

# Analisa Metode *Cross-Coupling Generalized Predictive Control* untuk Mengurangi *Contour Error* pada Mesin CNC

Muhammad Fathu Sunu Mukti dan Rusdhianto Effendi AK  
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: ditto@ee.its.ac.id

**Abstrak**—*Computerized Numerical Control (CNC)* semakin banyak diminati di bidang industri manufaktur [1]. Pada umumnya, CNC dibutuhkan untuk dapat menghasilkan pergerakan yang sinkron diantara tiap sumbu geraknya untuk mengikuti suatu lintasan yang telah ditentukan. Tantangan utamanya adalah bagaimana cara untuk mengeliminasi *contour error* daripada menghilangkan *error* pada masing-masing sumbu [2]. Penelitian ini mengajukan desain *cross-coupling* dari *Generalized Predictive Control* untuk mesin CNC dengan sistem servo dua sumbu gerak. Untuk mendapatkan sinkronisasi dan respon *tracking* yang baik, metode *Cross-Coupled Control*, yang cukup banyak digunakan untuk mengurangi *contour error*[5], dikombinasikan dengan metode *Generalized Predictive Control*. Metode *Cross-Coupling Generalized Predictive Control (CC-GPC)* akan dibandingkan dengan metode *modified Cross-Coupled Control (M-CCC)* [3] untuk menunjukkan metode kontrol mana yang menghasilkan performa paling memuaskan untuk mengurangi *contour error*. Simulasi menunjukkan keuntungan dari metode CC-GPC adalah dapat meningkatkan respon dari sistem motor servo AC. *Rise time* ( $t_r$ ) menurun menjadi 1.6 detik dan *error steady state* ( $e_{ss}$ ) berkurang menjadi 0.49. Keuntungan lain adalah koreksi dari tiap sumbu terjadi secara bersamaan, sehingga sehingga sistem memiliki kemampuan yang baik untuk menahan gangguan. Dari pengujian gangguan, hasil menunjukkan *Root Mean Square Error (RMSE)* dari metode CC-GPC adalah 0.3806. Cukup baik dibanding RMSE metode M-CCC yang sebesar 1.0478.

**Kata Kunci**—*Computerized Numerical Control, Contour Error, Cross-Coupled Control, Generalized Predictive Control.*

## I. PENDAHULUAN

KEAKURATAN dari mesin *Computerized Numerical Control (CNC)* untuk mengikuti suatu lintasan yang telah ditentukan adalah kriteria yang sangat penting dalam menentukan kinerja dari mesin [1]. Konfigurasi antar sumbu merupakan faktor yang penting sangat berpengaruh dalam menentukan keakuratan pembentukan suatu *contour*. *Contour* dibentuk berdasarkan pergerakan dari tiap-tiap unsur pembentuk lintasan. Dalam pergerakan tersebut, apabila terjadi ketidaksesuaian pergerakan akan berakibat terjadinya penyimpangan. Hal ini disebut dengan *contour error*, yaitu penyimpangan antara lintasan referensi yang diinginkan dengan lintasan aktual yang dihasilkan [4]. Ketika tiap sumbu melacak koordinat lintasan tertentu yang sudah ditentukan, maka pergerakan masing-masing sumbu akan berpengaruh terhadap pembentukan lintasan tersebut. Pergerakan tiap sumbu perlu dikoordinasi agar kontur yang dihasilkan sesuai dengan yang diinginkan. Sebelumnya, pada sistem servo multi-sumbu, tiap sumbu dikontrol terpisah secara konvensional [2]. *Cross-Coupled Control (CCC)* merupakan

salah satu metode yang digunakan untuk mengontrol suatu sistem dengan beberapa sumbu gerak secara terkoordinasi. Konsep ini telah berhasil diterapkan untuk mengurangi *contour error* dalam pengendalian gerakan dari sistem multi-sumbu. Akan tetapi, apabila metode ini dihadapkan dengan perubahan parameter dan kondisi sistem yang nonlinear, maka solusi ini kurang ideal. Beberapa terapan dari metode CCC telah diajukan, namun kelemahan utama metode ini adalah tidak dapat secara langsung beradaptasi atas perubahan pada parameter sistem dan penambahan gangguan sehingga tidak dapat memberikan performa yang memuaskan.

Metode kontrol prediktif telah diterima secara luas dan telah banyak di aplikasikan di industri pada beberapa tahun terakhir [6]. Salah satu metode kontrol prediktif paling populer adalah *Generalized Predictive Control (GPC)*. Metode ini populer karena kelebihanannya yang lebih sederhana dan ketajaman kontrolnya. Terutama untuk aplikasi pada sistem yang nonlinear dan membutuhkan performa tinggi, GPC memiliki *robustness* yang lebih baik.

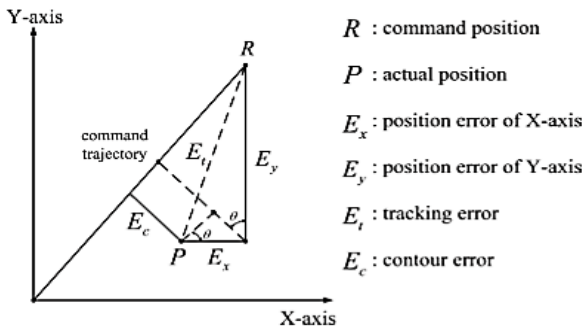
Pada penelitian ini akan diajukan perancangan kontroler *Cross-Coupling Generalized Predictive Control (CC-GPC)* untuk sistem dengan dua sumbu gerak. Metode ini diharapkan dapat mengurangi masalah *contour error* yang terjadi pada sistem multi-sumbu seperti mesin CNC.

## II. TEORI PENUNJANG

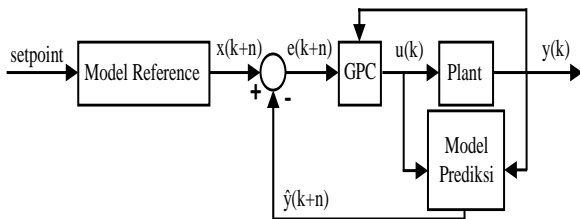
### A. *Computerized Numerical Control*

*Numerical Control (NC)* merupakan salah satu metode untuk mengendalikan operasi dari sebuah mesin. Pengendalian tersebut menggunakan program yang mengandung data numerik untuk menentukan pergerakan sumbu menuju lintasan yang diinginkan. *Computerized Numerical Control (CNC)* merupakan pengembangan dari teknologi NC dimana digunakan komputer berbasis perangkat lunak dan *driver*. Kemudian teknologi CNC semakin semakin berkembang sehingga kapabilitas dan kehandalan mesin dalam sebuah proses produksi menjadi lebih tinggi. Mesin CNC adalah mesin perkakas yang digunakan untuk mengikis, memotong, atau membentuk sebuah benda. Mesin CNC adalah pengembangan dari mesin perkakas yang menggunakan program dalam proses membentuk sebuah benda.

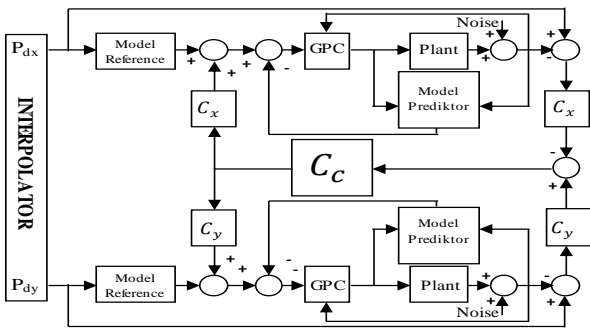
Pada mesin CNC terdapat bagian yang berputar yang disebut dengan *spindle*. Berdasarkan bagian tersebut, mesin CNC dapat dibedakan menjadi dua jenis yaitu mesin CNC *lathe* dan *milling*. Pada mesin CNC *lathe*, benda kerja



Gambar 1. Perbedaan Tracking Error dan Contour Error.



Gambar 2. Diagram Blok Kontroler GPC.



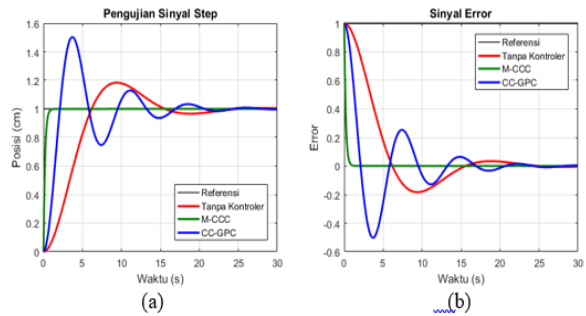
Gambar 3. Diagram Blok Kontroler CC-GPC.

terdapat pada bagian *spindle*. Mesin CNC *lathe* digunakan untuk mengikis atau mengukir benda kerja tersebut sehingga menjadi bentuk yang diinginkan. Untuk mesin CNC *milling*, terdapat dua jenis penempatan motor *spindle*, yaitu *spindle* yang bergerak saat penggerak sumbu X dan Y membentuk benda kerja dan motor *spindle* yang diam tetapi benda kerjanya yang bergerak dan digerakkan oleh motor servo AC sebagai penggerak.

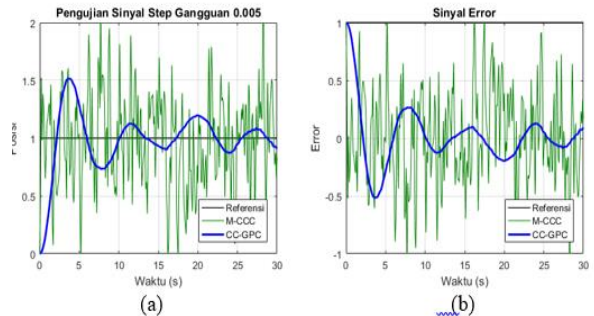
**B. Contour Error**

*Contour Error* merupakan istilah dari penyimpangan yang terjadi karena adanya perbedaan antara lintasan referensi dengan lintasan yang dihasilkan sebenarnya dalam proses pembentukan kontur. Ilustrasi tersebut ditunjukkan pada Gambar 1. Dimana R sebagai titik referensi dari pergerakan akhir kontur yang diinginkan. Titik P merupakan penyimpangan dari lintasan.  $E_x$  merupakan penyimpangan dari sumbu X dan  $E_y$  merupakan penyimpangan dari sumbu Y,  $E_t$  merupakan *tracking error*, dan  $E_c$  merupakan *contour error*. Lintasan yang ditunjukkan dengan garis adalah lintasan yang diinginkan, sedangkan lintasan yang sebenarnya ditunjukkan pada garis putus-putus.

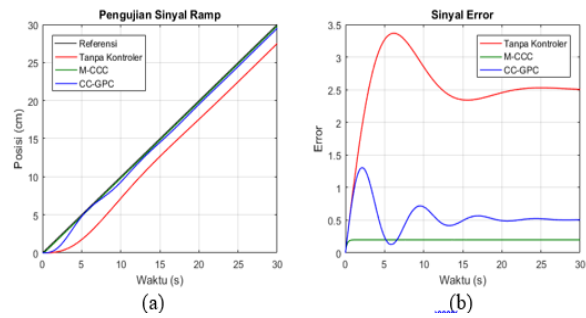
Untuk meningkatkan akurasi dalam pembentukan kontur, dapat digunakan kontroler untuk *tracking* dan kontroler untuk proses pembentukan kontur. Untuk kontroler *tracking*, penggunaannya adalah untuk mengurangi atau menghilangkan penyimpangan yang terjadi berdasarkan masing-masing sumbu atau pada bagian  $E_x$  dan  $E_y$ . Sedangkan untuk kontroler pada kontur, digunakan untuk mengurangi atau



Gambar 4. (a) Respon Uji Step (b) Sinyal Error.



Gambar 5. (a) Uji Step Gangguan 0.05 mm (b) Sinyal Error.



Gambar 6. (a) Respon Uji Ramp (b) Sinyal Error.

menghilangkan penyimpangan secara langsung tanpa memperhatikan apakah terjadi pengurangan penyimpangan pada  $E_x$  dan  $E_y$ . Namun kontroler dapat mengurangi penyimpangan masing-masing sumbu sekaligus kesalahan kontur secara langsung.

**C. Generalized Predictive Control (GPC)**

Metode *Generalized Predictive Control* (GPC) yang diperkenalkan oleh Clarke et al [[6][7] banyak digunakan pada proses dalam industri. Hal ini disebabkan kontroler GPC memiliki kemampuan pengaturan yang stabil untuk sistem dengan parameter, *dead time*, orde model yang berubah ubah, dan ketahanan terhadap gangguan yang berupa nilai acak. GPC umumnya menggunakan pemodelan sistem *Controlled Auto-Resgresive Integrated Moving Average* (CARIMA) yang memasukkan model gangguan.

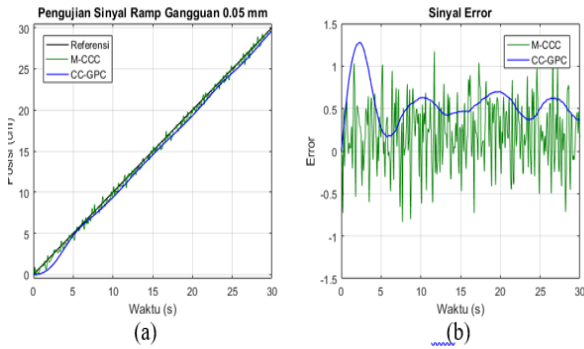
$$A(q^{-1})y(t) = B(q^{-1})u(t - 1) + C(q^{-1}) \frac{\xi(t)}{\Delta} \quad (1)$$

dimana  $u(t)$  adalah *input* kontrol,  $y(t)$  adalah *output*, dan  $\xi(t)$  adalah gangguan acak. Di mana A, B, dan C adalah polinomial dari operator *backward*  $q^{-1}$ .

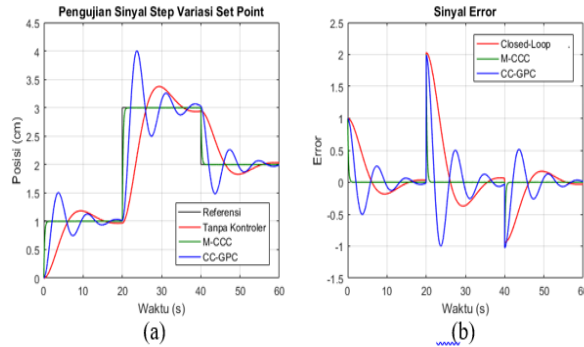
$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \quad (2)$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_{nb}q^{-nb} \quad (3)$$

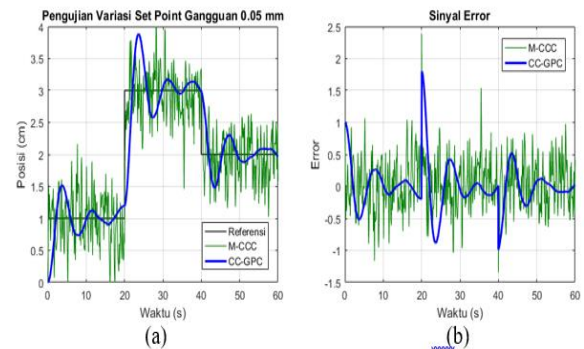
$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} \quad (4)$$



Gambar 7. (a) Uji Ramp Gangguan 0.05 mm (b) Sinyal Error.



Gambar 8. (a) Respon Uji Set Point (b) Sinyal Error.



Gambar 9. (a) Uji Set Point Gangguan 0.05 mm (b) Sinyal Error.

Dengan model linier, yang membatasi kriteria kontrol optimal kuadrat diajukan dalam kenaikan *input output plant*. Fungsi harga yang diminimumkan yaitu

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \quad (5)$$

dimana  $N_1$  adalah *costing horizon* minimal,  $N_2$  adalah *costing horizon* maksimal,  $\lambda(j)$  adalah urutan kontrol pembebanan. Sinyal *output* prediksi  $\hat{y}(k+d+1+n_c)$  dituliskan sebagai penjumlahan dari *free response* dan *force response*. Persamaan *force response* adalah sebagai berikut :

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

Tabel 1. Nilai RMSE Pengujian Sinyal Step

Variasi Gangguan (mm)	RMSE	
	M-CCC	CC-GPC
0	0.0738	0.1492
0.005	0.3827	0.1697
0.05	1.2035	0.2818
0.1	1.7018	0.3588

Tabel 2. Nilai RMSE Pengujian Sinyal Ramp

Variasi Gangguan (mm)	RMSE	
	M-CCC	CC-GPC
0	0.1994	0.5227
0.005	0.3908	0.5297
0.05	1.0773	0.5809
0.1	1.5097	0.6371

Tabel 3. Nilai RMSE Pengujian Sinyal Step Variasi Set Point

Variasi Gangguan (mm)	RMSE	
	M-CCC	CC-GPC
0	0.1393	0.2925
0.005	0.3957	0.3092
0.05	1.2101	0.3191
0.1	1.7076	0.3700

Tabel 4. Nilai RMSE Pengujian Pembentukan Kontur Linier

Variasi Gangguan (mm)	RMSE	
	M-CCC	CC-GPC
0	2.1700e-7	5.5237e-06
0.005	0.2749	0.1048
0.05	0.8686	0.3313
0.1	1.2283	0.4702

*Free Response* merupakan respons hasil pengaruh dari *input* lampau pada *output* yang akan datang dengan asumsi aksi kontrol dari waktu saat ini.

$$\begin{aligned} \hat{y}_{fr}(k+d+1+n_c|k) &= \frac{B(q^{-1})}{A(q^{-1})} u(k+1+n_c|k-1) \\ \Delta u(k+i) &= 0; i = 0,1,2, \dots \end{aligned} \quad (7)$$

Vektor dari *output* prediksi adalah penjumlahan dari respons prediksi *force* dan *free*,

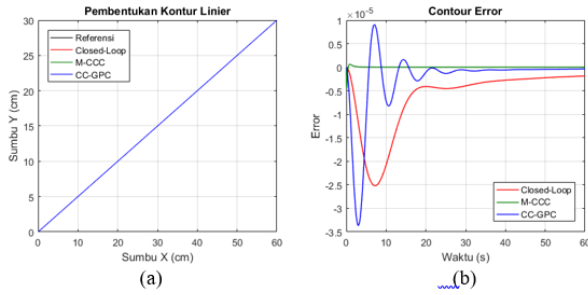
$$\hat{y} = \hat{y}_{fc} + \hat{y}_{fr} \quad (8)$$

Output *force response* dapat ditunjukkan dengan

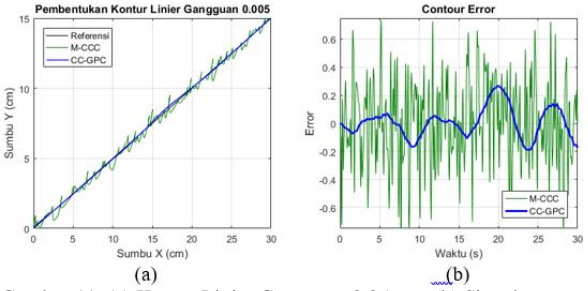
$$\hat{y}_{fc} = H \Delta u_f \quad (9)$$

$$\Delta u_f = [\Delta u(k|k), \Delta u(k+1|k), \dots, \Delta u(k+n_{c2}-1|k)]^T \quad (10)$$

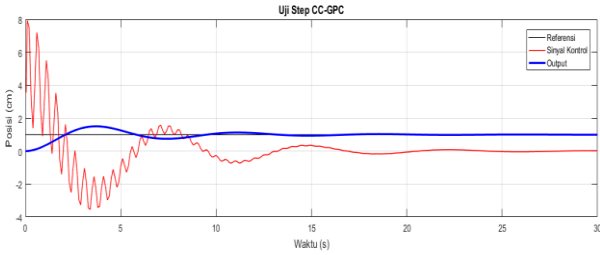
dan  $H(q^{-1})$  adalah fungsi alih pulsa dari proses yang diperpanjang dengan integrator tanpa *dead time* fisik. Variabel manipulasi untuk saat ini dan akan datang disebut  $\Delta u(k+i), i \geq 0$  didesain dengan waktu diskrit, dan bisa ditulis  $\Delta u(k+i|k)$ . Untuk lebih mudahnya penulisan  $\Delta u(k+i|k)$  hanya ditulis  $\Delta u(k+i)$ .  $H$  merupakan koefisien respons *unit step*.



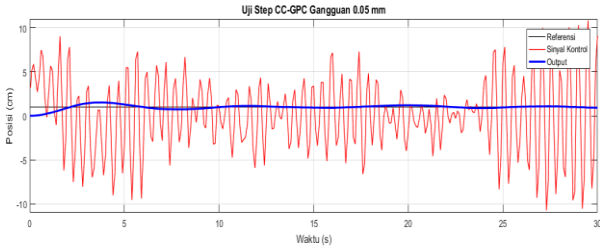
Gambar 10. (a) Pembentukan Kontur Linier (b) Sinyal Error



Gambar 11. (a) Kontur Linier Gangguan 0.05 mm (b) Sinyal Error.



Gambar 12. Sinyal Kontrol CC-GPC saat Uji Step.



Gambar 13. Sinyal Kontrol CC-GPC saat Uji Step Gangguan 0.05 mm.

$$H = \begin{bmatrix} h_{nc1+1} & h_{nc1} & \dots & h_{nc1-n_u+2} \\ h_{nc1+2} & h_{nc1+1} & \dots & h_{nc1-n_u+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{nc2+1} & h_{nc1} & \dots & h_{nc2-n_u+2} \end{bmatrix} \quad (11)$$

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

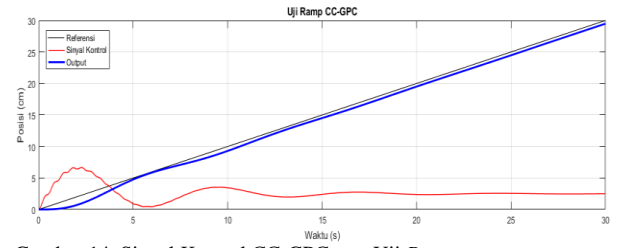
$$\Delta u(k) = [1, 0, \dots, 0][H^T H + \lambda_u I]^{-1} H^T (y_r - \hat{y}_{free}) \quad (12)$$

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

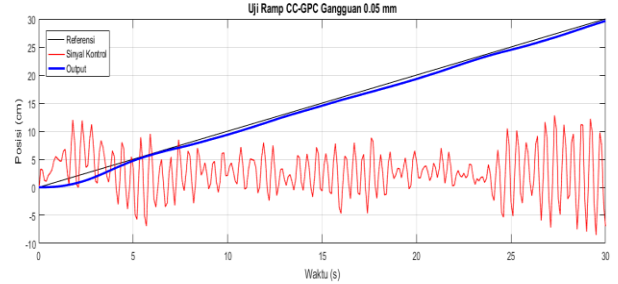
### III. PERANCANGAN SISTEM

#### A. Pemodalan Matematis Motor Servo AC

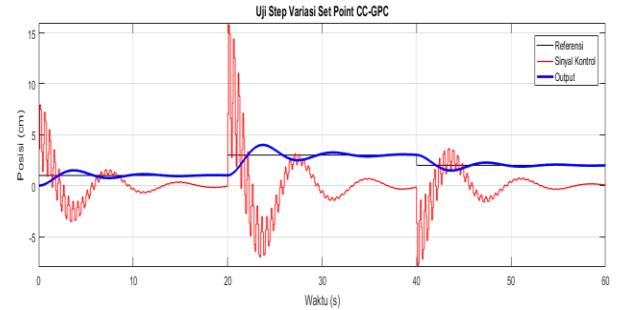
Persamaan fungsi alih orde dua dari model matematika motor servo AC diperoleh melalui persamaan antara torsi



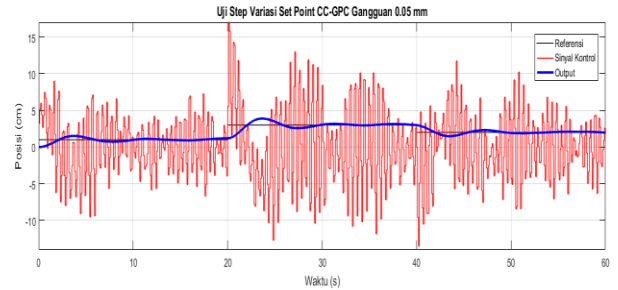
Gambar 14. Sinyal Kontrol CC-GPC saat Uji Ramp.



Gambar 15. Sinyal Kontrol CC-GPC saat Uji Ramp Gangguan 0.05 mm.



Gambar 16. Sinyal Kontrol CC-GPC saat Uji Step Variasi Set Point.



Gambar 17. Sinyal Kontrol CC-GPC saat Uji Step Variasi Set Point Gangguan 0.05 mm.

motor dan beban motor [8]. Seperti pada persamaan dibawah yang merupakan persamaan dari torsi motor.

$$T_m = K_c e_c - K_n \frac{d\theta}{dt} \quad (13)$$

dengan  $e_c$  merupakan tegangan pada fasa kontrol motor servo,  $\frac{d\theta}{dt}$  adalah posisi sudut motor servo.  $K_c$  dan  $K_n$  adalah konstanta motor. Sedangkan untuk persamaan dari torsi pada beban motor,

$$T_w = J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} \quad (14)$$

dengan  $\frac{d^2\theta}{dt^2}$  merupakan kecepatan sudut dari motor. Untuk  $J$  dan  $B$  merupakan momen inersia dan koefisien gesek viskos dari motor. Kemudian dengan mensubstitusikan Persamaan (13) dan Persamaan (14), bisa didapat persamaan,

$$J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} = K_c e_c - K_n \frac{d\theta}{dt} \quad (15)$$

$$[Js^2 + Bs + K_n s]\theta(s) = K_c E_c(s) \quad (16)$$

Sehingga didapat fungsi alih motor servo AC,

$$\frac{\theta(s)}{E_c(s)} = \frac{\left(\frac{K_c}{Bs + K_n}\right)}{s \left(\frac{J}{Bs + K_n}s + 1\right)} \quad (17)$$

$$\frac{\theta(s)}{E(s)} = \frac{k_m}{s(\tau_m s + 1)} \quad (18)$$

dengan  $\tau_m$  adalah konstanta waktu motor dan  $k_m$  adalah konstanta penguatan motor. Dari spesifikasi yang didapat dari referensi [8], dapat diperoleh fungsi alih dari motor servo AC yang ditunjukkan pada persamaan dibawah ini.

$$G(s) = \frac{0.4}{s(2.776s + 1)} \quad (19)$$

### B. Penentuan Prediction Horizon dan Control Horizon

Jika *dead-time* dari sistem diketahui, waktu tersebut bisa menjadi nilai minimum dari *prediction horizon*. Sedangkan nilai maksimum *prediction horizon* mendekati nilai *rise time* dari sistem. Nilai *prediction horizon* yang terlalu besar bisa membuat respon sistem menjadi lambat walaupun tidak terdapat *overshoot* [6]. Untuk menentukan *control horizon* nilainya berada diantara minimum dan maksimum *prediction horizon*. Menggunakan model matematika pada Persamaan (19) diketahui parameter sistem  $\omega_n = 0.38$  dan  $\xi = 0.47$ . Sistem tergolong *underdamped* karena memiliki nilai  $0 < \xi < 1$ . Kemudian dapat dihitung nilai *rise time*,

$$t_r = \frac{1 - 0.4167\xi + 2.917\xi^2}{\omega_n}; 0 < \xi < 1 \quad (20)$$

Dan didapat nilai  $t_r = 3.8s$ . Dengan nilai *rise time* diketahui, maka dapat ditentukan panjang *prediction horizon* dan *control horizon* sebesar tiga.

### C. Desain Kontroler GPC

Diagram blok dari kontroler GPC dapat dilihat pada Gambar 2. Model referensi dari hasil *input* masa depan yang diinginkan akan dibandingkan dengan *output* masa depan yang diprediksi oleh model prediktor. Kemudian perkiraan *error* masa depan yang akan terjadi digunakan oleh kontroler GPC untuk menentukan respon sinyal kontrol yang akan digunakan pada sistem. Kontroler GPC ini akan disusun berdasarkan kinerja dari motor servo AC. Dari model matematika motor servo AC, maka dilakukan pemodelan yang sesuai dengan GPC yaitu ARMA (*Auto Regressive Moving Average*). Model matematika motor servo AC adalah seperti pada Persamaan (19). Bentuk ARMA didapatkan dari penyusunan blok identifikasi ARMAX pada *simulink* Matlab dengan modifikasi. Ditentukan orde sistem 2, orde input 2, time sampling sebesar 0.1, dengan dijalankan selama 100s maka didapatkan hasil permodelan ARMA dibawah ini.

$$\frac{Y(z)}{U(z)} = \frac{0.0007119z + 0.0007034}{z^2 - 1.965z + 0.9646} \quad (21)$$

$$y(k) = 1.965y(k-1) - 0.9645y(k-2)$$

$$+ 0.0007119u(k-1) - 0.0007034u(k-2) \quad (22)$$

Setelah model ARMA ditemukan dan panjang *prediction horizon* dan *control horizon* telah ditentukan, langkah selanjutnya yaitu menurunkan persamaan beda pada Persamaan (3.42) sampai mendapatkan  $y(k + NU)$  dengan  $NU = 3$  sesuai panjang *horizon*.

$$y(k) = b_1y(k-1) + b_2y(k-2) + a_1u(k-1) + a_2u(k-2) \quad (23)$$

$$y(k+1) = b_1y(k) + b_2y(k-1) + a_1u(k) + a_2u(k-1) \quad (24)$$

$$y(k+2) = (b_1^2 + b_2)y(k) + b_1b_2y(k-1) + a_1u(k+1) + (b_1a_1 + a_2)u(k) + b_1a_2u(k-1) \quad (25)$$

$$y(k+3) = (b_1^3 + 2b_1b_2)y(k) + (b_1^2b_2 + b_2^2)y(k-1) + a_1u(k+2) + (b_1a_1 + a_2)u(k+1) + (b_1^2a_1 + b_2a_1 + b_1a_2)u(k) + (b_1^2b_2 + b_2^2)u(k-1) \quad (26)$$

Dari persamaan diatas dapat dibentuk matriks,

$$\hat{y} = \hat{y}_{force} + \hat{y}_{free} \quad (27)$$

$$\begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix} = \mathbf{H} \begin{bmatrix} u(k) \\ \hat{u}(k+1) \\ \hat{u}(k+2) \end{bmatrix} + \begin{bmatrix} \hat{y}_{free}(k+1) \\ \hat{y}_{free}(k+2) \\ \hat{y}_{free}(k+3) \end{bmatrix} \quad (28)$$

dengan matriks  $\mathbf{H}$  dan  $\hat{y}_{free}$  pada persamaan dibawah ini,

$$\mathbf{H} = \begin{bmatrix} a_1 & 0 & 0 \\ b_1a_1 + a_2 & a_1 & 0 \\ b_1^2a_1 + b_2a_1 + b_1a_2 & b_1a_1 + a_2 & a_1 \end{bmatrix} \quad (29)$$

$$\hat{y}_{free} = \begin{bmatrix} b_1 & b_2 & a_2 \\ b_1^2 + b_2 & b_1b_2 & b_1a_2 \\ b_1^3 + 2b_1b_2 & b_1^2b_2 + b_2^2 & b_1^2b_2 + b_2^2 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ u(k-1) \end{bmatrix} \quad (30)$$

Kemudian bisa didapat persamaan sinyal kontrol GPC,

$$u(k) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [H^T H + \lambda_u I]^{-1} H^T (y_r - \hat{y}_{free}) \quad (31)$$

dengan

$$y_r = \begin{bmatrix} y_{r+1} \\ y_{r+2} \\ y_{r+3} \end{bmatrix} \quad (32)$$

### D. Desain Model Referensi

Sinyal referensi masa depan diprediksi berdasarkan model referensi yang telah ditentukan. Model referensi merupakan model yang digunakan sebagai tujuan kontrol. Model ini digunakan agar respons sistem dapat sesuai dengan keinginan. Respon dari model referensi yang diharapkan dapat dicapai oleh sistem berbentuk persamaan orde satu.

$$G(s) = \frac{1}{0.2s+1} \quad (33)$$

Kemudian dari persamaan diatas dapat dibentuk pemodelan diskrit dengan *time sampling* sebesar 0.1 detik

$$G(z) = \frac{0.3935}{z - 0.6065} \quad (34)$$

Sehingga bisa didapat,

$$y_r(k) = b_r y_r(k-1) + a_r x(k-1) \quad (35)$$

$$y_r(k) = 0.6065 y_r(k-1) + 0.3935 x(k-1) \quad (36)$$

Dari persamaan diatas dapat diturunkan persamaan  $y_r$  tiga langkah kedepan, sehingga didapatkan persamaan  $y_r(k+3)$ . Dalam hal ini nilai  $x(k)$  dianggap sama pada setiap waktu.

$$y_r(k+1) = b_r y_r(k) + a_r x(k) \quad (37)$$

$$y_r(k+2) = b_r^2 y_r(k) + (b_r a_r + b_r) x(k) \quad (38)$$

$$y_r(k+3) = b_r^3 y_r(k) + (b_r^2 a_r + b_r a_r + a_r) x(k) \quad (39)$$

#### E. Desain Model Prediktor

Untuk menentukan model prediktor, digunakan model matematika dari motor servo AC pada Persamaan (19). Kemudian dapat dibentuk pemodelan diskrit orde dua dengan *time sampling* sebesar 0.1 detik.

$$\frac{Y(z)}{U(z)} = \frac{0.0007119z + 0.0007034}{z^2 - 1.965z + 0.9646} \quad (40)$$

Sehingga bisa didapat,

$$y_b(k) = b_{b1} y(k-1) + b_{b2} y(k-2) + a_{b1} x(k-1) + a_{b2} x(k-2) \quad (41)$$

$$y_b(k) = 1.965 y(k-1) - 0.9645 y(k-2) + 0.0007119 x(k-1) + 0.0007034 x(k-2) \quad (42)$$

Dari persamaan diatas maka dapat diturunkan nilai  $y_b(k)$  tiga langkah kedepan. sehingga didapatkan nilai  $y_b(k+3)$ . Sama dengan model referensi nilai  $x(k)$  dianggap sama pada setiap waktu.

$$y_b(k+1) = b_{b1} y(k) + b_{b2} y(k-1) + (a_{b1} + a_{b2}) x(k) \quad (43)$$

$$y_b(k+2) = (b_{b1}^2 + b_{b2}) y(k) + b_{b1} b_{b2} y(k-1) + (b_{b1} + 1)(a_{b1} + a_{b2}) x(k) \quad (44)$$

$$y_b(k+3) = (b_{b1}^3 + 2b_{b1} b_{b2}) y(k) + (b_{b1}^2 + b_{b1} b_{b2}^2) y(k-1) + (b_{b1}^2 + b_{b1} + b_{b2} + 1)(a_{b1} + a_{b2}) x(k) \quad (45)$$

Dari persamaan  $y_b(k+3)$  dan  $y_r(k+3)$  yang dihubungkan dengan model prediktor dan model referensi, maka didapatkan  $e(k+3)$  yang menjadi sinyal *error* masa depan untuk referensi kontroler GPC. Dengan nilai *error* tersebut kontroler GPC akan bekerja menentukan sinyal kontrol yang harus dihasilkan.

#### F. Desain Kontroler CG-GPC

Sinyal *error* asli dari tiap sumbu dibandingkan nilainya untuk mendapatkan nilai *contour error*. Kemudian kontroler CCC mengolah sinyal *contour error* agar respon *contour error* yang diharapkan bisa dicapai. Hasil respon kemudian akan ditambahkan ke *output* dari model referensi untuk kemudian dicari selisihnya dengan *output* dari model

prediktor. Diagram blok kontroler CC-GPC sepertidapat dilihat pada Gambar 3. Persamaan *contour error* dapat dituliskan seperti pada persamaan dibawah ini.

$$\varepsilon = -E_x C_x + E_y C_y \quad (46)$$

dimana

$$C_x = \sin \theta \quad (47)$$

$$C_y = \cos \theta \quad (48)$$

Kemudian untuk *Cross-Coupled Control* ( $C_c$ ) ditentukan menggunakan kontroler Proporsional + Integral untuk memperbaiki kesalahan kontur dan mendapatkan respon transien kesalahan kontur yang diinginkan. Dari *manual tuning* yang dilakukan, kemudian didapat parameter kontroler PI sebagai berikut,

$$C_c = 0.1 \left( 1 + \frac{1}{0.25s} \right) \quad (49)$$

### IV. PENGUJIAN DAN ANALISA SISTEM

Hasil dari pengujian berupa plot kurva referensi, kurva aktual dan kurva kesalahan. Kurva kesalahan yang dibentuk pada saat pengujian ini akan dihitung nilai RMSE (*Root Mean Square Error*) untuk mendapat nilai rata-rata *error*-nya.

#### A. Pemodelan Matematis Motor Servo AC

Pengujian dengan *input* sinyal *step* pertama dilakukan tanpa memberi gangguan. Respon dari masing-masing metode kontrol dan sinyal *error*-nya dapat dilihat pada Gambar 4. Sistem *closed-loop* memiliki respon dengan nilai *time constant* ( $\tau$ ) = 4 detik, *rise time* 5%-95% ( $t_r$ ) = 4.9 detik, *settling time*  $\pm 2\%$  ( $t_s$ ) = 21.8 detik, dan *max overshoot* ( $\%M_p$ ) = 18%. Kontroler M-CCC memiliki respon dengan nilai *time constant* ( $\tau$ ) = 0.2 detik, *rise time* 5%-95% ( $t_r$ ) = 0.56 detik, *settling time* ( $t_s$ ) = 1.2 detik, dan tidak mengalami *overshoot*. Kontroler CC-GPC memiliki respon dengan nilai *time constant* ( $\tau$ ) = 1.5 detik, *rise time* 5%-95% ( $t_r$ ) = 1.6 detik, *settling time*  $\pm 2\%$  ( $t_s$ ) = 19.7 detik, dan *max overshoot* ( $\%M_p$ ) = 50.3%. Kontroler M-CCC memiliki respon paling cepat dibanding kontroler lainnya karena kontroler PD pada yang digunakan pada tiap sumbunya mampu mempercepat respon sistem dan menghilangkan *overshoot*. RMSE sistem *closed-loop* sebesar 0.1708, kontroler M-CCC memiliki RMSE sebesar 0.0738, dan kontroler CC-GPC memiliki RMSE sebesar 0.1492. Pengujian dengan *input* sinyal *step* kemudian dilakukan dengan memberi variasi gangguan. Dari Tabel 1, dapat terlihat bahwa kontroler M-CCC memiliki nilai RMSE paling kecil ketika tidak diberi gangguan. Tetapi ketika diberi variasi gangguan, kenaikan RMSE-nya lebih besar dibanding kontroler CC-GPC. Hal ini disebabkan kontroler M-CCC memiliki respon yang sangat cepat sehingga rentan terhadap gangguan dapat dilihat pada Gambar 5. Kontroler CC-GPC walaupun memiliki respon yang pelan, tetapi cenderung lebih stabil terhadap gangguan. Perubahan nilai RMSE yang cukup kecil ketika gangguan ditambah menunjukkan kemampuan kontroler CC-GPC dalam mempertahankan kestabilannya terhadap variasi gangguan.



### B. Pengujian Sinyal Ramp

Pengujian dengan *input* sinyal *ramp* pertama dilakukan tanpa memberi gangguan. Respon dari masing-masing metode kontrol dan sinyal *error*-nya dapat dilihat pada Gambar 6. Sistem *closed-loop* memiliki respon dengan nilai *error steady state* ( $e_{ss}$ ) = 2.45. Kontroler M-CCC memiliki respon dengan nilai *error steady state* ( $e_{ss}$ ) = 0.2. Kontroler CC-GPC memiliki respon dengan nilai *error steady state* ( $e_{ss}$ ) = 0.49. RMSE sistem *closed-loop* sebesar 2.5224, kontroler M-CCC memiliki RMSE sebesar 0.1994, dan kontroler CC-GPC memiliki RMSE sebesar 0.5227.

Pengujian dengan *input* sinyal *ramp* kemudian dilakukan dengan memberi variasi gangguan. Dari Tabel 2, dapat dilihat bahwa kontroler CC-GPC memiliki respon *tracking* yang cukup baik dan kenaikan RMSE-nya ketika diberi gangguan cukup kecil. Sedangkan kontroler M-CCC kembali menjadi tidak stabil ketika diberi gangguan. Kontroler CC-GPC dapat melakukan *tracking* dengan sangat baik karena dapat mengikuti sinyal referensi tanpa nilai *error steady state* yang besar dapat dilihat pada Gambar 7. Selain itu juga perubahan nilai RMSE yang kecil menunjukkan kontroler CC-GPC mampu mempertahankan kestabilannya saat melakukan *tracking* dan diberi variasi gangguan.

### C. Pengujian Sinyal Step Variasi Set Point

Pengujian dengan *input* sinyal *step* variasi *set point* pertama dilakukan tanpa memberi gangguan. Respon dari masing-masing metode kontrol dan sinyal *error*-nya dapat dilihat pada Gambar 8. Sistem *closed-loop* memiliki respon yang lama dan tidak mencapai nilai *set point*. Kontroler M-CCC memiliki respon yang cepat dan mampu mencapai *set point*. Kontroler CC-GPC memiliki respon yang lama tetapi masih bisa mencapai *set point*. RMSE sistem *closed-loop* sebesar 0.4118, kontroler M-CCC memiliki RMSE sebesar 0.1393, dan kontroler CC-GPC memiliki RMSE sebesar 0.2925 dapat dilihat pada Gambar 9.

Pengujian dengan *input* sinyal *step* variasi *set point* kemudian dilakukan dengan memberi variasi gangguan. Dari Tabel 3, dapat dilihat bahwa kontroler CC-GPC memiliki respon *tracking* yang cukup baik dan kenaikan RMSE-nya ketika diberi gangguan cukup kecil. Sedangkan kontroler M-CCC kembali menjadi tidak stabil ketika diberi gangguan. Kontroler CC-GPC dapat melakukan *tracking* dengan sangat baik karena dapat mengikuti sinyal referensi tanpa nilai *error steady state* yang besar. Selain itu juga perubahan nilai RMSE yang kecil menunjukkan kontroler CC-GPC mampu mempertahankan kestabilannya saat melakukan *tracking* dan diberi variasi gangguan.

### D. Pengujian Pembentukan Kontur Linier

Pengujian pembentukan kontur linier pertama dilakukan tanpa memberi gangguan. Respon dari masing-masing metode kontrol dan sinyal *error*-nya dapat dilihat pada Gambar 10. RMSE sistem *closed-loop* sebesar 7.1699e-6, kontroler M-CCC memiliki RMSE sebesar 0.21709e-6, dan kontroler CC-GPC memiliki RMSE sebesar 5.5237e-06 dapat dilihat Gambar 11.

Pengujian pembentukan kontur linier kedua dilakukan dengan memberi variasi gangguan. Dari Tabel 4, dapat terlihat bahwa dalam melakukan pembentukan kontur linier, ketiga sistem memiliki RMSE *contour error* mendekati nol

ketika tidak ada gangguan. Hal ini disebabkan *contour error* merupakan resultan *error* dari kedua sumbu gerak. Akan tetapi saat gangguan diberikan, kontroler M-CCC mengalami kenaikan RMSE yang cukup besar. Kontroler CC-GPC mampu mempertahankan kestabilan dari gangguan dan membentuk kontur linier yang lebih baik dari sistem dengan kontroler lain.

## V. KESIMPULAN

Berdasarkan hasil simulasi dan pengujian, dapat diperoleh beberapa kesimpulan. Kontroler CC-GPC dengan prediksi tiga langkah dapat meningkatkan performa sistem motor servo AC. Ketika dilakukan pengujian sinyal *step*, *rise time* ( $t_r$ ) menjadi lebih cepat dari 4.9 detik menjadi 1.6 detik. Ketika dilakukan pengujian sinyal *ramp*, *error steady state* ( $e_{ss}$ ) berkurang dari 2.45 menjadi 0.49. Walaupun kontroler M-CCC ketika tanpa gangguan memiliki respon lebih cepat, kontroler CC-GPC lebih baik dalam meningkatkan ketahanan terhadap gangguan dibanding kontroler M-CCC. Rata-rata RMSE kontroler CC-GPC sebesar 0.3391 sedangkan rata-rata RMSE kontroler M-CCC sebesar 0.7727 hasilnya dapat dilihat pada Gambar 12 sampai Gambar 17.

Berdasarkan hasil penelitian dan pengujian, dapat diperoleh beberapa saran untuk penelitian selanjutnya. Melakukan perancangan kontroler GPC dengan memilih *prediction horizon* dan *control horizon* yang lebih tepat agar bisa mendapat respon sistem yang lebih baik. Pada perancangan kontroler CC-GPC, direkomendasikan untuk menggunakan kontroler PD untuk kontrol pada masing-masing sumbu untuk mempercepat respon sistem dan menggunakan kontroler GPC pada kontroler *cross-coupled* untuk meningkatkan ketahanan terhadap gangguan dan mengurangi *contour error*.

## DAFTAR PUSTAKA

- [1] Y. Koren, "Cross-Coupled Biaxial Computer Control for Manufacturing Systems," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 102, no. 4, pp. 265–272, Dec. 1980, doi: 10.1115/1.3149612.
- [2] Y. Koren and C. C. Lo, "Variable-Gain Cross-Coupling Controller for Contouring," *CIRP Ann. - Manuf. Technol.*, vol. 40, no. 1, pp. 371–374, 1991, doi: 10.1016/S0007-8506(07)62009-5.
- [3] Y. T. Shih, C. S. Chen, and A. C. Lee, "A Novel Cross-Coupling Control Design for Bi-Axis Motion," *Int. J. Mach. Tools Manuf.*, vol. 42, no. 14, pp. 1539–1548, Nov. 2002, doi: 10.1016/S0890-6955(02)00109-8.
- [4] J. Sun and C. Hu, "Research on Modeling of Contour Error for Motion Control System of CNC Machine," in *2011 2nd International Conference on Mechanic Automation and Control Engineering, MACE 2011 - Proceedings*, 2011, pp. 1553–1556, doi: 10.1109/MACE.2011.5987246.
- [5] Y. Xiao and K. Zhu, "Cross-Coupling Generalized Predictive Control for Motion Systems," *Proc. 7th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2002*, pp. 1664–1669, 2002, doi: 10.1109/icarcv.2002.1235025.
- [6] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized Predictive Control-Part I. The basic Algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, Mar. 1987, doi: 10.1016/0005-1098(87)90087-2.
- [7] D. W. Clarke and C. Mohtadi, "Properties of Generalized Predictive Control," *Automatica*, vol. 25, no. 6, pp. 859–875, Nov. 1989, doi: 10.1016/0005-1098(89)90053-8.
- [8] M. Vijayakarthick and P. K. Bhaba, "Position Tracking Performance of AC Servomotor based on New Modified Repetitive Control Strategy," *IJJRAS*, vol. 10, no. 1, pp. 119–128, 2014.