

Perancangan Sistem Kendali NCTF Berbasis Arduino Mega untuk Sistem Putar Eksentris Satu Massa Horizontal

Perwita Kurniawan^{1, a *}, Purtojo^{2, b}, Herianto^{3, c} dan Gesang Nugroho^{4, d}

¹ Program Studi S2 Ilmu Teknik Mesin, Fakultas Teknik, Universitas Gadjah Mada, Jl. Grafika No 2 Yogyakarta 55281, Indonesia.

² Program Studi S3 Ilmu Teknik Mesin, Fakultas Teknik, Universitas Gadjah Mada, Jl. Grafika No 2 Yogyakarta 55281, Indonesia.

^{3,4} Jurusan Teknik Mesin dan Industri, Fakultas Teknik, Universitas Gadjah Mada, Jl. Grafika No 2 Yogyakarta 55281, Indonesia

^aperwita_k@yahoo.co.id, ^bpurtojo@uii.ac.id, ^cherianto@ugm.ac.id, ^dgesangnugroho@ugm.ac.id

Abstrak

Sistem kendali *nominal characteristic trajectory following* (NCTF) merupakan sebuah sistem kendali yang praktis. Sistem kendali ini terdiri dari sebuah *nominal characteristic trajectory* (NCT) dan sebuah kompensator. Aspek praktis dari sebuah sistem kendali NCTF diperoleh dari pembuatan pengendali yang hanya didasarkan pada eksperimen *open-loop* sederhana. Arduino Mega digunakan sebagai pengendali sistem. Penelitian dilakukan untuk mengetahui performa sistem kendali NCTF berbasis Arduino Mega yang diaplikasikan pada sistem putar eksentris satu massa horizontal. Perangkat lunak MATLAB[®] digunakan untuk memberikan perintah ke Arduino Mega agar menggerakkan motor dan membaca *encoder*. Untuk dapat melihat performa sistem, maka dilakukan pengujian dengan menggunakan variasi pembebanan dan variasi posisi sudut tujuan, dengan *sampling time* 0.01detik. Variasi pembebanan yang diberikan diantaranya 0, 0.066, 0.156 dan 0.266Kg. Sedangkan variasi sudut tujuan diantaranya 90, 180, 270 dan 360 derajat. Informasi yang berupa nilai *steady state error*, *settling time*, dan *overshoot*, dianalisa untuk dapat menentukan performa sistem yang telah dibuat. Hasil penelitian menunjukkan bahwa sistem kendali NCTF berbasis Arduino Mega menunjukkan performa pengendalian posisi yang baik. Keakuratan penunjukan posisi dapat dilihat dari kecilnya rata-rata nilai *steady state error* yang terjadi sebesar 0.18%. Kecepatan respon yang baik dapat ditunjukkan dengan kecilnya rata-rata nilai *settling time* sebesar 0.968 detik. Masih munculnya rata-rata *overshoot* sebesar 1.78% merupakan sebuah reaksi normal dari adanya momen inersia akibat pembebanan.

Kata kunci : NCTF, Pengendalian posisi, Sistem putar eksentris, *Point to point*

Pendahuluan

Sistem pengendalian posisi dari titik ke titik menuntut keakuratan yang tinggi dan respon yang cepat. Secara umum, pengendalian posisi dilakukan dengan menggunakan sistem kendali *Proportional Integral Derivative* (PID). Namun penggunaan sistem kendali PID memerlukan sebuah model yang dibangun secara matematis. Model matematis pada dasarnya cenderung rumit dan memerlukan pengetahuan yang mendalam tentang teori sistem kendali dan rancang bangun. Banyak variabel yang harus diketahui seperti nilai inersia dan beberapa variabel motor. Dalam

aplikasi praktis, seorang desainer mekanik yang tidak ahli dalam sistem kendali terkadang perlu untuk merancang sebuah sistem kendali. Selain itu, pembuatan model dan parameter identifikasi umumnya merepotkan dan membutuhkan waktu yang lebih banyak. Oleh karena itu, kemudahan desain sistem kendali dan kesederhanaan struktur pengendali merupakan hal yang sangat penting dalam aplikasi praktis.

Untuk mengatasi masalah ini, sistem kendali *nominal characteristic trajectory following* (NCTF) telah dikembangkan sebagai pengontrol praktis untuk sistem

kendali posisi [1]. Sistem kendali NCTF memiliki kinerja posisi dan ketahanan yang baik [2]. Sistem kendali NCTF juga efektif untuk mengimbangi efek gesekan yang merupakan sumber utama ketidaktepatan posisi [3]. Apabila dibandingkan dengan sistem kendali PID, sistem kendali NCTF memiliki struktur yang lebih sederhana dan *robustness* yang lebih baik [4]. Sistem kendali NCTF terdiri dari dua bagian utama, yaitu *Nominal Characteristic Trajectory* (NCT) dan sebuah kompensator. Kompensator yang paling sederhana untuk digunakan adalah kompensator *Proportional-Integral* (PI).

Sistem kendali NCTF merupakan sebuah pengendali praktis yang memiliki keunggulan dalam hal desainnya yang praktis. Sistem kendali NCTF mirip dengan sistem kendali yang berbasis model, akan tetapi dalam proses desainnya sistem kendali NCTF tidak memerlukan model yang sempurna beserta identifikasi parameter obyek yang dikendalikan secara pasti. Dengan kemudahannya inilah maka muncul banyak peluang bagi seorang desainer mekanik untuk menciptakan sebuah mekanisme mekanik terlebih dahulu, dan memikirkan sistem pengendalinya dengan lebih mudah setelah mekanisme mekaniknya dibuat. Hal ini juga membuka peluang lain bagi seorang desainer mekanik untuk melakukan proses *reverse engineering* dari peralatan yang sudah ada.

Kepraktisan penggunaan sistem kendali NCTF dapat ditingkatkan dengan dukungan perangkat keras dan perangkat lunak yang digunakan. Untuk meningkatkan kemudahan dalam aplikasinya, maka sistem kendali NCTF ini dibangun dengan menggunakan pengendali yang sederhana yaitu Arduino Mega. Penggunaan Arduino Mega membuat sistem menjadi lebih ringkas karena program pengendalian bisa dibangun dengan menggunakan perangkat Matlab Simulink yang berbasis diagram blok. Sumber pembelajaran yang terbuka dan komunitas Arduino yang banyak membuat penggunaan Arduino Mega menjadi lebih praktis.

Penelitian tentang sistem kendali NCTF sudah banyak dilaporkan. Beberapa penelitian diantaranya diaplikasikan pada sistem gerak linear maupun gerak berputar, baik secara

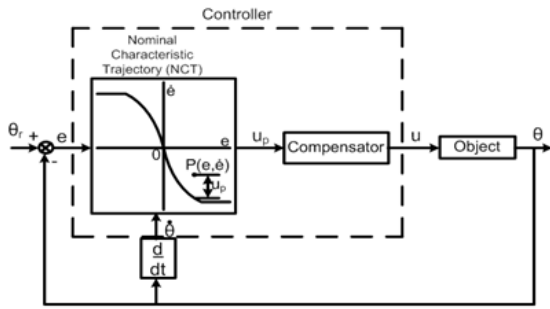
Point to Point (PTP) [5,6,7,8] maupun secara *Continous Path* (CP) [9]. Berbagai cara dan metode untuk menemukan langkah yang paling optimal pun telah dilakukan oleh beberapa peneliti, diantaranya untuk menemukan kompensator yang berbasis PI. Namun sampai saat ini, keseluruhan penelitian sistem kendali NCTF dengan kompensator berbasis PI untuk mekanisme PTP dan CP tersebut hanya diaplikasikan untuk kendali posisi putar dengan massa yang seimbang [10]. Sistem kendali NCTF untuk mengendalikan posisi putar dengan massa tidak seimbang belum pernah dilaporkan.

Tujuan dari penelitian ini adalah untuk melihat performa sistem kendali posisi putar horisontal dengan massa yang tidak seimbang. Respon sistem terhadap perintah sudut tujuan juga akan diamati, terkait penggunaan Arduino Mega sebagai pengendali gerakannya.

Sistem Kendali NCTF

Sistem kendali NCTF merupakan sebuah sistem kendali yang terdiri dari dua buah bagian yaitu NCT dan kompensator. NCT adalah sebuah karakteristik dasar obyek, dan kompensator adalah sebuah mekanisme yang memaksa obyek untuk mengikuti NCT. Pada sistem kendali NCTF, masukan pengendali adalah *error*, keluaran kendali adalah sinyal u yang digunakan untuk menggerakkan obyek, dan gerak obyek.

Cara kerja sistem kendali NCTF pada prinsipnya adalah dengan membandingkan masukan berupa gerak obyek dengan *error-rate* yang disajikan oleh NCT, pada nilai *error* tertentu. Perbedaan nominal antara *error-rate* aktual dari obyek dengan yang ada di NCT adalah sinyal u_p . Sinyal u_p merupakan keluaran dari NCT. Oleh karena itu apabila gerak obyek mengikuti NCT dengan sempurna, maka nilai dari sinyal u_p adalah nol, sehingga tidak ada aksi yang dilakukan oleh kompensator. Apabila sinyal u_p tidak sama dengan nol, maka kompensator berperan agar nilai dari sinyal u_p mencapai nol [11].



Gambar 1. Sistem Kendali NCTF

Sistem kendali NCTF adalah sebuah sistem kendali yang dibuat berdasarkan eksperimen *open-loop* sederhana. Secara garis besar pembuatan sistem kendali NCTF meliputi tiga tahapan, yaitu:

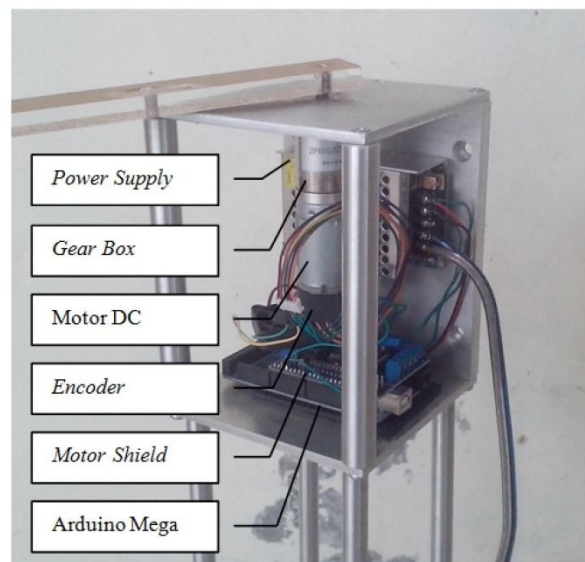
1. Menggerakkan obyek dengan masukan *stepwise* dan mengukur perpindahan serta kecepatan obyek.
2. Membuat NCT dari respon obyek yang diperoleh dari langkah pertama.
3. Mendesain kompensator berdasar informasi dari NCT.

Set-up Eksperimen

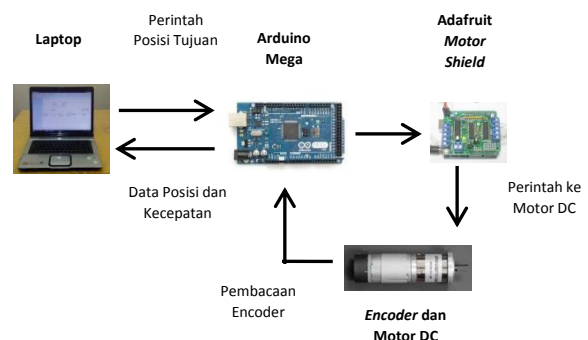
Pada bagian ini akan dijelaskan peralatan yang digunakan untuk pengujian, mekanisme pengaturan perintah dan pengambilan data. Peralatan yang digunakan terdiri dari satu buah motor DC yang terangkai bersama transmisi dan *encoder*-nya, satu buah *power supply* berarus 3 Ampere, satu buah Arduino Mega dan satu buah *Adafruit motor shield*. Semua peralatan diatas dirangkai menjadi satu pada sebuah rangka alumunium. Untuk melakukan pengujian pembebanan, maka disiapkan tiga buah beban dengan massa 0.066kg, 0.156kg, dan 0.266kg. Rangkaian alat uji secara keseluruhan dapat dilihat pada Gambar 2.

Set-up eksperimen dapat dilihat pada Gambar 3. Perintah pergerakan posisi motor untuk pengujian dilakukan pada sebuah laptop dengan menggunakan perangkat lunak Matlab. Dalam Matlab, program dibangun dengan menggunakan blok Simulink. Perintah posisi ini akan disampaikan oleh laptop melalui perangkat lunak Matlab menuju ke Arduino Mega. Untuk dapat mengkomunikasikan Matlab dengan Arduino Mega, maka harus dilakukan penginstalasian

program koneksi pada perangkat lunak Matlab dan perangkat lunak Arduino IDE. Perintah posisi yang telah diterima oleh Arduino Mega diteruskan menuju *Adafruit motor shield*, untuk dieksekusi pada motor DC. Hasil pergerakan motor DC akan dibaca oleh *encoder* sebagai data perubahan posisi dan perubahan kecepatan, dan dikembalikan menuju laptop melalui Arduino Mega. Data ini akan diolah dengan menggunakan perangkat lunak Microsoft Excel, dengan terlebih dahulu mengambilnya dari lembar kerja Matlab.



Gambar 2. Alat uji



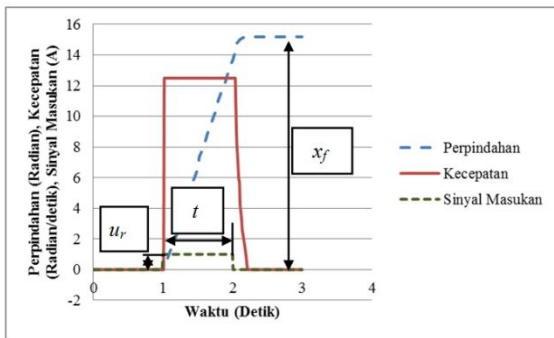
Gambar 3. *Set-up* eksperimen

Desain NCT

Untuk membuat NCT, maka sebuah eksperimen sederhana dilakukan. Dalam eksperimen sederhana ini, motor DC tanpa beban lengan digerakkan dengan sebuah sinyal masukan berupa *step*. Perubahan posisi dan perubahan kecepatan diukur pada saat

yang bersamaan. Grafik pada Gambar 4 menunjukkan sinyal masukan, kecepatan dan perpindahan.

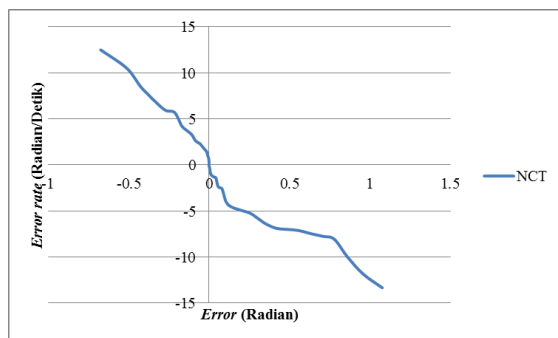
Dari yang diperlihatkan pada Gambar 4, kita dapat memperoleh beberapa parameter yang dibutuhkan untuk membuat NCT dan kompensator diantaranya adalah x_f yang merupakan perpindahan maksimal, u_r yang merupakan tinggi sinyal input, dan t_r yang merupakan waktu sinyal input. Proses eksperimen sederhana ini dilakukan pada dua nilai sinyal masukan yang berbeda, agar didapatkan respon pada arah yang berbeda.



Gambar 4. Respon *open-loop* untuk membuat NCT

$$K = \frac{x_f}{u_r \cdot t_r} \dots\dots\dots (1)$$

Data hasil pemutaran motor DC secara *open-loop* ini diambil dari sejak sinyal masukan dihilangkan. Nilai perubahan kecepatan yang terjadi sejak sinyal masukan dihilangkan digabungkan dengan nilai perubahan perpindahan sejak sinyal masukan dihilangkan. Gabungan kedua data ini merupakan karakteristik dari sistem yang disebut dengan NCT, seperti yang diperlihatkan pada Gambar 5.



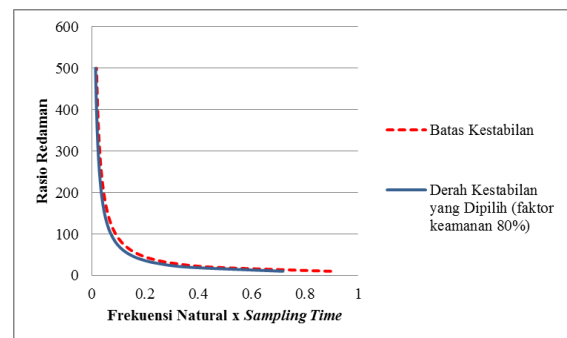
Gambar 5. NCT

Desain Kompensator

Pada bagian ini akan disampaikan sebuah metode sederhana untuk dapat menentukan besarnya kompensator, yang terdiri dari nilai konstanta proporsional dan konstanta integral. Daerah batas kestabilan dapat ditentukan dengan cara menggerakkan mekanisme NCT yang sudah didapatkan sebelumnya dengan menggunakan konstanta proporsional saja. Nilai konstanta proporsional diperoleh dengan cara menggerakkan sistem dengan konstanta proporsional saja secara bertahap dari nilai konstanta proporsional kecil kemudian diulang seterusnya, hingga diperoleh respon gerak osilasi yang tak berhingga. Nilai konstanta proporsional yang diperoleh dengan metode ini dinamakan dengan K_{pu} atau K_p ultimate (bernilai 0.39 untuk mekanisme lengan eksentris horisontal). Nilai K_{pu} dapat dimasukkan dalam Pers.2.

$$\zeta = K_{pu} \frac{\alpha K}{2\omega_n} \dots\dots\dots (2)$$

Pers. 2 memiliki dua buah variabel yang belum diketahui, yaitu rasio redaman (ζ) dan frekuensi natural (ω_n). Kedua variabel ini akan menjadi variabel dalam grafik. Sumbu X akan diwakili oleh frekuensi natural dikalikan dengan *sampling time* (ditentukan oleh peneliti sebesar 0.01detik), sedangkan sumbu Y akan diwakili oleh rasio redaman. Grafik yang terbuat dari nilai K_p ultimate ini dinamakan grafik batas kestabilan, seperti yang disajikan pada Gambar 6.



Gambar 6. Grafik kestabilan sistem

Grafik batas kestabilan berfungsi untuk membatasi daerah stabil dengan daerah yang

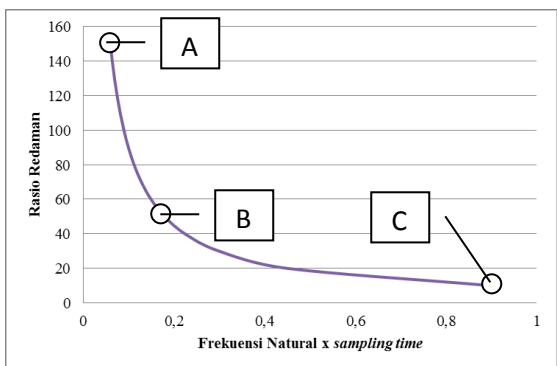
tidak stabil. Agar sistem dapat dipastikan bekerja dengan stabil, maka nilai rasio redaman dan frekuensi natural harus diambil dengan menggunakan angka keamanan (ditentukan oleh peneliti sebesar 80%).

Pada gambar menunjukkan daerah kestabilan yang dipilih. Dengan menggunakan nilai perkalian antara rasio redaman dan frekuensi natural, maka dapat dilakukan perhitungan untuk menentukan Konstanta Proporsional dengan Pers. 3 dan Konstanta Integral dengan Pers. 4.

$$K_p = \frac{2\zeta\omega_n}{\alpha K} \dots\dots\dots (3)$$

$$K_i = \frac{\omega_n^2}{\alpha K} \dots\dots\dots (4)$$

Daerah kestabilan yang dipilih masih memberikan variasi rasio redaman dan frekuensi natural. Hal ini akan mengakibatkan munculnya berbagai variasi konstanta integral yang disebabkan karena konstanta integral hanya dihitung dari satu variabel saja yaitu frekuensi natural. Untuk itu peneliti melakukan pendekatan dengan menentukan tiga buah variasi rasio redaman di dalam daerah kestabilan yang sama, yaitu sebesar 10, 50 dan 150, seperti yang diperlihatkan pada Gambar 7.



Gambar 7. Variasi Kompensator

Perhitungan untuk mencari K_p dan K_i dilakukan dengan menggunakan Pers.3 dan Pers.4. Perhitungan ini menggunakan kombinasi variabel pada kondisi A, B dan C. Hasil perhitungan dapat dilihat pada Tabel 1. Semakin besar nilai frekuensi natural maka mengakibatkan semakin besarnya nilai K_i .

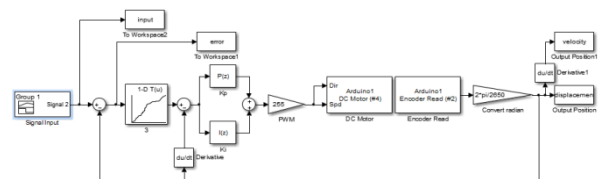
Tabel 1. Karakteristik Kompensator

| Keterangan | Kompensator | | |
|------------------------|-------------|--------|--------|
| | A | B | C |
| Rasio redaman | 150 | 50 | 10 |
| Frekuensi natural [Hz] | 5.984 | 17.951 | 89.754 |
| K_p | 0.312 | 0.312 | 0.312 |
| K_i | 0.008 | 0.070 | 1.750 |

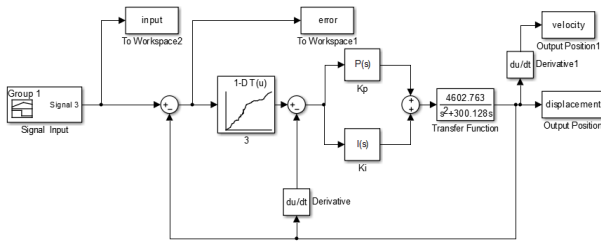
Variasi hasil perhitungan konstanta proporsional dan konstanta integral pada ketiga titik kemudian diuji pada alat eksperimen, kemudian diambil respon yang terbaik sesudah dibandingkan. Kombinasi konstanta proporsional dan konstanta integral yang terbaik (kompensator B), ditentukan sebagai konstanta proporsional dan konstanta integral sistem.

Evaluasi Performa Sistem

Dalam bagian ini akan disampaikan hasil pengujian sistem yang sudah diperoleh. Pengujian dilakukan dengan memasukkan konstanta proporsional dan konstanta integral pada blok K_p dan K_i di program blok Simulink Matlab. Pengujian secara umum dilakukan dalam dua metode, yaitu metode eksperimen dan metode simulasi. Metode eksperimen dilakukan dengan cara mengeksekusi gerakan pada alat uji secara langsung. Sedangkan metode simulasi dilakukan dengan mengeksekusi program pada program Matlab saja (program hanya melakukan perhitungan matematis gerakan) dengan menggunakan *transfer function*-nya saja. Gambar 8 menunjukkan program blok Simulink Matlab untuk pengujian secara eksperimen dan Gambar 9 menunjukkan program blok Simulink Matlab untuk pengujian secara simulasi.

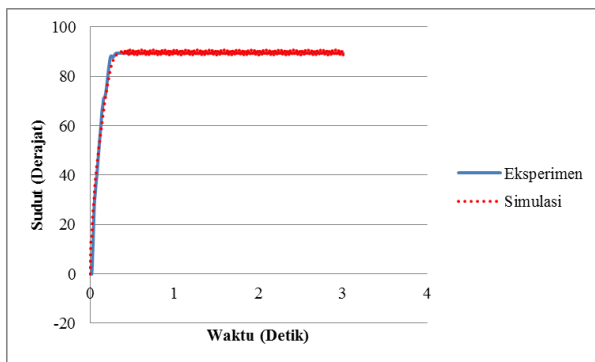


Gambar 8. Program Blok Simulink untuk Pengujian Eksperimen

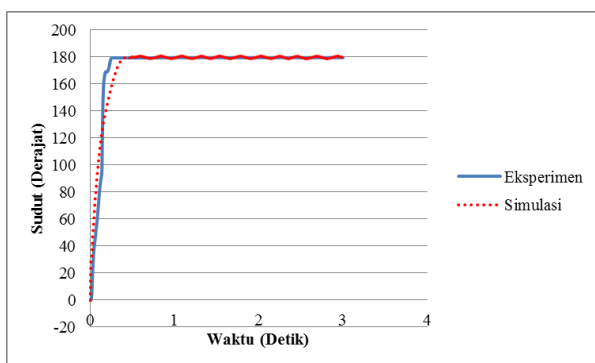


Gambar 9. Program Blok Simulink untuk Pengujian Simulasi

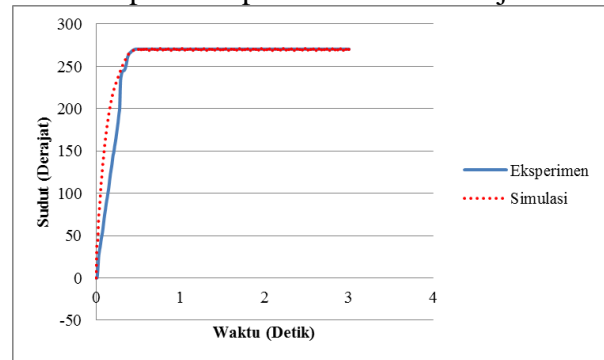
Program blok yang sudah dibuat digunakan untuk melakukan pengujian secara eksperimen dan simulasi. Hasil pengujian dibandingkan untuk dapat dianalisa, seperti yang tampak pada Gambar 10, Gambar 11, Gambar 12 dan Gambar 13. Secara umum, respon pengujian simulasi tampak lebih cepat, namun pada kondisi *steady* tampak bahwa hasil simulasi sedikit beresilasi. Grafik hasil uji eksperimen tidak menunjukkan hasil yang jauh berbeda dengan grafik hasil uji simulasi. Seandainya dilihat pada data uji Tabel 2, tampak bahwa data hasil uji secara eksperimen lebih baik daripada data hasil uji secara simulasi. Hal ini menunjukkan bahwa respon sistem masih cukup baik.



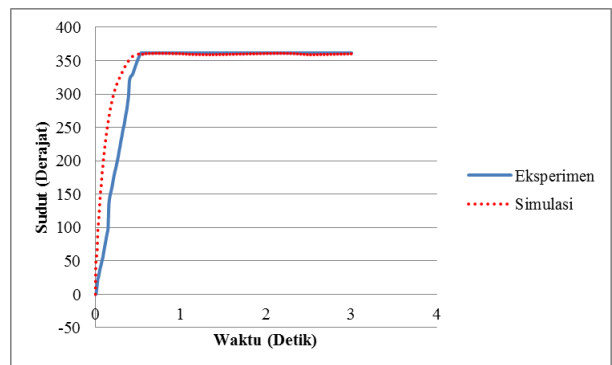
Gambar 10. Perbandingan hasil simulasi dan eksperimen pada sudut 90 derajat



Gambar 11. Perbandingan hasil simulasi dan eksperimen pada sudut 180 derajat



Gambar 12. Perbandingan hasil simulasi dan eksperimen pada sudut 270 derajat



Gambar 13. Perbandingan hasil simulasi dan eksperimen pada sudut 360 derajat

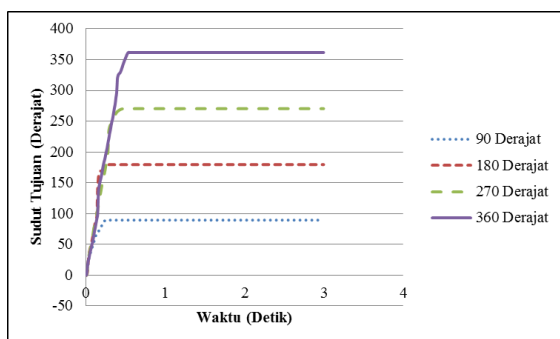
Tabel 2. Perbandingan hasil simulasi dan eksperimen

| Sinyal Masukan [Derajat] | Metode Pengujian | Overshoot [%] | Settling Time [Detik] | Steady State Error [%] |
|--------------------------|------------------|---------------|-----------------------|------------------------|
| 90 | Simulasi | 0.82 | 0.31 | 1.35 |
| | Eksperimen | 0.00 | 0.31 | 0.63 |
| 180 | Simulasi | 0.47 | 0.34 | 0.66 |
| | Eksperimen | 0.00 | 0.27 | 0.25 |
| 270 | Simulasi | 0.38 | 0.37 | 0.38 |
| | Eksperimen | 0.05 | 0.49 | 0.13 |
| 360 | Simulasi | 0.33 | 0.41 | 0.33 |
| | Eksperimen | 0.08 | 0.57 | 0.43 |

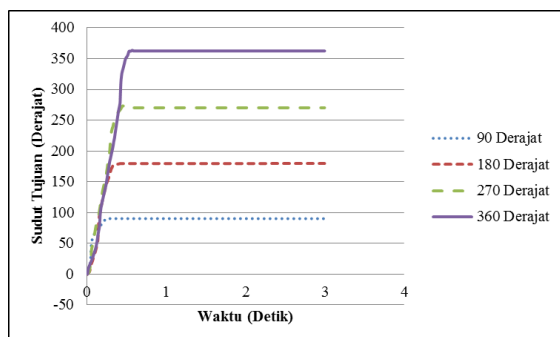
Sebuah sistem kendali dapat dikatakan baik apabila memiliki respon yang hampir sama dengan simulasi, ketika dilakukan berbagai

pembebanan (*robust*). *Robustness* sistem dapat dilihat dengan melakukan pengujian secara eksperimen dengan menggunakan berbagai variasi perintah sudut tujuan dan variasi massa pembebanan.

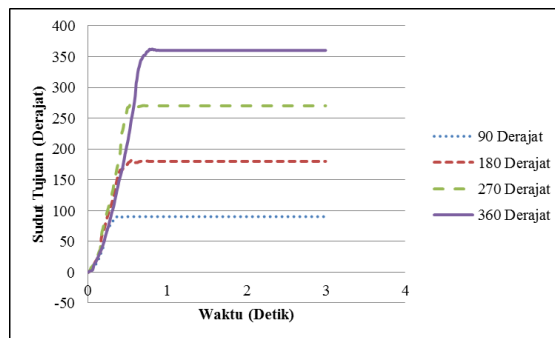
Pengujian secara eksperimen yang dilakukan pada pembebanan yang sama menghasilkan respon sistem terhadap variasi sudut yang relatif sama. Hasil pengujian menunjukkan perbedaan respon ketika beban yang berbeda diberikan. Semakin besar beban diberikan maka *overshoot* akan semakin tinggi, dan efeknya juga membuat *settling time* juga menjadi semakin lama. Pada Gambar 14 dan Gambar 15, *overshoot* belum tampak muncul pada grafik, namun pada Gambar 16 sudah mulai muncul adanya *overshoot* dan semakin tampak membesar pada Gambar 17. Hal ini juga berlaku sama pada *settling time*.



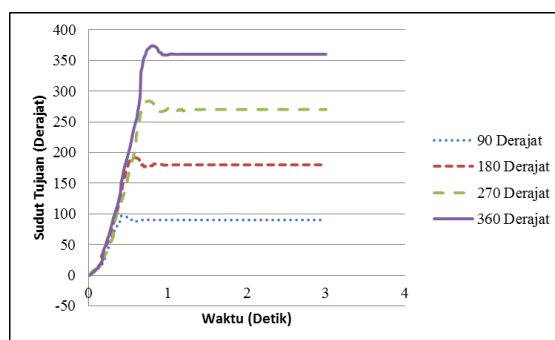
Gambar 14. Respon sistem secara eksperimen tanpa beban



Gambar 15. Respon sistem secara eksperimen dengan beban uji 0.066kg



Gambar 16. Respon sistem secara eksperimen dengan beban uji 0.156kg



Gambar 17. Respon sistem secara eksperimen dengan beban uji 0.266kg

Data hasil pengujian secara eksperimen yang tersaji pada Gambar 14, Gambar 15, Gambar 16, dan Gambar 17, terangkum pada Tabel 3. Pada Tabel 3 tampak bahwa nilai *steady state error* cukup bervariasi namun masih dibawah 1%. Hal ini menunjukkan bahwa sistem yang telah dibuat mampu *robust* terhadap berbagai variasi sudut tujuan dan pembebanan.

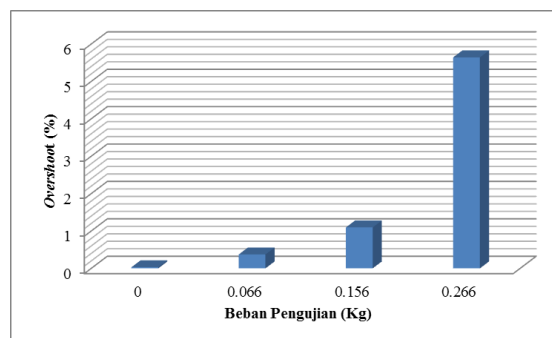
Tabel 3. Data hasil uji secara eksperimen

| Massa Uji [Kg] | Sinyal Masukan [Derajat] | Overshoot [%] | Settling Time [Detik] | Steady state error [%] |
|----------------|--------------------------|---------------|-----------------------|------------------------|
| 0 | 90 | 0.00 | 0.63 | 0.63 |
| | 180 | 0.00 | 0.86 | 0.25 |
| | 270 | 0.05 | 0.49 | 0.13 |
| | 360 | 0.08 | 0.57 | 0.43 |
| 0.066 | 90 | 0.15 | 2.16 | 0.13 |
| | 180 | 0.00 | 2.3 | 0.10 |
| | 270 | 1.11 | 0.5 | 0.08 |

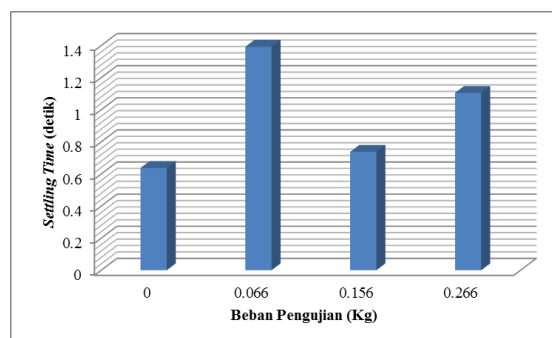
| | | | | |
|-------|-----|------|------|------|
| | 360 | 0.23 | 0.61 | 0.69 |
| 0.156 | 90 | 1.51 | 0.47 | 0.13 |
| | 180 | 1.06 | 0.83 | 0.02 |
| | 270 | 1.21 | 0.74 | 0.08 |
| | 360 | 0.60 | 0.91 | 0.05 |
| 0.266 | 90 | 7.25 | 0.79 | 0.02 |
| | 180 | 6.42 | 1.02 | 0.02 |
| | 270 | 5.08 | 1.43 | 0.03 |
| | 360 | 3.81 | 1.18 | 0.05 |

Kesimpulan

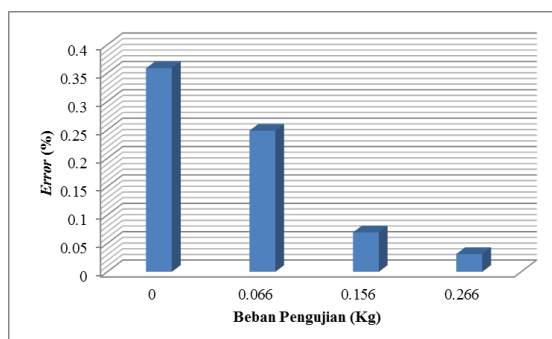
Sistem kendali NCTF berbasis Arduino Mega yang diaplikasikan pada sistem gerak eksentris satu massa horisontal, mampu menunjukkan performa yang cukup baik. Hal ini tampak dari Gambar 18, Gambar 19 dan Gambar 20. Masih munculnya rata-rata *overshoot* sebesar 1.78% merupakan sebuah reaksi normal dari adanya momen inersia akibat pembebanan. Nilai *overshoot* ini dapat terlihat secara rinci pada Gambar 18. Kecepatan respon yang baik dapat dilihat dari kecilnya rata-rata nilai *settling time* sebesar 0.968 detik. Kecepatan respon secara rinci dapat dilihat pada Gambar 19. Keakuratan penunjukan posisi dapat dilihat dari kecilnya rata-rata nilai *steady state error* yang terjadi sebesar 0.18%. Rincian besarnya nilai *steady state error* pada masing masing pengujian variasi beban dapat dilihat pada Gambar 20. Pengujian secara eksperimen memberikan hasil yang nyata bahwa sistem mampu *robust* untuk mengatasi faktor luar pengganggu seperti beban, gesekan, inersia, dan gravitasi untuk mekanisme putar eksentris satu massa horisontal.



Gambar 18. Nilai *overshoot* sistem



Gambar 19. Nilai *settling time* sistem



Gambar 20. Nilai *steady state error* sistem

Referensi

- [1] Wahyudi. (2002). New practical control of PTP positioning systems. Ph.D. Thesis. Tokyo Institute of Technology.
- [2] Wahyudi, Sato K. & Shimokohbe A. Characteristics of Practical Control for Point-to-point Positioning Systems: Effect of design parameters and actuator saturation, Precision Engineering, Vol. 27, No. 2, 2003, pp. 157-169.
- [3] Wahyudi, Sato K. & Shimokohbe A. Robustness Evaluation of Three Friction Compensation Methods for Point-to-point (PTP) Positioning Systems, Robotics and

Autonomous System, Elsevier, Vol. 52, Issues 2-3, 2005, pp. 247-256.

[4] Chong & Sato. (2008). Practical Control of Non-Friction Mechanism for Precision Positioning. International Conference on Control, Automation, and System. Seoul, Korea.

[5] Purtojo, Akmeliawati R., & Wahyudi. (2008). Fuzzy-based NCTF Controller for PTP Positioning: Fuzzy Membership and Rule Based Modifications, 3rd International Conference on Mechatronics (ICOM'08). Kuala Lumpur, Malaysia.

[6] Purtojo, Akmeliawati R., & Wahyudi. (2009). Intelligent Fuzzy-based NCTF Controller for Point-to-point Positioning System: Reduced Controller Structure, The First International Seminar on Science and Technology (ISSTEC 2009). Yogyakarta, Indonesia.

[7] Purtojo, Wahyudi, Akmeliawati, R. & Shafie, A.A. (2009). Fuzzy-based NCTF

Control of PTP Positioning System, Experimental Results. 4th IEEE Conference on Industrial Electronics and Applications ICIEA 2009. 1741 – 1747.

[8] Purtojo. (2010). Fuzzy-based NCTF control of point-to-point positioning systems. Master Dissertation. International Islamic University Malaysia.

[9] Sato K. & Maeda. A practical control method for precision motion—Improvement of NCTF control method for CM control. Precision Engineering, Vol. 33, 2009, pp. 173-186.

[10] Fitri & Rini. (2011). Study Review on NCTF Controller for Point-To-Point Positioning System. 11th International Conference on Control, Automation, and System. Gyeonggi-do, Korea.

[11] Maeda, G.J., & Sato, K. (2008). Practical control method for ultra-precision positioning using a ballscrew mechanism. Precision Engineering, 32(4), 309-318.