

Kendali Kecepatan Motor DC Berbasis WNCS Menggunakan Pengendali PI *Anti-Windup*

Baisrum, Adnan Rafi Al Tahtawi, Sarjono Wahyu Jadmiko

Jurusan Teknik Elektro, Politeknik Negeri Bandung
Jl. Gegerkalong Hilir, Ds. Ciwaruga, Bandung 40012, Indonesia
baisrum@polban.ac.id

Abstrak

Pemanfaatan jaringan komunikasi dalam lingkaran tertutup sistem kendali dapat memberikan beberapa keuntungan seperti dalam hal fleksibilitas. Akan tetapi, adanya beberapa parameter jaringan dapat menurunkan performansi pengendalian. Penelitian ini bertujuan untuk merancang pengendalian kecepatan motor DC berbasis *Wireless Networked Control System* (WNCS) menggunakan modul nirkabel nRF24L01. *Controller node* dan *plant node* dirancang agar dapat berkomunikasi satu sama lain secara nirkabel. Pada *controller node* terdapat mikrokontroler, modul nirkabel, dan penampil antarmuka, sedangkan pada *plant node* terdapat mikrokontroler, modul nirkabel, *driver* motor, dan motor DC. Pengendali Proporsional Integral *Anti-Windup* (PI-AW) yang dapat mempertimbangkan saturasi sinyal kendali digunakan untuk memperoleh respon pengendalian yang terbaik. Hasil pengujian eksperimen perangkat keras menunjukkan bahwa pengendali yang dirancang mampu menghasilkan respon dengan *settling time* 0,7 detik dan *maximum overshoot* 1,17%. Sinyal kendali yang dihasilkan juga tidak melebihi batas maksimum yaitu 12 V.

Kata kunci: kendali kecepatan, WNCS, nRF24L01, pengendali PI-AW

Abstract

The utilization of wireless networks in a control system can provide several advantages such as flexibility. However, the presence of several network parameters can reduce control performance. This study aims to design a DC motor speed control based on Wireless Networked Control System (WNCS) using nRF24L01 wireless module. Controller node and plant node are designed to communicate each other wirelessly. The wireless network based on nRF24L01 is placed between controller and plant nodes. Controller node consists of microcontroller, wireless module, and display interface, while the plant node consists of microcontroller, wireless module, motor driver, and DC motor itself. Proportional Integral Anti-Windup (PI-AW) controller which can consider the saturation of the control signal is used to obtain the control performances. The hardware experimental results show that the designed controller is able to produce responses with settling time 0.7 s and overshoot 1.17%. The control signal generated also does not exceed the maximum limit 12 V.

Keywords: speed control, WNCS, nRF24L01, PI-AW controller

I. PENDAHULUAN

Sistem kendali berbasis jaringan atau dikenal dengan istilah *Networked Control System* (NCS) merupakan konsep pengendalian sistem dimana terdapat parameter jaringan di dalam lingkaran tertutupnya [1]. Dalam konsep NCS, antara pengendali dengan *plant* ditempatkan terpisah dan mungkin dalam jarak yang cukup jauh. Hal ini tentu saja menjadi kelebihan tersendiri dalam perancangan sistem kendali seperti dalam hal fleksibilitas. Namun demikian, terdapatnya jaringan

dalam lingkaran tertutup sistem kendali dapat menimbulkan beberapa gangguan seperti adanya waktu tunda dan *packet dropout* dalam pengiriman data. Dengan demikian diperlukan metode kendali yang handal dan mampu mengatasi adanya beberapa permasalahan tersebut.

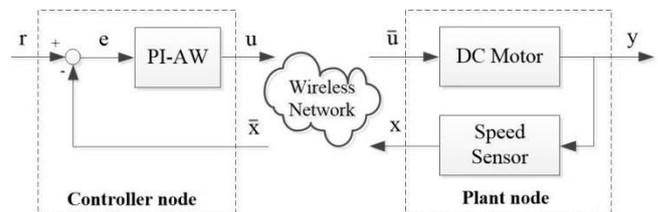
Berbagai metode sistem kendali dapat dirancang dalam perancangan kendali berbasis NCS. Pengendali PID konvensional telah dikembangkan dengan konsep NCS [2]. Penelitian ini memaparkan proses perancangan pengendali PID hingga beberapa metodologi dalam melakukan optimasi

konstanta PID dalam konsep NCS. Kemudian penelitian serupa dilakukan kembali dengan fokus kepada pemanfaatan jaringan nirkabel pada pengendali PID [3]. Selain memaparkan proses perancangan pengendalian, penelitian ini juga memberikan gambaran mengenai perancangan estimator untuk mengatasi gangguan dan ketidakpastian dalam konsep NCS. Sistem kendali tingkat lanjut pun dikembangkan dalam konsep NCS, seperti konsep *Dissipative* NCS dengan umpan balik keluaran dan umpan balik keadaan [4], [5]. Metodologi ini dirancang melalui pendekatan *Markovian Jump System* (MJS) yang mampu mengantisipasi masalah stokastik yang muncul dalam parameter jaringan. Konsep ini pun telah diimplementasikan untuk mengendalikan kecepatan motor DC melalui jaringan komunikasi [6]. Kemudian efek adanya jaringan nirkabel dalam pengendali PI dan *Fuzzy Logic Controller* (FLC) tipe 2 juga telah diteliti [7]. Pada penelitian ini, performa pengendali PI dan FLC tipe 2 dibandingkan dengan tiga skenario permasalahan, yaitu perbedaan periode sampling, *overload* pada jaringan, dan *packet dropout*. Terakhir, baru-baru ini muncul konsep pengendali *Predictive* PI (PPI) pada aplikasi NCS menggunakan WirelessHART untuk kebutuhan industri. Konsep pengendali PPI diusulkan untuk mengatasi permasalahan stokastik waktu tunda dalam jaringan WirelessHART. Metode ini juga telah dianalisis kekokohan dan kestabilannya dengan memperhitungkan ketidakpastian parameter jaringan dan sifat stokastik waktu tunda [8].

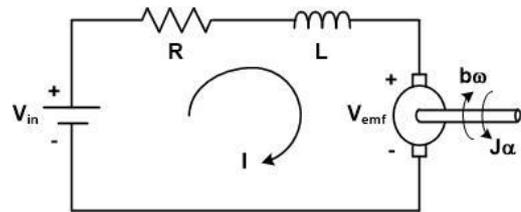
Berbeda dengan penelitian sebelumnya, penelitian ini bertujuan untuk merancang pengendalian kecepatan motor DC berbasis WNCS menggunakan modul nRF24L01. Pengendali Proporsional Integral *Anti-Windup* (PI-AW) dirancang sebagai pengendali kecepatan yang mampu memperhitungkan adanya saturasi [9]. Penelitian ini dilakukan melalui simulasi dan eksperimen perangkat keras. Simulasi dilakukan dengan mengintegrasikan parameter waktu tunda minimum dan maksimum dari modul nRF24L01. Modul ini digunakan karena memiliki beberapa kelebihan jika dibandingkan modul nirkabel lainnya seperti ZigBee dan ESP8266 [10]-[12]. Kemudian hasil simulasi divalidasi dengan melakukan percobaan dalam perangkat keras.

II. METODE PENELITIAN

Perancangan sistem kendali dilakukan menggunakan perangkat MATLAB/Simulink untuk mengendalikan kecepatan motor DC dengan mengintegrasikan parameter waktu tunda.



Gambar 1. Perancangan sistem kendali



Gambar 2. Rangkaian motor DC

Gambaran umum sistem pengendalian yang dirancang dapat dilihat pada Gambar 1.

Dari perancangan tersebut dapat dilihat bahwa terdapat dua bagian diagram blok, yaitu bagian pengendali (*controller node*) dan bagian kendalian (*plant node*). Kedua bagian ini terpisah dan dihubungkan dengan menggunakan jaringan nirkabel. Pada bagian pengendali terdapat r sebagai nilai referensi, e adalah nilai kesalahan, u adalah sinyal kendali yang dikirim ke kendalian, dan \bar{x} adalah data kecepatan setelah melewati jaringan nirkabel. Pada bagian kendalian terdapat \bar{u} sebagai sinyal kendali yang dikirim menuju *plant* setelah melewati jaringan nirkabel, y adalah keluaran, dan x adalah data kecepatan yang dikirim ke pengendali.

A. Model Motor DC

Ada beberapa cara yang dapat dilakukan untuk menentukan model dari suatu *plant*, diantaranya: pengukuran parameter secara langsung, identifikasi sistem, dan pendekatan respon orde satu. Secara umum rangkaian motor DC digambarkan pada Gambar 2. Berdasarkan rangkaian tersebut, secara matematis terdiri dari model mekanis dan elektrik.

Persamaan mekanis motor DC diuraikan sebagai berikut:

$$J\ddot{\theta} + b\dot{\theta} = KI$$

$$\ddot{\theta} = \frac{d\dot{\theta}}{dt} = -\frac{b}{J}\dot{\theta} + \frac{K}{J}I \quad (1)$$

dengan J adalah momen inersia (kg.m^2), $\dot{\theta}$ atau ω adalah kecepatan sudut (rad/s), b adalah koefisien gesekan (N.m.s), K adalah konstanta motor, $\ddot{\theta}$ atau α adalah percepatan sudut (rad/s^2), I adalah arus (A). Adapun persamaan elektrik motor DC dapat diuraikan sebagai berikut:

$$L \frac{dI}{dt} + RI + V_{emf} = V_{in} \quad (2)$$

Karena $V_{emf} = K\dot{\theta}$, maka:

$$\frac{dI}{dt} = -\frac{R}{L}I + \frac{V_{in}}{L} - \frac{K}{L}\dot{\theta} \quad (3)$$

dengan L adalah induktansi (H), V_{emf} adalah tegangan balik (V), R adalah resistansi (Ohm), dan V_{in} adalah tegangan masukan (V).

Jika dibentuk dalam model ruang keadaan dengan $x_1 = \dot{\theta}$, $x_2 = i$ dan $u = V_{in}$ maka model motor DC dapat memenuhi persamaan berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u \quad (4)$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (5)$$

Jika dibentuk dalam model fungsi alih, maka model motor DC dapat dituliskan sebagai berikut.

$$\frac{\Omega(s)}{V(s)} = \frac{K^2}{LJs^2 + (Lb + Rf)s + Rb + K^2} \quad (6)$$

Pada praktiknya, parameter motor DC terkadang sulit didapatkan khususnya parameter induktansi (L), momen inersia (J), dan koefisien gesekan (b). Oleh karena itu, pemodelan secara lengkap dapat didekati oleh persamaan dinamik orde satu yang hanya membutuhkan parameter konstanta motor (K) dan konstanta waktu (T).

B. Model Parameter Jaringan Nirkabel

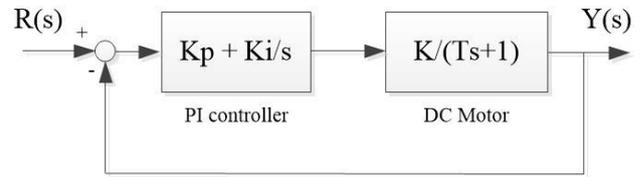
Seperti diketahui bahwa model motor DC dapat didekati dalam fungsi alih orde satu sebagai berikut.

$$\frac{\Omega(s)}{V(s)} = \frac{K}{Ts + 1} \quad (7)$$

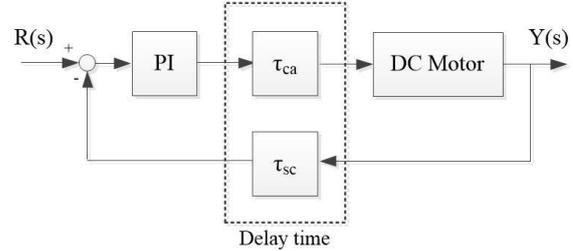
dengan Ω adalah kecepatan, V adalah tegangan, K adalah konstanta motor, dan T adalah konstanta waktu respon. Penelitian ini menggunakan pengendali PI untuk mengendalikan kecepatan motor DC dengan persamaan sebagai berikut.

$$U(s) = K_p \cdot E(s) + \frac{K_i}{s} \cdot E(s) \quad (8)$$

dengan U adalah sinyal kendali, E adalah nilai kesalahan, K_p adalah konstanta proporsional, dan K_i adalah konstanta integral.



Gambar 3. Motor DC dengan pengendali PI



Gambar 4. Integrasi parameter waktu tunda

Dengan demikian persamaan lingkaran tertutup dari motor DC dengan pengendali PI seperti pada Gambar 3 memenuhi persamaan berikut.

$$\frac{\Omega(s)}{R(s)} = \frac{K(K_p s + K_i)}{Ts^2 + (1 + KK_p)s + KK_i} \quad (9)$$

Dengan mengintegrasikan parameter jaringan waktu tunda seperti pada Gambar 4 dari sensor ke pengendali (τ_{sc}) dan dari pengendali ke aktuator (τ_{ca}) serta mengabaikan variabel waktu tunda pengendali (τ_c), maka diperoleh persamaan lingkaran tertutup sebagai berikut.

$$\frac{\Omega(s)}{R(s)} = \frac{(KK_p s + K_i)\tau_{ca}}{Ts^2 + (1 + KK_p\tau_{ca}\tau_{sc})s + KK_i\tau_{ca}\tau_{sc}} \quad (10)$$

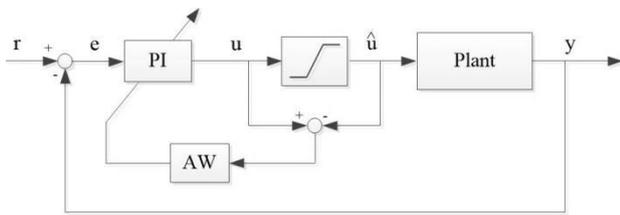
dengan τ_{ca} dan τ_{sc} merupakan fungsi waktu tunda yang dapat dibentuk menggunakan pendekatan fungsi Pade orde satu [13] sebagai berikut.

$$\tau_{ca} = \tau_{sc} = e^{-\tau s} = \frac{2 - \tau s}{2 + \tau s} \quad (11)$$

dengan τ adalah besarnya waktu tunda.

C. Pengendali PI-AW

Perancangan pengendali dengan *Anti-Windup* (AW) diperlukan saat sinyal kendali yang dibutuhkan mengalami saturasi. Kondisi ini tidak dapat dihindari karena pada praktiknya sumber tegangan untuk pengendalian motor DC memiliki batas maksimum. Ketika kondisi tersebut terjadi, maka respon pengendalian dapat terdegradasi. Dengan demikian diperlukan mekanisme AW untuk mengatasi permasalahan tersebut seperti ditunjukkan pada Gambar 5.



Gambar 5. Diagram blok pengendali PI-AW

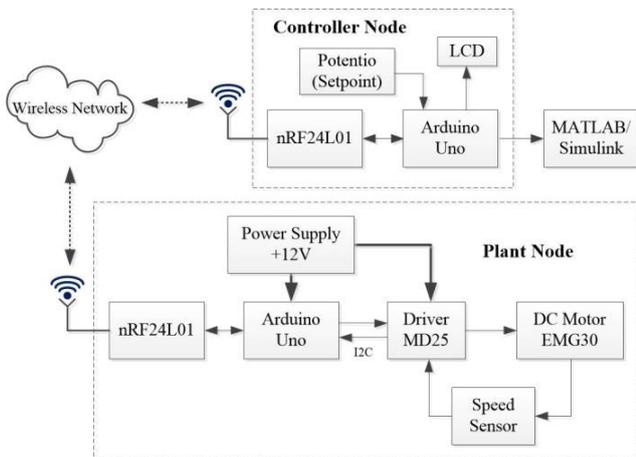
AW bekerja dengan cara menghitung selisih antara sinyal kendali yang dihasilkan dengan sinyal kendali maksimum. Jika terdapat selisih, maka nilai tersebut dikalkulasi dengan konstanta waktu integral dan diumpam balikkan ke bagian integrator. Dengan mekanisme ini, maka pengendali integral dapat menyesuaikan nilai pengendaliannya berdasarkan umpan balik tersebut. Adapun besarnya sinyal kendali yang dihasilkan saat terjadi mekanisme AW adalah sebagai berikut [14].

$$U(s) = Kp \cdot E(s) + \frac{Ki}{s} \cdot E(s) - \frac{1}{Ti s} [U(s) - \bar{U}(s)] \tag{12}$$

dengan T_i adalah waktu integral dan \bar{U} adalah sinyal kendali maksimum. Nilai $1/T_i$ biasa disebut nilai konstanta *back calculation* (Kb). Besarnya nilai T_i dapat diperoleh dari perbandingan nilai Kp/Ki . Parameter AW akan aktif saat nilai U lebih besar dari nilai \bar{U} . Jika nilai U tidak melebihi nilai \bar{U} , maka $U = \bar{U}$ dan mekanisme AW tidak akan aktif atau sama dengan nol.

D. Perangkat Keras

Perangkat keras dirancang ke dalam dua bagian, yaitu bagian pengendali (*controller node*) dan bagian kendalian (*plant node*) seperti pada Gambar 6. Pada bagian *controller node* terdapat potensiometer, modul Arduino Uno, modul nirkabel nRF24L01, dan LCD.



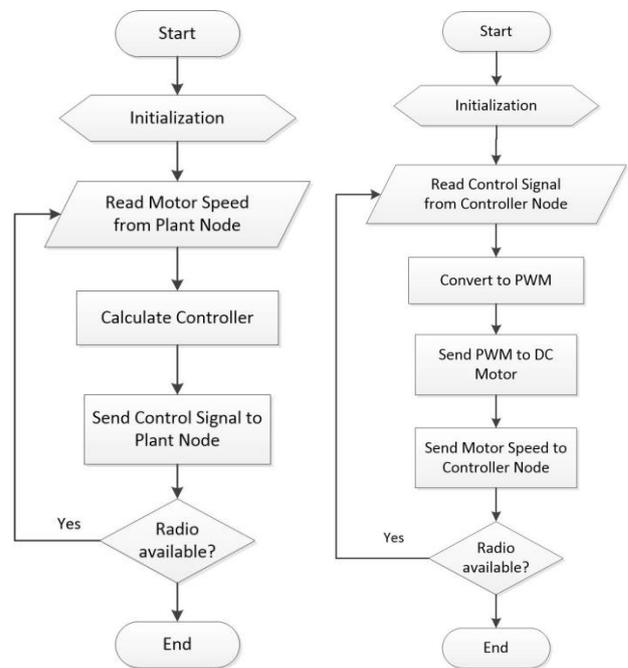
Gambar 6. Perancangan perangkat keras

Potensiometer digunakan untuk memberikan nilai referensi (*setpoint*) kecepatan motor DC. Modul mikrokontroler Arduino Uno berfungsi sebagai perangkat pengendalian dimana pengendali PI-AW ditanamkan. Modul nirkabel nRF24L01 digunakan untuk mengirimkan sinyal kendali yang dikalkulasi oleh mikrokontroler ke *plant node*. LCD digunakan untuk menampilkan parameter pengendalian seperti *setpoint*, *error*, sinyal kendali, dan *output*. Pada bagian *controller node*, perangkat antarmuka MATLAB/Simulink dihubungkan untuk menampilkan parameter pengendalian kecepatan motor DC dalam bentuk grafik.

Pada bagian *plant node* terdapat modul nirkabel nRF24L01, modul Arduino Uno, *driver* MD25, motor DC EMG30, sensor kecepatan, dan catu daya. Modul nirkabel yang digunakan sama seperti pada bagian pengendali yang berfungsi untuk menerima sinyal kendali dan mengirimkan data sensor. Modul mikrokontroler Arduino Uno digunakan sebagai perangkat untuk menggerakkan motor DC melalui modul *driver* MD25. Pada bagian ini terdapat catu daya sebesar 12 V untuk menggerakkan motor DC.

E. Perangkat Lunak

Pada bagian perangkat lunak, program pengendalian motor DC dirancang menggunakan bahasa pemrograman C pada aplikasi Arduino IDE. Adapun diagram alir program perangkat lunak untuk *controller node* dan *plant node* dapat dilihat pada Gambar 7.



Gambar 7. Diagram alir pemrograman pada *controller node* (kiri) dan pada *plant node* (kanan)

Terdapat dua program yang dirancang yaitu program untuk *controller node* dan program untuk *plant node*. Secara umum, program pada kedua *node* ini ditanamkan dengan tujuan untuk mengirimkan dan menerima data secara *wireless* dari sensor ke pengendali maupun dari pengendali ke sensor. Perbedaannya terletak pada struktur program dimana pada *controller node* hanya terdapat program pengendalian saja, sedangkan pada *plant node* terdapat program baca dan kirim data sensor serta terima data sinyal kendali dan meneruskannya ke aktuator.

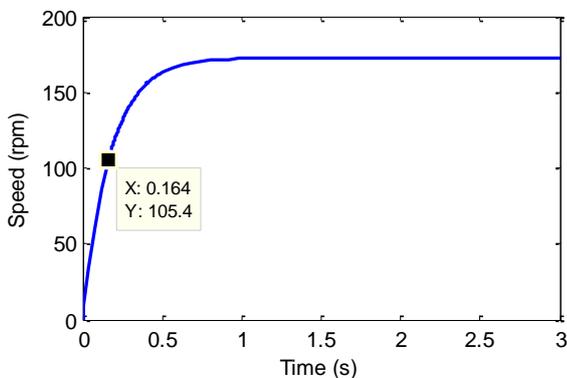
III. HASIL DAN PEMBAHASAN

A. Pengujian Respon Open Loop

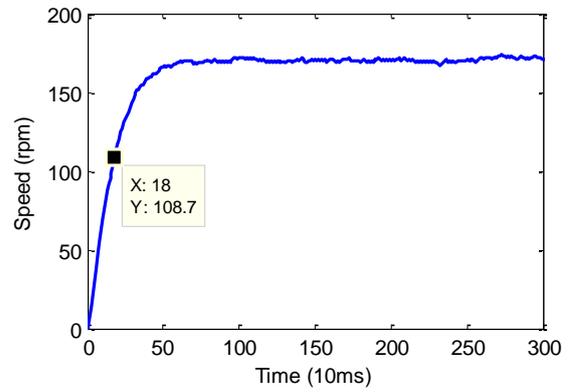
Pengujian respon *open loop* dilakukan untuk mengetahui respon kecepatan motor DC saat diberi tegangan masukan. Untuk kebutuhan simulasi, maka terlebih dahulu harus ditentukan model motor DC EMG30. Berdasarkan referensi [15], motor DC EMG30 memiliki parameter $K = 0,509 \text{ N.m/A}$; $L = 0,0034 \text{ H}$; $R = 7,101 \text{ }\Omega$; $J = 0,00567 \text{ Kg.m}^2$; dan $b = 0,000931$. Dengan menggunakan parameter tersebut, maka diperoleh fungsi alih berdasarkan persamaan (7) sebagai berikut.

$$\frac{\Omega(s)}{V(s)} = \frac{3,308}{0,04s + 0,23} \quad (13)$$

Untuk mengetahui respon *open loop* dari model tersebut, maka dilakukan simulasi menggunakan MATLAB/Simulink dengan hasil seperti ditunjukkan pada Gambar 8. Pada saat simulasi, model motor DC diberi sinyal masukan sebesar 12 V sesuai dengan spesifikasi tegangan motor DC EMG30. Dengan memberikan catu daya 12 V, hasil pengujian respon *open loop* secara eksperimen juga ditampilkan pada Gambar 9.



Gambar 8. Respon *open loop* motor DC simulasi



Gambar 9. Respon *open loop* motor DC eksperimen

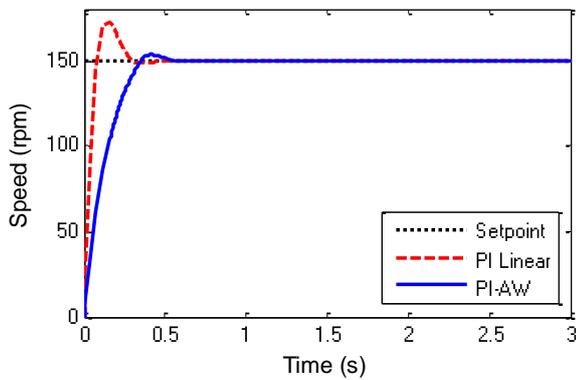
Berdasarkan hasil pengujian diperoleh bahwa model motor DC EMG30 dapat mencapai kecepatan 170 rpm saat diberi tegangan masukan 12 V dengan konstanta waktu sebesar 0,164 s pada simulasi dan 0,175 s pada eksperimen. Untuk nilai konstanta waktu diperoleh dari waktu respon saat *output* yang dihasilkan mencapai 63,2% dari nilai maksimum yaitu sebesar 105,91 rpm. Konstanta waktu yang dihasilkan tidak memiliki perbedaan secara signifikan. Dengan hasil ini, maka dapat dikatakan model motor DC yang digunakan dianggap mampu mewakili motor DC sebenarnya.

B. Pengujian Pengendali PI-AW

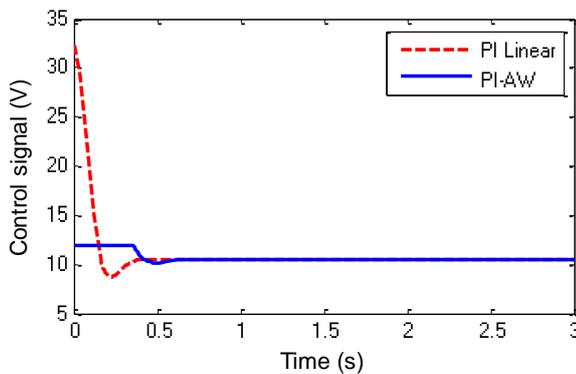
Pengujian ini bertujuan untuk mengetahui kinerja pengendalian PI-AW sebelum diintegrasikannya parameter waktu tunda pada sistem *closed-loop*. Pada bagian ini dilakukan pengujian pengendalian kecepatan motor DC membandingkan respon pengendali PI linier dengan PI-AW. Besarnya konstanta K_p dan K_i ditentukan menggunakan fasilitas *PID Tuner* pada MATLAB/Simulink. Berdasarkan hasil optimasi menggunakan *PID Tuner* diperoleh nilai $K_p = 0,213$ dan $K_i = 4,169$. Untuk nilai konstanta *back calculation* (K_b) diperoleh dari $1/T_i$ seperti pada persamaan (12) dengan besarnya T_i adalah K_p/K_i . Dengan demikian diperoleh nilai K_b sebesar 19,572. Hasil pengujian respon untuk ketiga skenario pengendali ditampilkan pada Gambar 10 dengan sinyal kendali pada Gambar 11. Adapun spesifikasi lengkap hasil pengendalian yang dihasilkan untuk pengujian pengendalian tanpa menggunakan parameter jaringan diperlihatkan pada Tabel 1.

Tabel 1. Spesifikasi pengendali PI linier dan PI-AW

| Jenis kendali | Rise Time | Settling Time | Over-shoot | Sinyal kendali maks. |
|---------------|-----------|---------------|------------|----------------------|
| PI Linier | 0,08 s | 0,50 s | 15,20 % | 32,3 V |
| PI-AW | 0,35 s | 0,56 s | 2,23 % | 12 V |



Gambar 10. Respon pengendali PI linier dan PI-AW



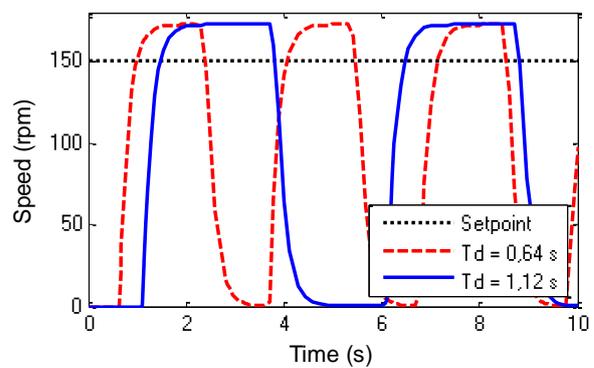
Gambar 11. Sinyal kendali PI linier dan PI-AW

Berdasarkan hasil pengujian secara simulasi diperoleh bahwa pengendali PI-AW memiliki kinerja terbaik jika dibandingkan dengan PI linier. Hal ini dapat dilihat dari respon kecepatan motor DC dengan PI-AW memiliki *overshoot* lebih kecil yaitu sebesar 2,23 % meskipun memiliki *rise time* 0,35 detik dan lebih lama jika dibandingkan dengan PI linier dengan *rise time* 0,08 detik. Selain itu, PI-AW juga lebih baik jika dilihat dari sinyal kendali yang dihasilkan. PI-AW menghasilkan sinyal kendali yang tidak melebihi batas maksimum 12 V, sedangkan PI linier menghasilkan sinyal kendali yang melebihi batas maksimum.

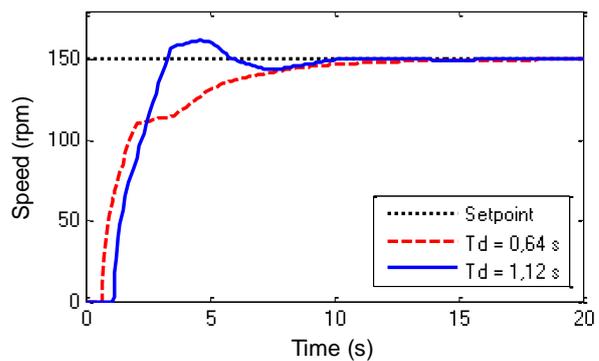
C. Pengujian Pengendali PI-AW Dengan Parameter Waktu Tunda

Selanjutnya, untuk mengetahui kinerja pengendalian dalam konsep WNCS, diantara pengendali dan *plant* ditempatkan parameter waktu tunda. Pengujian dilakukan dengan menggunakan pengendali PI-AW dan memberikan waktu tunda yang sama antara τ_{sc} dan τ_{ca} yaitu masing-masing sebesar 0,64 s dan 1,12 s. Nilai ini merupakan nilai waktu tunda minimum dan maksimum dari modul nRF24L01 berdasarkan penelitian [16]. Berdasarkan hasil simulasi, diperoleh respon kecepatan motor DC seperti pada Gambar 12. Dari hasil simulasi terlihat bahwa adanya waktu tunda, respon kecepatan menjadi beresilasi.

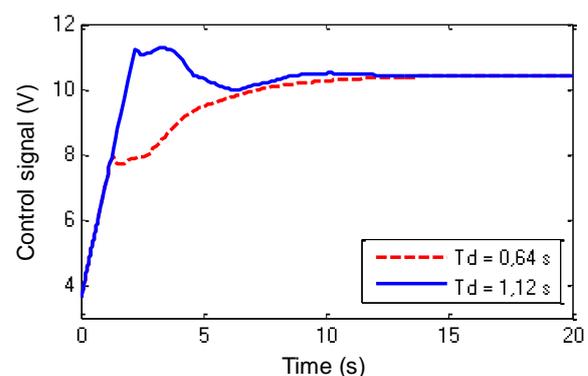
Kemudian untuk memperbaiki kembali respon pengendalian, maka diperlukan perancangan ulang pengendali PI-AW. Proses penalaan dilakukan kembali dengan asumsi waktu tunda yang paling besar yaitu 1,12 detik. Dengan kembali melakukan penalaan parameter K_p dan K_i , maka diperoleh respon pengendalian yang lebih baik seperti ditunjukkan pada Gambar 13 dan sinyal kendali pada Gambar 14. Adapun spesifikasi pengendalian yang diperoleh ditunjukkan pada Tabel 2. Besarnya konstanta pengendali K_p dan K_i yang diperoleh sama yaitu sebesar 0,023 dengan nilai $K_b = 1$.



Gambar 12. Respon pengendalian dengan integrasi parameter waktu tunda



Gambar 13. Respon PI-AW dengan $K_p = K_i = 0,023$



Gambar 14. Sinyal kendali PI-AW dengan $K_p = K_i = 0,023$

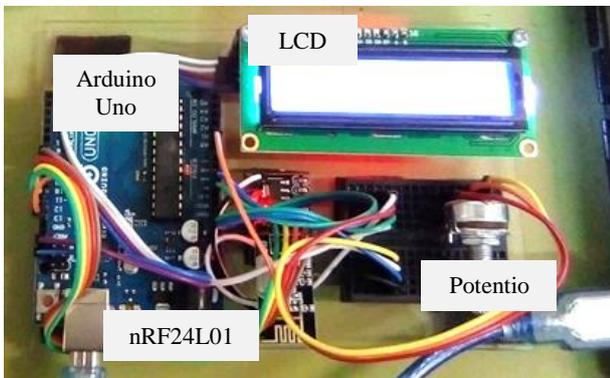
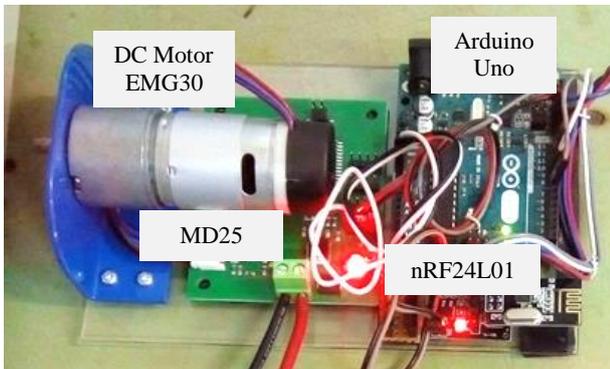
Tabel 2. Spesifikasi pengendali PI-AW dengan waktu tunda

| Waktu Tunda | Rise Time | Settling Time | Over-shoot | Sinyal kendali maks. |
|-------------|-----------|---------------|------------|----------------------|
| 0,64 s | 16,2 s | 16,2 s | 0 % | 10,4 V |
| 1,12 s | 3,25 s | 11,8 s | 7,8 % | 11,26 V |

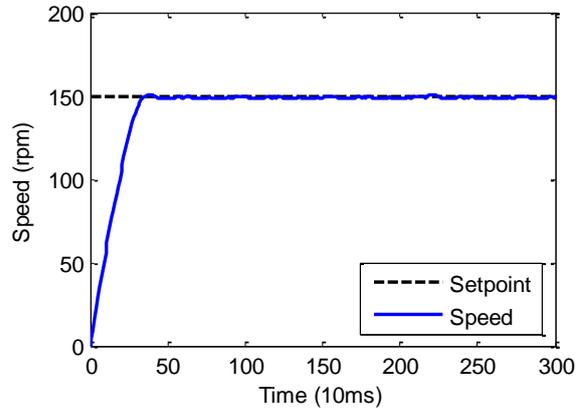
Berdasarkan hasil pengujian diperoleh bahwa pengendali PI-AW mampu memperbaiki respon kecepatan motor DC dengan sinyal kendali yang tidak melebihi batas maksimum meskipun menghasilkan *settling time* yang lebih lama. Hasil pengujian menggunakan waktu tunda yang lebih lama dapat kembali menurunkan respon kecepatan. Kondisi ini dapat diatasi dengan kembali melakukan penalaan parameter K_i ataupun merancang optimasi pengendali PID berdasarkan variasi waktu tunda.

D. Pengujian Perangkat Keras

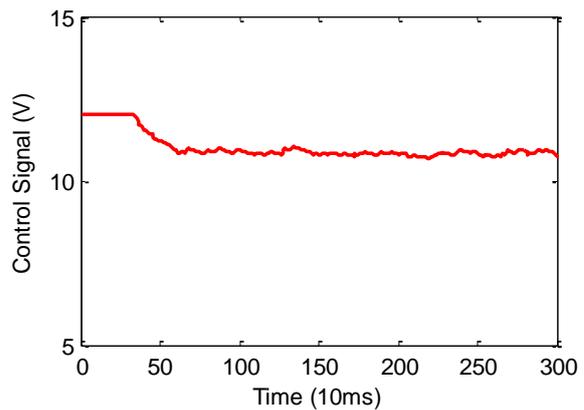
Sistem diimplementasikan dalam bentuk purwarupa perangkat keras sesuai dengan perancangan yang telah dilakukan. Hasil implementasi dari sistem dapat dilihat pada Gambar 15.



Gambar 15. Implementasi perangkat keras: *plant node* (atas) dan *controller node* (bawah)



Gambar 16. Respon kecepatan motor DC EMG30 dengan pengendali PI-AW



Gambar 17. Sinyal kendali dengan pengendali PI-AW

Pengujian dalam bentuk eksperimen perangkat keras dilakukan untuk memvalidasi hasil pengujian secara simulasi. Pengujian dilakukan dengan cara mengamati respon kecepatan motor DC EMG30 dan sinyal kendali yang dihasilkan setelah diintegrasikannya modul nRF24L01 sesuai dengan perancangan perangkat keras. Pengujian dilakukan pada *plant node* dengan cara menanamkan pengendali PI-AW ke mikrokontroler dalam bentuk bahasa pemrograman. Pengujian dilakukan dengan cara mengamati respon kecepatan menggunakan pengendali PI-AW dengan konstanta $K_p = 0,152$ dan $K_i = 0,477$. Konstanta tersebut diperoleh menggunakan metode *trial and error*. Hasil pengujian secara eksperimen dengan adanya parameter jaringan dapat dilihat pada Gambar 16 dengan sinyal kendali pada Gambar 17.

Berdasarkan hasil pengujian secara eksperimen diperoleh bahwa dengan nilai *setpoint* 150 rpm, pengendali PI-AW mampu menghasilkan respon kecepatan dengan *rise time* 0,35 s, *settling time* 0,7 s, dan *maximum overshoot* 1,17 %. Sinyal kendali yang dihasilkan juga tidak melebihi batas maksimum 12 V dengan nilai *steady state* sebesar 10,8 V. Jika dibandingkan dengan hasil simulasi,

respon pengendalian yang dihasilkan secara eksperimen dapat dikatakan lebih baik. Terdapatnya perbedaan kinerja pengendalian disebabkan oleh beberapa hal seperti perbedaan nilai konstanta PI dan adanya perbedaan perangkat komputasi antara simulasi dan eksperimen.

E. Diskusi dan Pembahasan

Berdasarkan hasil pengujian baik secara simulasi dan eksperimen, diperoleh bahwa adanya parameter waktu tunda dalam sistem *closed-loop* dapat mempengaruhi kinerja pengendalian. Hasil simulasi dengan asumsi waktu tunda yang bersifat konstan menunjukkan bahwa semakin besar nilai waktu tunda maka respon pengendalian yang dihasilkan semakin buruk dan begitu pula sebaliknya. Dengan demikian, diperlukan pengendalian yang bersifat kokoh sehingga perubahan waktu tunda dapat diatasi dengan melakukan *re-tuning* kembali parameter pengendali.

Dari sisi pengendali yang dirancang, PI-AW mampu menghasilkan respon terbaik jika dibandingkan dengan PI linier dan PI tersaturasi untuk beberapa skenario waktu tunda. Hal ini disebabkan karena PI-AW memiliki karakteristik mampu mengatasi adanya saturasi pada sinyal kendali. Namun demikian, PI-AW yang dirancang belum mampu mempertahankan kinerja pengendaliannya saat diintegrasikannya parameter waktu tunda dalam sistem *closed loop*. Hasil yang sama juga diperlihatkan saat dilakukannya eksperimen perangkat keras. Proses *re-tuning* kembali parameter PI-AW harus dilakukan saat digunakannya modul nirkabel antara *controller node* dan *plant node*. Hal ini terjadi karena adanya perbedaan karakteristik parameter waktu tunda antara simulasi dan eksperimen perangkat keras. Dengan demikian, diperlukan optimasi yang bersifat otomatis yang mampu mengatasi sifat stokastik dari parameter waktu tunda.

IV. KESIMPULAN

Pengendalian kecepatan motor DC berbasis WNCS telah berhasil dirancang dan diimplementasikan. Pengendali PI-AW digunakan karena memiliki karakteristik mampu mengatasi adanya waktu tunda dan saturasi pada sinyal kendali. Hasil simulasi menunjukkan bahwa PI-AW mampu menghasilkan respon yang lebih baik jika dibandingkan dengan PI linier. Setelah diintegrasikannya parameter waktu tunda minimum 0,64 s dan maksimum 1,12 s dari modul nRF24L01, PI-AW mampu menghasilkan respon dengan *settling time* 11,8 detik dan 16,2 detik. Hasil pengujian secara eksperimen dilakukan

menggunakan *re-tuning* konstanta PI-AW dengan metode *trial and error*. Hasilnya menunjukkan PI-AW mampu menghasilkan respon dengan *settling time* 0,7 s. Perbedaan karakteristik respon antara simulasi dan eksperimen terjadi karena terdapatnya perbedaan konstanta pengendalian dan karakteristik waktu tunda. Penelitian selanjutnya diperlukan dalam hal optimasi konstanta PID agar mampu mengatasi sifat stokastik dari parameter waktu tunda.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Unit Penelitian dan Pengabdian kepada Masyarakat (UPPM) Politeknik Negeri Bandung atas dukungan dana pada penelitian ini melalui Hibah Penelitian Mandiri dengan Nomor Kontrak: 560.20/PL1.R7/LT/2019.

REFERENSI

- [1] R. A. Gupta and M. Y. Chow, "Networked control system: Overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527-2535, 2010.
- [2] L. Eriksson, "PID controller design and tuning in networked control system," *Doctoral Dissertation*, Helsinki University of Technology, 2008.
- [3] V. Urgan, "Networked PID Controllers for Wireless Systems," *Master degree project*, KTH Electrical Engineering, 2010.
- [4] A. Najmurokhman, Y. B. Zainal, Sunubroto, and S. C. Sari, "Perancangan pengendali dengan umpan balik keadaan untuk networked control system," in *Prosiding Seminar Nasional Multidisiplin Ilmu*, Jakarta, 2011.
- [5] A. Najmurokhman, B. R. Trilaksono, A. S. Rohman, and Hendrawan, "Dissipative Controller Design for Networked Control Systems via the Markovian Jump System Approach," *J. Eng. Technol. Sci.*, vol. 45, no. 1, pp. 25-46, 2013.
- [6] A. Najmurokhman, P. H. Rusmin, B. R. Trilaksono, A. S. Rohman, Hendrawan, and Zulhariansyah, "Implementasi networked control system dalam pengendalian kecepatan motor DC melalui jaringan komunikasi," in *Prosiding EECCIS*, 2012.
- [7] M. Blaho, M. Urban, P. Fodrek, and M. Foltin, "Wireless Network Effect on PI and Type-2 Fuzzy Logic Controller," *International Journal of Communication*, vol. 6, no. 1, pp. 18-25, 2012.
- [8] S. M. Hassan, R. Ibrahim, N. Saad, V. S. Asirvadam, K. Bingi, and T. D. Chung, "Robustness and stability analysis of a predictive PI controller in wirelessHART network characterized by stochastic delay," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2605-2613, 2017.
- [9] A. R. Al Tahtawi and A. S. Rohman, "Perancangan Pengendali PI-AW pada Simulator Mobil Listrik

- untuk Kendali Kecepatan dan Torsi,” *Jurnal Teknik Elektro, Komputer, dan Informatika (ELECTRANS)*, vol. 14, no. 1, pp. 1-7, 2016.
- [10] P. Park, J. Araujo, and K. H. Johansson, “Wireless Networked Control System Co-Design,” in *Proc. International Conference on Networking, Sensing, and Control*, Delft, April 2011, pp. 486-491.
- [11] N. P. Aung, Z. M. Naing, M. M. M. Wai, and H. M. Tun, “Simulation of Wireless Networked Control System using TrueTime and Matlab,” *International Journal of Scientific and Technology Research*, vol. 5, no. 06, pp. 31-35, 2016.
- [12] N. Boughanmi, Y. Q. Song, and E. Rondeau, “Wireless Networked Control System using ZigBee/IEEE 802.15.4*,” in *Proc. of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, Moscow, June 2009, pp. 313-318.
- [13] K. Ogata, *Modern Control Engineering*, Prentice Hall, New Jersey, 2002.
- [14] D. Ibrahim, *Microcontroller Based Applied Digital Control*, John Wiley and Sons, England, 2006.
- [15] J. Goncalves, J. Lima, P. J. Costa, and A. P. Moreira, “Modeling and Simulation of The EMG30 Geared Motor with Encoder Resorting to SimTwo: The Official Robot @Factory Simulator,” 2013.
- [16] U. J. Shobrina, R. Primananda, and R. Maulana, “Analisis Kinerja Pengiriman Data Modul Transceiver nRF24L01, Xbee, dan WiFi ESP8266 pada Wireless Sensor Network,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 4, pp. 1510-1517, 2018.

