem daya gerak listrik dari motor listrik. ICE dan motor listrik pada HEV dengan konfigurasi parallel dapat bekerja bersama – sama, sehingga tenaga yang dihasilkan bisa lebih besar untuk mempertahankan kecepatan terhadap adanya pembebanan pada kendaraan yang disebut dengan permasalahan regulator. Hal tersebut merupakan kelebihan dari HEV dibandingkan dengan mesin konvensional yang hanya menggunakan ICE. Metode optimal kontrol *Linear Quadratic Regulator* (LQR) digunakan untuk pengaturan kecepatan pada simulator HEV. Indeks performansi yang minimum dan nilai kesalahan terkecil menjadi perhatian dalam penelitian ini. Penentuan matriks pembobot Q dan R merupakan salah satu langkah agar tercapainya tujuan tersebut. Penggunaan algoritma cerdas *Particle Swarm Optimization* (PSO) dapat melakukan proses *tuning* matriks Q dan R berdasar multi kriteria yang diinginkan. Hasil pengujian, didapatkan kontrol LQR mampu mereduksi sinyal kesalahan pada saat terjadi pembebanan berlebih pada rentang beban nominal 0-60 VDC rem elektromagnetik, sehingga kecepatan yang diinginkan dapat terus dipertahankan dengan bantuan motor DC sebesar 99,4 %. Hasil implementasi  $e_{ss}$  relatif kecil sebesar 1,9%.

**Kata Kunci**— *Hybrid Electric Vehicle*, *Linear Quadratic Regulator*, *Particle Swarm Optimization*, *Internal Combustion Engine*, motor listrik

## I. PENDAHULUAN

Seiring berkembangnya teknologi otomotif, kendaraan *Hybrid Electrical Vehicles* (HEV) mulai dikembangkan untuk menyempurnakan *Electrical Vehicles* (EV). EV adalah suatu penemuan baru karena adanya permasalahan akibat *Internal Combustion Engine* (ICE). Semakin banyak kendaraan menggunakan mesin konvensional maka semakin banyak pula polusi berupa gas berbahaya yang berasal dari ICE yang dapat merusak lingkungan. Selain itu, bahan bakar sebagai sumber daya yang tak terbarukan di perut bumi juga semakin berkurang [1]. EV memiliki jarak tempuh yang pendek, pengisian baterai yang cukup lama dan boros.

Kendaraan *Hybrid Electrical Vehicles* (HEV) dengan mesin *hybrid* adalah kendaraan yang menggunakan dua atau lebih tenaga sebagai sumber penggerak [1], dalam hal ini



Gambar 1. Simulator HEV

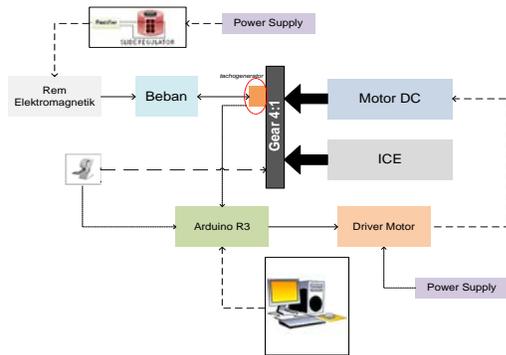
menggabungkan antara mesin bakar dan motor listrik. Sehingga Tenaga yang digunakan bisa berasal dari mesin bakar, motor listrik, maupun gabungan diantara keduanya. Kelebihan dari mesin *hybrid* adalah saat mesin bakar tidak mampu mempertahankan kecepatan maka motor listrik akan membantu memberikan tenaga, sehingga dengan penggunaan bahan bakar yang minimal kendaraan tetap bisa mempertahankan performa terbaiknya.

Metode kontrol optimal *Linear Quadratic Regulator* (LQR) digunakan untuk mengatur kecepatan motor listrik agar dapat membantu ICE saat terjadi pembebanan yang dapat menurunkan kecepatan kendaraan. Untuk mendapatkan indeks performansi dan nilai kesalahan yang minimum dapat diperoleh dengan menentukan matriks pembobotan Q dan R yang optimal berdasarkan algoritma *Particle Swarm Optimization* (PSO). Kontrol optimal diharapkan mampu mempertahankan kecepatan PHEV agar dapat mencapai kecepatan yang diinginkan dengan adanya pembagian kerja antara kedua sumber penggerak. Pada penelitian sebelumnya telah dihasilkan sebuah simulator HEV dengan konfigurasi parallel yang juga digunakan dalam Tugas Akhir ini [2][3].

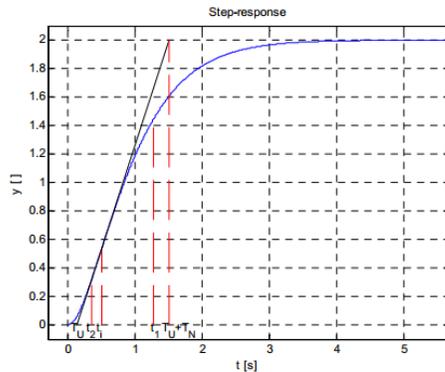
## II. SISTEM SIMULATOR HEV DAN METODE KONTROL

### A. Simulator HEV [2][3]

HEV merupakan kendaraan *hybrid* yang menggunakan *Internal Combustion Engine* (ICE) dan motor listrik sebagai penggerak kendaraan untuk menanggung beban.



Gambar 2. Konfigurasi Perangkat Keras Simulator HEV



Gambar 3. Respon Step Sistem dengan Waktu  $t_1$ ,  $t_2$ ,  $t_i$ ,  $T_U$ , dan  $T_N$ [4]

Pada konfigurasi paralel seperti pada simulator HEV Gambar 1, yang digunakan pada Tugas Akhir ini, saat ICE sebagai penggerak utama diberikan beban berlebih maka kecepatan putar yang dihasilkan akan berkurang sehingga motor DC sebagai penggerak tambahan akan membantu ICE untuk mencapai kecepatan putar yang diinginkan. Pengujian ini dilakukan dengan memberikan beban pada saat ICE telah mencapai kecepatan stabil sehingga kecepatan sistem akan menurun dan diketahui fungsi motor DC membantu kerja ICE untuk mencapai kecepatan awal. Konfigurasi perangkat keras simulator HEV ditunjukkan pada Gambar 2.

**B. Metode Identifikasi Strejc [4]**

Identifikasi sistem dengan menggunakan metode Strejc merupakan pendekatan sistem yang tidak beresilasi tanpa waktu tunda dengan bantuan perhitungan waktu  $T_U$  dan  $T_N$  yang didapat dari garis singgung yang dibuat bersinggungan dengan respon sistem seperti yang terlihat pada Gambar 3. Perbandingan antara waktu tersebut diperoleh parameter  $\tau$ .

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

Untuk  $\tau \geq 0$ , sistem didekati dengan model orde ke-n.

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

Dengan nilai  $\tau_{ST}$  dapat ditentukan dengan cara berikut:

- 1) Menentukan perbandingan  $\tau$  seperti pada Persamaan (1).
- 2) Berdasarkan dari nilai  $\tau$ , cari orde yang sesuai dari pendekatan model dan koordinat titik singgung  $y_i$  antara respon sistem dengan garis singgung berdasar Tabel 1.
- 3) Berdasarkan  $y_i$ , cari  $t_i$  yang merupakan waktu saat respon sistem mencapai  $y_i$  berdasarkan pada Gambar 3.

Tabel 1.

Taksiran dari Orde ke-n dan Nilai Titik Singgung

n	$\tau$	$y_i$	n	$\tau$	$y_i$	n	$\tau$	$y_i$
2	0,014	0,264	5	0,41	0,371	8	0,642	0,401
3	0,218	0,327	6	0,493	0,384	9	0,709	0,407
4	0,319	0,359	7	0,57	0,394	10	0,773	0,413

4) Mencari konstanta waktu  $\tau_{ST}$  dari Persamaan (2).

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

**C. Linear Quadratic Regulator (LQR) [5]**

Pokok bahasan dalam desain sistem kontrol optimal adalah menentukan sinyal kontrol yang akan di proses untuk memenuhi batasan fisik sesuai dengan kriteria performansi yang diinginkan. Desain kontrol optimal dilakukan melalui formulasi variabel *state*. Pada dasarnya, solusi kontrol optimal dititikberatkan pada pencarian nilai sinyal kontrol optimal  $u(t)$  sehingga indeks performansi dioptimasi. *Linear Quadratic Regulator* merupakan salah satu metode kontrol optimal pada sistem linier dengan kriteria kuadratik untuk menyelesaikan permasalahan regulator.

Diberikan suatu *plant* linier dalam persamaan *state space*:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4}$$

di mana,

- A** :  $n \times n$  matriks *state*
- B** :  $n \times r$  matriks *control*
- x(t)** :  $n$ th vektor *state*
- u(t)** :  $r$ th vektor *control*

dan indeks performansi dalam interval  $[0, \infty ]$ :

$$J = \frac{1}{2} \int_0^{\infty} [\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t)] dt \tag{5}$$

di mana,  $\mathbf{u}(t)$  tanpa batasan,  $t_0$  adalah waktu awal dan  $t_f$  adalah waktu akhir *infinite*. **Q** adalah matriks  $n \times n$  simetri, semi-definit positif dan **R** adalah matriks  $m \times m$  simetri, definit positif. Untuk mendapatkan sinyal kontrol  $u(t)$  sehingga apabila harga keluaran awal tidak nol, maka masukannya harus ditentukan agar keluarannya menjadi nol. Kondisi ini mungkin terjadi ketika *plant* mengalami gangguan yang tidak diinginkan pada *state*. Kontrol optimal diberikan:

$$\mathbf{u}^* = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{x}^* = -\mathbf{K} \mathbf{x}^* \tag{6}$$

dengan Kalman gain

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \tag{7}$$

di mana,  $\bar{\mathbf{P}}$ ,  $n \times n$  simetri, definit positif, matriks konstan, merupakan solusi dari nonlinier dengan persamaan *Algebraic Riccati Equation* (ARE) yang dituliskan sebagai berikut:

$$\bar{\mathbf{P}} \mathbf{A} + \mathbf{A}^T \bar{\mathbf{P}} + \mathbf{Q} - \bar{\mathbf{P}} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \bar{\mathbf{P}} = 0 \tag{8}$$

Penentuan matriks pembobot **Q** dan **R** berpedoman pada:

- a. Semakin besar harga matriks **Q**, maka akan memperbesar harga elemen matriks gain kontrol dan mempercepat sistem mencapai *steady state*.
- b. Semakin besar harga matriks **R**, maka akan memperkecil harga elemen matriks gain kontrol dan memperlambat sistem mencapai *steady state*.

D. Particle Swarm Optimization (PSO) [6] [7]

PSO merupakan metode komputasi stokastik berdasarkan pergerakan kawanan dalam mencari lokasi makanan yang dikembangkan dari perilaku sosial kawanan burung, ikan, lebah dan lainnya. Variasi atau modifikasi dari algoritma PSO orisinal untuk mengendalikan konvergensi partikel dengan memasukkan parameter “inertia weighting”, sehingga persamaannya:

$$v_{i,m}^{(t+1)} = wv_{i,m}^{(t)} + c_1R(pbest_{i,m} - x_{i,m}^{(t)}) + c_2R(gbest_m - x_{i,m}^{(t)}) \tag{9}$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)} \tag{10}$$

$$i = 1,2,\dots, n$$

$$m = 1,2,\dots,d$$

dengan  $w$  adalah inertia weighting,

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} iter$$

di mana,

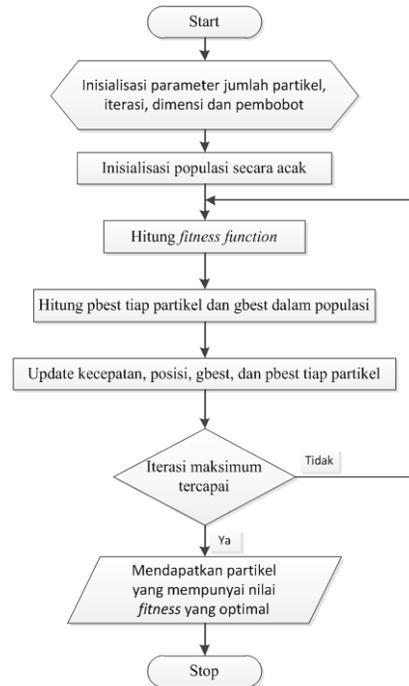
- $n$  : jumlah partikel dalam kelompok
- $d$  : dimensi
- $t$  : petunjuk iterasi (generasi)
- $v_{i,m}^{(t)}$  : kecepatan partikel ke- $i$  pada iterasi ke  $t$
- $c_1, c_2$  : konstanta akselerasi (learning rate)
- $r_1, r_2$  : bilangan random (0 – 1)
- $x_{id}(t)$  : posisi saat ini dari partikel ke- $i$  pada iterasi ke- $t$
- $pbest_i$  : posisi terbaik sebelumnya dari partikel ke- $i$
- $gbest$  : partikel terbaik diantara semua partikel dalam satu kelompok atau populasi
- $w_{max}$  : initial inertia weight
- $w_{min}$  : final inertia weight
- $iter$  : iterasi berjalan

Persamaan PSO memiliki 3 bagian antara lain bagian pertama adalah momentum part, kecepatan partikel tidak dapat berubah secara bebas melainkan mengalami perubahan melalui kecepatan sekarang. Bagian kedua adalah cognitive part, merupakan bagian pengalaman partikel, belajar dari pengalaman partikel itu sendiri. Dan bagian ketiga adalah social part, merupakan kerja sama antar partikel, belajar dari pengalaman partikel – partikel lain [6]. Gambar 4. menunjukkan diagram alir dari algoritma PSO

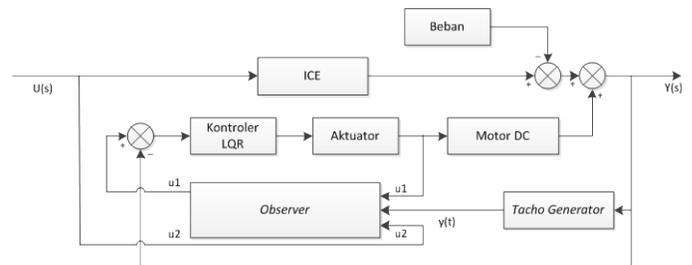
III. PERANCANGAN SISTEM

A. Identifikasi Sistem

Proses identifikasi yang dilakukan pada Tugas Akhir ini ialah identifikasi respon transien statis secara open loop dengan sinyal masukan step. Sinyal masukan step diberikan secara manual melalui pedal untuk ICE dan melalui auto transformator berupa tegangan DC untuk motor DC. Hasil respon plant yang terbaca oleh tacho generator, kemudian dikirimkan ke komputer melalui proses akusisi data oleh mikrokontroler Arduino untuk dapat ditampilkan dalam bentuk grafik pada HMI yang telah dirancang pada LabVIEW. Diagram blok sistem seperti yang ditunjukkan pada Gambar 3.



Gambar 4. Diagram Alir Algoritma PSO



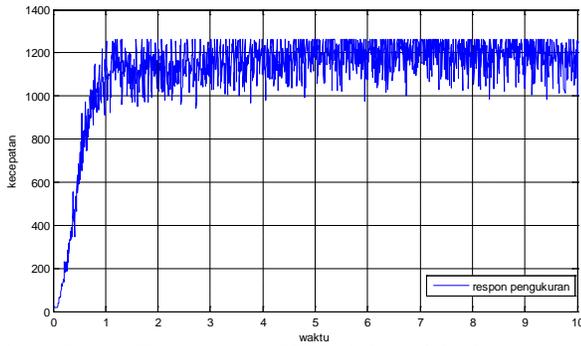
Gambar 3. Diagram Blok Plant

B. Identifikasi Mesin Bakar

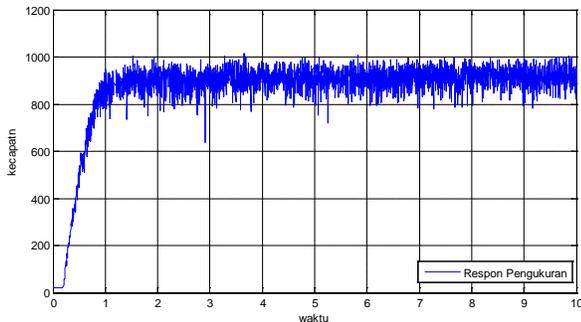
Pendekatan model pada ICE dilakukan dengan memberikan nilai yang sama antara  $X_{ss}$  dan  $Y_{ss}$  untuk beban nominal. Hal ini dilakukan karena pada Tugas Akhir ini membahas tentang permasalahan regulator, dimana bukaan throttle ICE terbuka penuh dianggap telah mencapai nilai masukan yang diinginkan dan difokuskan terhadap kinerja dari motor DC ketika membantu ICE ketika diberikan beban yang berubah-ubah agar respon sistem dapat kembali stabil menuju nilai steady state. Identifikasi pada ICE dilakukan dengan beban yang berbeda-beda, yaitu minimal, nominal, dan maksimal. Proses perhitungan untuk mendapatkan permodelan plant dilakukan sebanyak 4 kali dengan metode yang sama untuk mencari permodelan yang paling mendekati respon plant sebenarnya.

Hasil permodelan untuk ICE berbeban minimal dapat dilihat pada Gambar 5. Dari gambar tersebut kemudian dilakukan perhitungan pendekatan model dengan metode Strejc, didapat model orde ke-2 sebagai berikut:

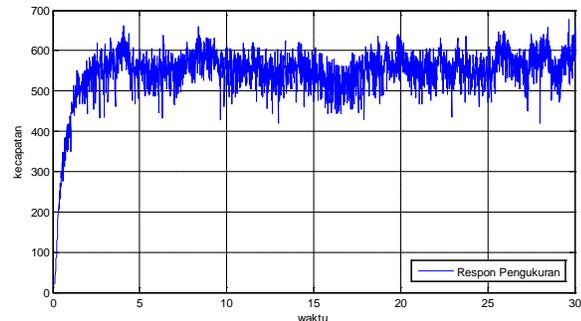
$$G(s) = \frac{1,288}{(0,3118 + 1)^2} \tag{11}$$



Gambar 5. Respon Kecepatan Putar ICE Berbeban Minimal



Gambar 6. Respon Kecepatan Putar ICE Beban Nominal



Gambar 7. Respon Kecepatan Putar ICE Beban Maksimal

Hasil permodelan untuk ICE berbeban nominal dapat dilihat pada Gambar 6. Pendekatan model dengan metode Strejc, didapat model orde ke-2 sebagai berikut:

$$G(s) = \frac{1}{(0,1848 + 1)^2} \tag{12}$$

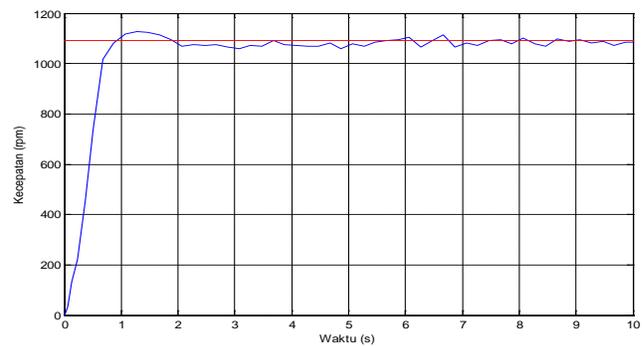
Hasil permodelan untuk ICE berbeban maksimal dapat dilihat pada Gambar 7. Pendekatan model dengan metode Strejc, didapat model orde ke-2 sebagai berikut:

$$G(s) = \frac{0,601}{(0,289 + 1)^2} \tag{13}$$

C. Identifikasi Motor Listrik

Permodelan *plant* pada motor DC dilakukan menggunakan permodelan analitik dengan beban minimal. Hubungan *input output* dalam fungsi alih sebagai berikut:

$$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)} \tag{14}$$



Gambar 8. Respon Kecepatan Motor DC

D. Perancangan Kontroler

Perancangan kontroler LQR terdapat beberapa prosedur yang dilakukan untuk mendapatkan parameter kontroler. Perhitungan kontroler LQR membutuhkan parameter *plant* dalam bentuk *state space*. Sehingga fungsi alih yang didapatkan dari proses identifikasi *plant* harus diubah dalam representasi *state space*.

Representasi *state space* model matematika ICE pada Persamaan (12) dinyatakan sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -29,412 & -10,87 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 29,412 \end{bmatrix} u_1$$

$$y_1 = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

dan model matematika motor DC pada Persamaan (14) dapat direpresentasi *state space* sebagai berikut

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -11,056 & -4,5 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 11,01 \end{bmatrix} u_2$$

$$y_2 = [1 \ 0] \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$

Perhitungan parameter kontroler LQR dilakukan melalui beberapa tahapan yaitu:

- Menentukan matriks pembobot Q dan R dengan menggunakan *Particle Swarm Optimization* (PSO)
- Menyelesaikan persamaan matriks *Algebraic Riccati Equation* (ARE)

Mendapatkan kontrol optimal  $u^*(t) = Kx^*(t)$  dengan Kalman Gain  $K = R^{-1} B^T P$

E. Algoritma PSO

Dalam algoritma PSO terdapat suatu fungsi objektif atau fungsi *fitness* yang ditentukan oleh dua faktor, yaitu faktor *root mean square* (RMSE) dan faktor indeks performansi (J). Fungsi *fitness* tersebut dapat dinyatakan dalam Persamaan (15) berikut:

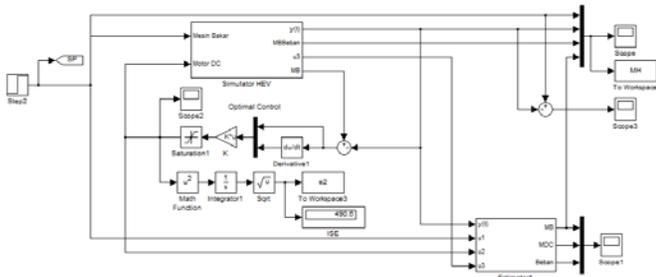
$$F(i) = \alpha \times RMSE(i) + \beta \times J(i) \tag{15}$$

di mana,

- RMSE : *Root mean square*
- J(i) : Indeks performansi
- $\alpha, \beta$  : *Improvement weight*

IV. SIMULASI

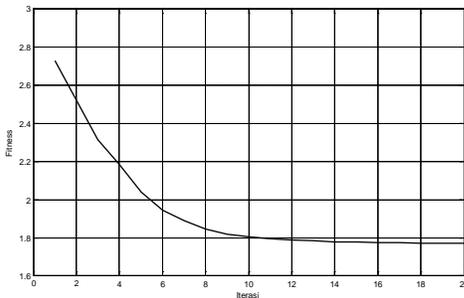
Dalam perancangan simulasi sistem, akan dibuat semirip mungkin dengan sistem *real* yang akan digunakan untuk



Gambar 9. Blok Simulink SimulasiBKontroler LQR

Tabel 2 . Pengaruh Nilai Bobot Inersia PSO pada Simulator HEV

ITEM Rata – rata		Bobot Inersia			
		w = 0,52	w = 0,54	w = 0,56	w = 0,58
Matrik Pembobot	Q1	45,5845	45,8086	48,2443	86,0326
	Q2	0,9815	1,4441	1,2352	0,7841
	R	0,002	0,002	0,002	0,002
	K1	149,9701	150,3409	154,3121	206,4020
	K2	22,3549	26,9688	25,0042	20,3208
Performansi	RMSE	0,8253	0,9041	0,8201	0,7092
	J	3,9533x10 <sup>6</sup>	7,9290x10 <sup>6</sup>	4,1884x10 <sup>6</sup>	7,4720x10 <sup>6</sup>
	ISE	434,3770	602,3992	434,6161	435,3558
Performansi PSO	Fitness Value	20,6311	22,6006	20,2948	17,7265



Gambar 10. Grafik Konvergensi PSO saat Bobot Inersia w = 0,58 implementasi. Perangkat lunak yang digunakan dalam perancangan simulasi yaitu menggunakan MATLAB 7.10.0 (R2010a) seperti pada Gambar 7.

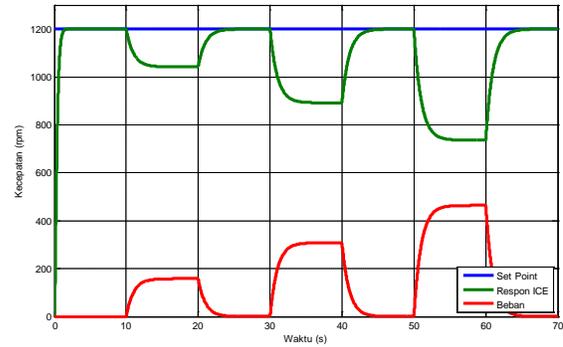
A. Simulasi Tuning Matriks Q Dan R

Proses *tuning* matriks **Q** dan **R** menggunakan PSO dengan parameter yang digunakan jumlah partikel, n = 10, jumlah iterasi, iter = 20, bobot inersia untuk PSO original, w = 0,54, bobot inersia untuk PSO modifikasi, w<sub>1</sub> = 0,75 dan w<sub>2</sub> = 0,25, konstanta akselerasi *cognitive*, c<sub>1</sub> = 1,5, konstanta akselerasi sosial, c<sub>2</sub> = 1,5. Proses *tuning* dilakukan beberapa kali dengan nilai bobot berbeda-beda seperti pada Tabel 2.

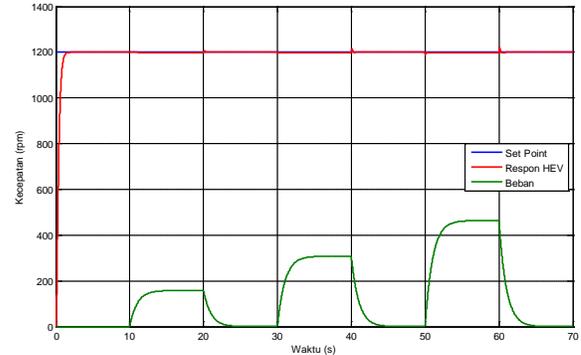
Grafik konvergensi hasil *tuning* matriks **Q** dan **R** dengan algoritma PSO ditunjukkan pada Gambar 9. untuk w = 0,58.

B. Simulasi ICE Tanpa Bantuan Motor DC

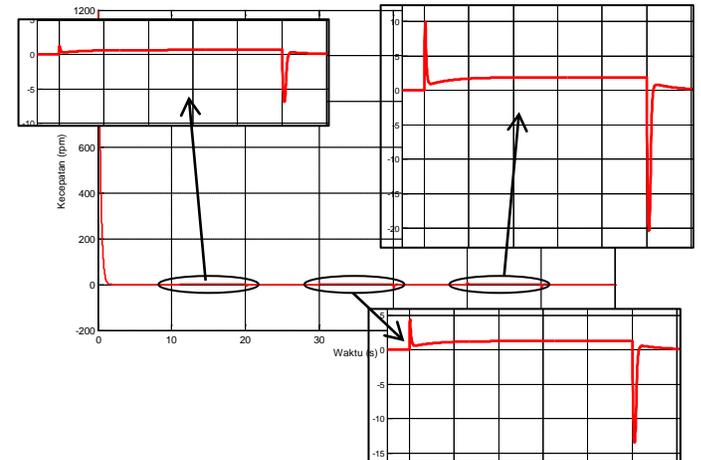
Untuk mengetahui efek pembebanan pada ICE, maka dilakukan simulasi tanpa motor DC. Sinyal masukan step 1200 rpm. Permodelan yang digunakan ICE beban nominal. Beban yang digunakan untuk permasalahan regulator terdapat 3, yaitu 20 VDC setara dengan 58,138 Watt, 40 VDC setara dengan 133,194 Watt dan 60 VDC setara dengan 241,8 Watt seperti yang ditampilkan pada Gambar 11.



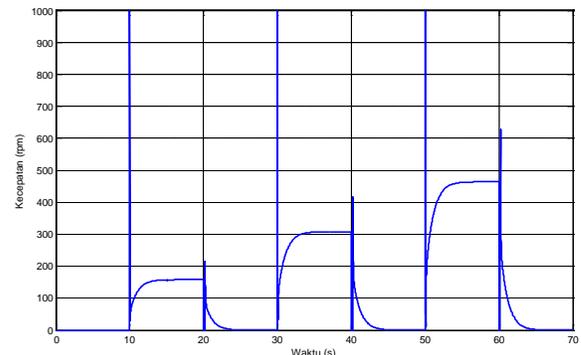
Gambar 11. Respon Kecepatan Putar ICE dengan 3 Variasi Beban



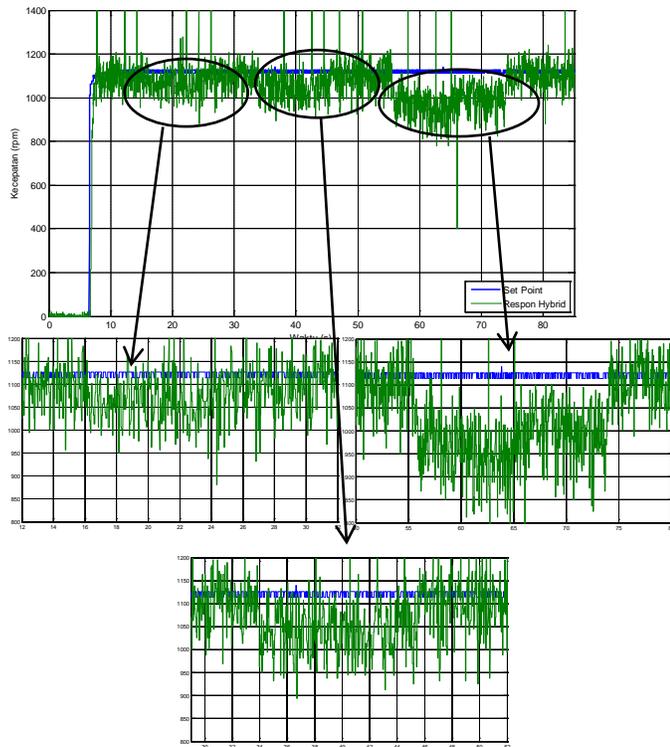
Gambar 12. Respon Kecepatan Putar HEV 3 Variasi Beban dengan Kontroler LQR



Gambar 13 Sinyal Kesalahan dari Sistem Variasi Beban



Gambar 14. Sinyal Kontrol Motor DC Variasi Beban



Gambar 15. Respon Kecepatan Putar HEV dengan Kontroler LQR dengan 3 Variasi Beban

C. Simulasi ICE Berbeban Dengan Bantuan Motor DC Menggunakan Kontroler Linear Quadratic Regulator

Pada Pengujian kontroler pada sistem HEV dilakukan pengujian dengan terdapat 3, yaitu 58,138 Watt, 133,194 Watt dan 241,8 Watt. Ketika terjadi penurunan kecepatan, kontroler LQR mampu mengontrol motor DC untuk membantu ICE agar dapat mempertahankan kecepatan menuju nilai *steady state* dari ICE seperti yang terlihat pada Gambar 12. Sinyal kesalahan dari keseluruhan sistem terhadap *set point* seperti yang ditampilkan pada Gambar 13.

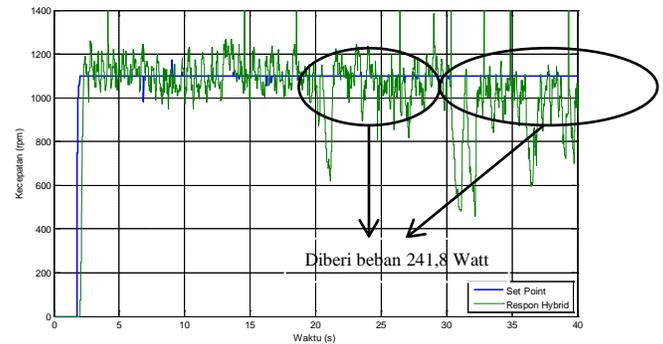
Sinyal kontrol pada motor DC yang digunakan untuk membantu ICE dapat dilihat pada Gambar 14.

Pada pengujian sistem yang pertama didapatkan hasil bahwa motor DC dapat membantu kerja dari ICE sehingga menghilangkan kesalahan sebesar 99,6%. Dan ketika beban dihilangkan, kontroler mampu mengembalikan respon sistem menuju nilai *steady state*.

D. Implementasi ICE Berbeban Dengan Bantuan Motor DC Menggunakan Kontroler Linear Quadratic Regulator

Setelah mengetahui efek pembebanan, kemudian diberikan bantuan motor DC agar kecepatan yang menurun saat terjadi beban dapat dihilangkan menggunakan kontroler LQR seperti pada Gambar 15.

Terlihat dari Gambar 4.22 bahwa ketika beban 58,138 Watt  $e_{ss}$  sebesar 5,5%, beban 133,194 Watt  $e_{ss}$  sebesar 7,7% dan saat beban 241,8 Watt  $e_{ss}$  sebesar 17,5%. Pada beban 241,8 Watt terlihat  $e_{ss}$  yang masih besar maka tegangan masukan untuk *driver* dinaikkan menjadi 100 Vdc sehingga 4.23  $e_{ss}$  menjadi 1,9%.



Gambar 16. Respon Kecepatan Putar HEV dengan Kontroler LQR dengan Beban 241,8 Watt

V. KESIMPULAN

Berdasarkan hasil pengujian dan analisa, Algoritma *Particle Swarm Optimization* (PSO) mampu mendapatkan matriks Q dan R yang optimal dari *tuning* parameter berdasarkan multi kriteria yang ditentukan. Kontrol optimal LQR juga dapat mereduksi sinyal kesalahan karena adanya pembebanan pada sistem HEV. Karena RMSE dan indeks performansi tidak ditemukan titik temu yang dapat diminimumkan keduanya sekaligus maka bobot pada RMSE lebih besar dibandingkan J. Hasil simulasi, motor DC mampu membantu kerja ICE sehingga kecepatan yang diinginkan dapat dipertahankan sebesar 99,4% dari beban maksimal 241,8 Watt yang diberikan pada sistem. Metode kontrol optimal LQR, karena variabel *state* yang selalu tetap, maka hanya mampu melakukan penyelesaian permasalahan regulator pada gangguan pada *state* yang bersifat sementara. Hasil implementasi pada simulator HEV  $e_{ss}$  relatif kecil dimana pada beban nominal 241,8 Watt,  $e_{ss}$  sebesar 1,9% hal tersebut juga bergantung dari tegangan masukan yang diberikan ke *driver*

DAFTAR PUSTAKA

- [1] M. Mohebbi, M. Charkhgard, and M. Farrokhi, "Optimal Neuro-Fuzzy Control of Parallel Hybrid Electric Vehicles", *Vehicle Power and Propulsion, 2005 IEEE Conference*, Pages:26 – 30, 7-9 Sept. 2005
- [2] Galih S.A.W, 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] Wahyu Sriwidodo, 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.
- [4] Ing. Pavel Jacoubek, "Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods", *Conference Studenske Tvuurci-Cinnosti*, 2009
- [5] Naidu, Desineni Subbaram. 2003. "Optimal Control System". CRC Press LLC : New York. 2013.
- [6] J.Hamidi,by, J.R.; Forster, I. "Control System Design Using Particle Swarm Optimization (PSO)", *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-1, Issue-6, January 2012
- [7] Alrijadjis, "Optimasi Parameter Kontroler PID Berbasis Algoritma Particle Swarm optimization (PSO) Untuk Sistem Dengan Waktu Tunda", *Tesis*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010