

Model Predictive Control (MPC) untuk Sistem Motor DC Berbasis LabVIEW

Hasian Syuhada Syahraputra Siregar¹, R M Sisdarmanto Adinandra, S.T., M.Sc, Ph.D.²

Jurusan Teknik Elektro, Universitas Islam Indonesia
Jl Kaliurang KM 14.5 Yogyakarta, Indonesia

¹13524024@students.uui.ac.id

²025240101@uui.ac.id

Dalam dunia pengendalian sangat banyak metode yang digunakan untuk memenuhi karakteristik pengendalian. Dalam setiap metode pengendalian dalam dunia industri maupun keseharian memiliki kelebihan dan kekurangannya. *Model Predictive Control* (MPC) merupakan salah satu metode pengendalian dan merupakan strategi pengendali lanjutan. Sistem miniplant Motor DC merupakan salah satu sistem yang dapat diterapkan oleh pengendalian MPC yang membutuhkan *settling time* yang cepat dan memiliki *overshoot* kecil. LabVIEW merupakan basis pemrograman yang digunakan karena program yang mudah untuk dipahami dan meminimalkan kesalahan. Parameter yang digunakan sebagai proses prediksi dan optimasi pengendalian adalah N_p (*Prediction horizon*) dan N_c (*Control horizon*). Hasil yang sudah sesuai dengan tujuan pengendalian terdapat pada saat set point 1000 RPM $N_p=3$ dan $N_c=1$ dengan *settling time* 2,461 detik dan *overshoot* 5,56 dan $N_p=7$ dan $N_c=6$ dengan *settling time* 2,881 detik dan *overshoot* 6,67 dan saat set point 2000 RPM $N_p=5$ dan $N_c=1$ dengan *settling time* 1,517 detik dan *overshoot* 5,83, $N_p=5$ dan $N_c=2$ dengan *settling time* 2,720 detik dan *overshoot* 5,28 dan $N_p=5$ dan $N_c=3$ dengan *settling time* 0,931 detik dan *overshoot* 5,28.

Kata kunci—MPC; Motor DC; settling time; over shoot; LabVIEW;

I. PENDAHULUAN

Pengendalian dalam sebuah sistem mempengaruhi perkembangan teknologi. Perkembangan teknologi sendiri dapat bersifat sederhana maupun kompleks. Fungsi pengendalian dalam sebuah sistem adalah meningkatkan performa sistem, meminimalkan *error* sistem dan menjaga stabilitas sistem agar sesuai dengan tujuan pengendaliannya.

Sistem motor DC adalah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC memiliki proses yang cepat disebabkan oleh nilai torsi awal motor DC yang besar dan performa kecepatan yang linier.

Dalam penelitian digunakan metode pengendalian *Model Predictive Control* (MPC) dalam sebuah sistem. *Model Predictive Control* (MPC) merupakan strategi pengendalian lanjutan [1]. Pemilihan pengendalian ini karena masih belum banyak penelitian yang menggunakan pengendalian ini dalam sebuah sistem. Keunikan dari pengendalian ini adalah terdapat sistem prediksi hasil pengendaliannya.

Penelitian ini memiliki tujuan untuk merancang MPC dan parameternya dalam sistem Motor DC untuk *settling time* kurang dari sama dengan 3 detik dan *overshoot* kurang dari

10 %. Kemudian membandingkan nilai tersebut dalam implementasi *real-time* dan simulasi.

II. TINJAUAN PUSTAKA

A. Studi Literatur

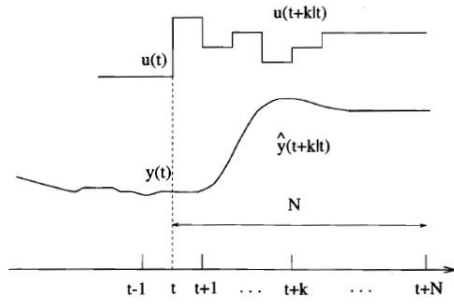
Penelitian menjelaskan tentang analisis pengendalian MPC terhadap performa sistem pemanas udara dengan melihat nilai *settling time*. Perancangan sistem untuk mendapatkan parameter pengendalian *Model Predictive Control* untuk mendapatkan performa sistem terbaik yang dilihat dari *settling time* dan pengujian pengendali *Model Predictive Control* kedalam sistem *real-time*. Nilai *settling time* yang kurang dari 90 detik belum tercapai dalam penelitian ini [2].

Pada penelitian lain menjelaskan tentang analisis pengendalian *finite-states* MPC pada *permanent magnet brushless DC motor* (BLDCM) untuk mengurangi komutasi riak torsi. Hasil penelitian FS-MPC dapat meminimalkan komutasi riak torsi pada BLDCMk [3].

Penelitian lain tentang perbandingan komperatif dalam implementasi LabVIEW pada *Model Predictive Control* (MPC) dengan *PID Controller* untuk *flow control loop*. Perbandingan yang digunakan dalam penelitian ini adalah *settling time* dan *Zero Pick Overshoot*. Hasil dari penelitian MPC *Controller* memiliki akurasi dan respon yang cepat dalam pengendalian dibandingkan *PID Controller* dalam proses [4].

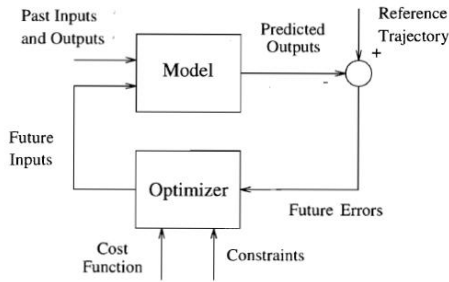
B. Model Predictive Control (MPC)

Model Predictive Control (MPC) adalah salah satu pengendalian yang menggunakan metode prediksi. Munculnya pengendalian prediksi berdasarkan keinginan dalam proses output dalam sebuah sistem selanjutnya dapat diketahui, rangkain perhitungan dalam pengendalian yang akan meminimalakan *objective function* dan setiap keluaran sistem sebelumnya akan masuk dalam perhitungan sebagai keluaran di masa depan atau selanjutnya pada setiap rangkain perhitungan [5].



Gambar 1. MPC Strategy [5]

Metode MPC atau pengendalian prediksi dalam pengendalian dapat dilihat dalam Gambar 1 yang merupakan strategi MPC. Metode dibagi menjadi tiga yaitu sinyal u atau sinyal kendali, sinyal y atau sinyal keluaran dan N atau *horizon*. Prinsip kerjanya yaitu menggunakan model sistem sebagai prediksi keluaran di masa depan dalam rentang N atau *horizon*, menghitung sinyal kendali untuk meminimalkan *objective function* yang telah ditetapkan untuk menjaga sinyal keluaran proses sedekat mungkin dengan karakteristik sistem dan sinyal keluaran berikutnya mengikuti perhitungan yang menggunakan sinyal kendali sebelumnya.



Gambar 2. Struktur pengendalian MPC [5]

Dalam Gambar 2 merupakan struktur dasar dalam pengendalian MPC. Model menggunakan pada prediksi keluarannya berdasarkan nilai sebelumnya dan nilai sekarang dalam pengoptimasian pengendalian di masa depan [5]. Seperti halnya pengendalian lainnya yang membedakan adalah ada 3 masukan yang mengendalikan model yaitu masukan dan keluaran sebelumnya dan prediksi keluaran yang masuk dalam perhitungan.

C. Motor DC

Motor DC adalah salah satu mesin yang mengubah energi listrik ke energi gerak [6]. Kelebihan Motor DC antara lain memiliki daya keluaran yang cukup besar dan penggunaan yang mudah.

Dalam *transfer function* motor DC yaitu [6] :

$$G_m(s) = \frac{\omega(s)}{V(s)} = \frac{K_T}{(sL_a + R)(sJ + D) + K_E K_T} \quad (1)$$

K_T = torsi konstan

L_a = induktansi

R = resistansi

J = *moment of inertia*

D = *viscous damping factor*

K_E = tegangan konstan

D. LabVIEW

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) adalah sebuah software pemrograman berbasis instrumen virtual yang terdiri dari komputer, perangkat lunak dan perangkat keras yang digabungkan dan dikonfigurasi untuk menyamai fungsi instrumentasi tradisional [7]. MPC dalam LabVIEW menggunakan beberapa VI yaitu "CD Create MPC Controller VI", "CD Step Forward MPC Window VI" dan "CD Implement MPC Controller VI". "CD Create MPC Controller VI" memiliki fungsi membuat pengendalian *Model Predictive Control* (MPC) untuk model *state-space*. "CD Step Forward MPC Window VI" memiliki fungsi menghitung nilai *set-point* dan/atau nilai *disturbance*. "CD Implement MPC Controller VI" memiliki fungsi menghitung *Control Action* ($u(k)$) yang diterapkan ke *plant*

III. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini terbagi atas pemodelan Motor DC, pengendalian MPC Motor DC (indeks performa MPC dan parameter MPC) dan implementasi MPC pada sistem Motor DC pada LabVIEW secara simulasi dan *real-time*.

Pemodelan sistem Motor DC merupakan proses mencari persamaan matematis dari sistem Motor DC yang berupa fungsi alih dan kemudian diubah menjadi *state space*. *Transfer function* sistem *miniplant* Motor DC dalam penelitian didapatkan dengan persamaan All-Pole Second Order Transfer Function dan metode Ziegler-Nichols [8]:

$$\frac{C(s)}{R(s)} = K \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2)$$

K = Gain overall

ω_n = Frekuensi alami

ζ = Rasio redaman

Berdasarkan persamaan 2 didapatkan transfer function sistem Motor DC dalam penelitian:

$$G_m(s) = \frac{3.39}{0.005s^2 + 0.0467s + 0.0522} \quad (3)$$

Pengendalian MPC sistem Motor DC dapat dibagi menjadi yaitu pengaturan parameter dan indeks performanya. Pengaturan parameter dalam MPC merupakan proses prediksi dan optimasi pengendalian. Parameter tersebut adalah N_p (*Prediction horizon*) dan N_c (*Control horizon*). Parameter ini memiliki ketentuan untuk menentukan nilainya, N_p dan N_c harus berupa bilangan bulat, nilai N_c tidak boleh besar dari nilai N_p dan nilai maksimal adalah N_c sama dengan nilai N_p . Indeks performa dalam MPC merupakan *cost function*

dengan meminimalkan nilainya. *Cost function* merupakan sebuah formula atau persamaan, cerminan perubahan *error* terhadap referensi dan aksi pengendalian. *Cost function* atau indeks performa yang digunakan dalam penelitian ini adalah[9]:

$$J(u, k) = \sum_{j=N_m}^N \hat{x}^T(k+j|k) Q \hat{x}(k+j|k) + \sum_{j=1}^N u^T(k+j-1|k) R u(k+j-1|k) \quad (3)$$

$J(u, k)$ = indeks performa output.

$x(k)$ = sinyal *state*

$u(k)$ = sinyal kendali

Q = matrix bobot *state*

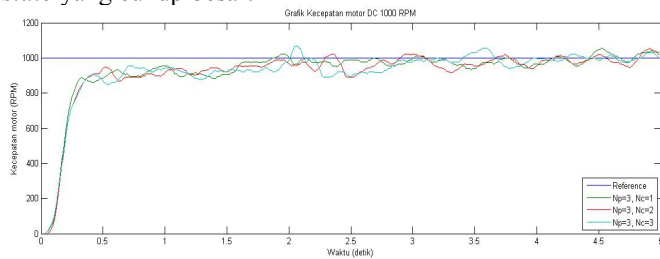
R = matrix bobot pengendali

Setelah pemodelan sistem, mengatur indeks performa dan parameter MPC langkah selanjutnya implementasikan pengendalian MPC pada sistem Motor DC secara simulasi dan *real-time*. Kemudian menganalisis nilai *settling time* dan *overshoot* dari kedua implementasi MPC Motor DC terhadap perubahan parameter.

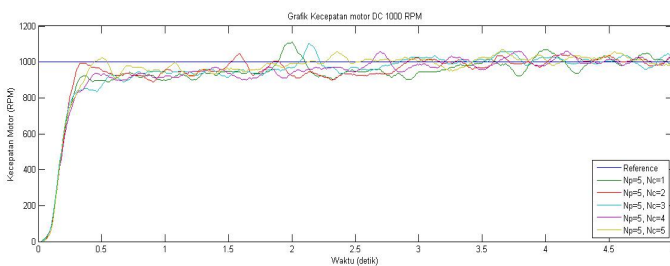
IV. HASIL DAN ANALISIS

A. Pengujian Motor DC

Pada pengujian dilakukan pada 3 kondisi yaitu pada kecepatan Motor DC 1000, 1500 dan 2000 RPM. Pengujian dilakukan sebanyak 15 data dengan berbeda parameter N_p dan N_c . Pada Gambar 3 dan Gambar 4 menunjukkan bahwa hasil yang diperoleh sudah sesuai dengan sinyal referensi 1000 RPM. Kekurang dari hasil yang diperoleh adalah masih memiliki nilai berubah-ubah setelah menuju keadaan *steady-state* yang cukup besar.



Gambar 3. Hasil pengujian pada 1000 RPM untuk Nilai $N_p=3$ dan $N_c=1, 2$, dan 3



Gambar 4. Hasil pengujian pada 1000 RPM untuk Nilai $N_p=5$ dan $N_c=1, 2, 3, 4$ dan 5

B. Pengujian untuk Kecepatan Motor DC 1000, 1500 dan 2000 RPM

Pada pengujian kecepatan Motor DC 1000, 1500 dan 2000 RPM dilakukan sebanyak 15 pengujian untuk setiap nilai parameter N_p dan N_c yang telah ditunjukkan pada Tabel 3.1. Pengujian simulasi dan *real-time* akan dibandingkan dari nilai *settling time* kurang dari sama dengan 3 detik dan *overshoot* kurang dari 10 %. Berikut ini tabel hasil pengujian:

Tabel 1. Hasil *settling time* dan *overshoot* pengujian pada 1000 RPM

N_p	N_c	Settling Time (s) Simulasi	Overshoot (%) Simulasi	Settling Time (s) Real-time	Overshoot (%) Real-time
3	1	3	0	2,461	5,56
3	2	3	0	5,032	8,33
3	3	3	0	5,618	8,33
5	1	3	0	4,260	11,11
5	2	3	0	3,772	9,44
5	3	3	0	4,126	10,56
5	4	3	0	4,649	7,78
5	5	3	0	3,452	6,67
7	1	3	0	3,528	6,67
7	2	3	0	4,023	8,33
7	3	3	0	4,983	8,33
7	4	3	0	2,841	11,11
7	5	3	0	3,841	8,33
7	6	3	0	2,881	6,67
7	7	3	0	3,841	11,11

Tabel 2. Hasil *settling time* dan *overshoot* pengujian pada 1500 RPM

N_p	N_c	Settling Time (s) Simulasi	Overshoot (%) Simulasi	Settling Time (s) Real-time	Overshoot (%) Real-time
3	1	3	0	4,439	34,44
3	2	3	0	4,023	26,3
3	3	3	0	2,923	20
5	1	3	0	2,242	21,85
5	2	3	0	3,878	30
5	3	3	0	2,048	25,19
5	4	3	0	0,69	34,07
5	5	3	0	1,016	32,96
7	1	3	0	2,933	31,11
7	2	3	0	2,278	22,96
7	3	3	0	3,093	29,63
7	4	3	0	1,182	28,52
7	5	3	0	2,867	25,93
7	6	3	0	2,173	44,81
7	7	3	0	2,586	26,30

Tabel 3. Hasil *settling time* dan *overshoot* pengujian pada 2000 RPM

Np	Nc	Settling Time (s) Simulasi	Overshoot (%) Simulasi	Settling Time (s) Real-time	Overshoot (%) Real-time
3	1	3	0	3,168	6,39
3	2	3	0	4,009	10
3	3	3	0	2,66	12,5
5	1	3	0	1,517	5,83
5	2	3	0	2,72	5,28
5	3	3	0	0,931	5,28
5	4	3	0	3,02	8,61
5	5	3	0	2,608	10,83
7	1	3	0	3,709	6,94
7	2	3	0	3,53	5,28
7	3	3	0	4,277	6,67
7	4	3	0	2,093	15,28
7	5	3	0	2,45	7,81
7	6	3	0	2,972	5
7	7	3	0	3,652	8,33

Analisis pengujian berdasarkan Tabel 1, Tabel 2 dan Tabel 3 pengaruh bertambahnya nilai N_p membuat *overshoot* menjadi bertambah akan tetapi untuk sebagian besar parameter dan pengaruh bertambahnya nilai N_c membuat *overshoot* berkurang akan tetapi untuk sebagian besar parameter. Pengaruh bertambahnya nilai N_p membuat *settling time* menjadi melambat akan tetapi untuk sebagian besar parameter dan pengaruh bertambahnya nilai N_c membuat *settling time* berkurang akan tetapi untuk sebagian besar parameter. Nilai *settling time* dan *overshoot* simulasi dan real-time masih memiliki selisih yang cukup besar.

Hasil yang sudah sesuai dengan tujuan pengendalian terdapat pada $N_p = 3$ dan $N_c = 1$, saat *set point* 1000 RPM (*settling time* 2,461 detik dan *overshoot* 5,56), $N_p = 7$ dan $N_c = 6$, saat *set point* 1000 RPM (*settling time* 2,881 detik dan *overshoot* 6,67), $N_p = 5$ dan $N_c = 1$, saat *set point* 2000 RPM (*settling time* 1,517 detik dan *overshoot* 5,83), $N_p = 5$ dan $N_c = 2$, saat *set point* 2000 RPM (*settling time* 2,720 detik dan *overshoot* 5,28) dan $N_p = 5$ dan $N_c = 3$, saat *set point* 2000 RPM (*settling time* 0,931 detik dan *overshoot* 5,28).

V. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan dapat diambil kesimpulan sebagai berikut:

1. Pemodelan pada pengendalian MPC sistem Motor DC sangat mempengaruhi hasil dikarenakan semakin dekat persamaan model dengan sistemnya akan mempercepat proses dan memperkecil adanya *overshoot*.
2. Pengendalian MPC hasil proses dipengaruhi oleh *cost function*, *constraint*, parameter N_p (*Prediction horizon*) dan N_c (*Control horizon*).

3. Hasil yang sudah sesuai dengan tujuan pengendalian terdapat pada :

- a) $N_p = 3$ dan $N_c = 1$, saat *set point* 1000 RPM (*settling time* 2,461 detik dan *overshoot* 5,56).
- b) $N_p = 7$ dan $N_c = 6$, saat *set point* 1000 RPM (*settling time* 2,881 detik dan *overshoot* 6,67).
- c) $N_p = 5$ dan $N_c = 1$, saat *set point* 2000 RPM (*settling time* 1,517 detik dan *overshoot* 5,83).
- d) $N_p = 5$ dan $N_c = 2$, saat *set point* 2000 RPM (*settling time* 2,720 detik dan *overshoot* 5,28).
- e) $N_p = 5$ dan $N_c = 3$, saat *set point* 2000 RPM (*settling time* 0,931 detik dan *overshoot* 5,28).

DAFTAR PUSTAKA

- [1] S. Abirami, A.S. Zahir hussain, D. Saravana Muth dan C.B. Aravind kumar, "Performance comparison of different controllers for a level process," *Int. Journal of Engineering Research and Applications*, vol. 4, pp. 341-344, 2014.
- [2] A. Baskoro, "Model Predictive Control(MPC) untuk Sistem Pemanas Udara Berbasis MATLAB," *Tugas Akhir, Universitas Islam Indonesia* ., 2014.
- [3] C. Xia, Y. Wang dan T. Shi, "Implementation of Finite-States Model Predictive Control for Commutation Torque Ripple Minimization of Permanent Magnet Brushless DC Motor," *IEEE Transaction on Industrial Electronics, IEEE* ., vol. 60, pp. 896-905, 2013.
- [4] C. Jichkar dan S. Sondkar, "Comparative study of real time implementation of LabVIEW based MPC controller and PID controller for flow control loops," *2nd International Conference for Convergence in Technology (I2CT), IEEE* ., pp. 464-470, 2017.
- [5] E. Camacho dan C. Bardons, "Model Predictive Control," *Springer*., 2006.
- [6] Electro-Craft Corporation, "DC Motor Speed Controls Servo Systems", 3rd ed. United State of America : Permagon Press, 2009.
- [7] J. Travis dan J. Kring, "LabVIEW for Everyone: Graphical Programming Made Easy and Fun, Third Edition," *Prentice Hall*., pp. 6964–6975, 2006.
- [8] J. Ziegler dan J. Nichols, "Optimum Settings for Automatic Controllers" *Transaction of The A.S.M.E.*, pp. 759–765, 1942.
- [9] A. Bemporad, M. Morari and N. L. Ricker, *Model Predictive Control Toolbox™ User's Guide*, The MathWorks, Inc., 2013