

Kendali PI-D dan Kontrol *Fuzzy* pada *Plant Level Air* Terdistribusi

ROBINSAR PARLINDUNGAN^{1*}, SUYANTO¹, FENI ISDARYANI¹, RANGGA VEGA SANTOSO²

¹Jurusan Teknik Elektro, Politeknik Negeri Bandung, Indonesia

²Mahasiswa D-IV Teknik Elektronika, Politeknik Negeri Bandung, Indonesia

*Email korespondensi: robinsar.p@polban.ac.id

Received 6 November 2021 | Revised 12 Desember 2021 | Accepted 27 Desember 2021

ABSTRAK

Penelitian ini bertujuan menginvestigasi algoritma kendali pada plant level air yang terdistribusi pada dua tangki. Terdapat dua metode yang digunakan yakni kendali PI-D yang ditempatkan pada masing-masing unit pengendali lokal (LCU) dan kendali Fuzzy sebagai unit pengendali utama (MCU) dimana performa sistem dapat di-monitoring pada Human Machine Interface (HMI). Kendali PI-D (tipe B) dipilih untuk mengantisipasi persoalan setpoint kick yang sering muncul ketika digunakan kendali PID konvensional. Hasil penelitian menunjukkan respons kendali PI-D dapat mengatasi setpoint kick ditandai dengan perubahan smooth ketika setpoint berubah. Respons kendali PI-D pada HMI menunjukkan hasil yang baik direpresentasikan melalui error steady state dapat bernilai 0, overshoot 0% dan settling time 0,01 detik. Disamping itu, kontrol Fuzzy yang digunakan pada MCU dapat menghasilkan nilai setpoint (SV) yang tepat untuk masing-masing LCU, sehingga diperoleh respons kendali di atas.

Kata kunci: LCU, MCU, PI-D, Level Air, Logika Fuzzy

ABSTRACT

This study aims to investigate the control algorithm on the plant water level distributed in two tanks. There are two methods used, namely PI-D control placed on each local control units (LCU) and Fuzzy control as the Main Control Unit (MCU) and synchronization for Human Machine Interface (HMI) and monitoring. The PI-D control (type-B) was chosen to anticipate the setpoint kick problem that often arises when conventional PID controls are used. The results showed that the PI-D control response was able to overcome the setpoint kick which represented by smooth acting of actuator response. Response of the PI-D control on the HMI shows a good response by a steady state error of 0, an overshoot of 0%, and a settling time of 0.01 seconds. Besides that, Fuzzy logic used in the MCU can produce the right setpoint (SV) value for each LCU, so that the control response above is obtained.

Keywords: LCU, MCU, PI-D, Water Level, Fuzzy Logic

1. PENDAHULUAN

Di industri, pengukuran level dilakukan di banyak tangki yang tersebar pada beberapa area yang berjauhan, dan tangki-tangki tersebut saling terkoneksi (*interconnected*) (Luyben, 2020b). Dampaknya adalah kekeliruan pada satu bagian mempengaruhi bagian lain. Ditambah lagi, pengukuran level ditujukan untuk proses pengendalian antar tangki, sehingga seringkali terjadi ketidaksesuaian pengukuran level akibat adanya tundaan (*delay*) dari transmisi bagian sensor ke pengendali atau sebaliknya (Li & Zhang, 2021). Oleh karenanya, persoalan pengukuran dan pengendalian antar bagian yang saling terkoneksi dan berjauhan jaraknya diatasi melalui penggunaan sistem kendali terdistribusi (Li & Zhang, 2021) (Xu, dkk, 2020).

Persoalan integrasi dan koordinasi *plant* proses yang berjauhan memiliki kelemahan diantaranya adanya tundaan, hilang dan tabrakan informasi, isu beban energi (*overload*) dll. Beberapa penelitian telah mencoba mengatasi masalah-masalah tersebut, dengan menerapkan teknik kendali klasik berbasis model *plant* linier (Ahmad, dkk, 2012) (Negash Getu, dkk, 2016) untuk mengatasi persoalan tundaan (*delay*) dari transmisi sensor ke pengendali (kontroler), pendekatan kecerdasan buatan dengan *Fuzzy* dan algoritma genetik (Aydogmus, 2015) (Komenda, dkk, 2010) untuk mengatasi hilangnya atau tabrakan informasi, dan model kendali prediktif untuk meningkatkan efisiensi energi (Luyben, 2020a) (Pasha & Ayub, 2021). Meskipun demikian, penelitian-penelitian tersebut masih memiliki keterbatasan dalam meningkatkan kecepatan atau performansi pengendalian dan akurasi pengukuran.

Selain itu, sejumlah penelitian juga telah dilakukan untuk menemukan solusi dalam masalah level fluida di industri. Penelitian (G. Chen & Tat Pham, 2008) (van Schuppen, dkk, 2011) menggunakan algoritma genetik untuk mengoptimasi kendali PID yang digunakan; dilanjutkan (Ahmad, dkk, 2012) menggunakan *Fuzzy* untuk mengendalikan level air. Namun, penelitian yang telah dilakukan menerapkan model linier dalam penyelesaiannya. Dalam kasus level fluida, model *plant* yang digunakan adalah non-linier, sehingga perlu pendekatan solusi lain dalam optimasi kendali yang digunakan (Chen, 2021).

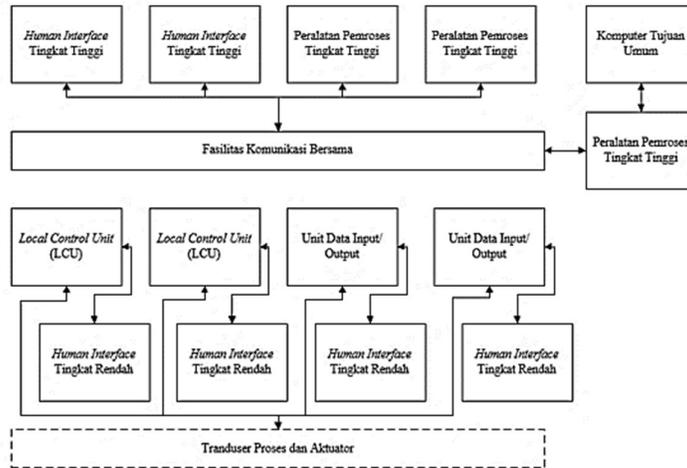
Selain itu, *plant* proses yang tidak berada dalam lokasi yang sama juga memberikan kontribusi kekeliruan hasil pengukuran yang menyebabkan kendali tidak dapat bekerja optimal. Dalam kasus semacam ini diterapkan kendali terdistribusi sebagai solusinya. Penelitian lain mencoba menyelesaikan dengan mereduksi waktu komputasi (Wang & Wang, 2018) (Yordanova, 2015), meminimalkan hilangnya data (Pambakian, 2011) (Wang & Wang, 2018), mengoptimasi transmisi data dan layanan paket (*packet of service*) (Chen & Tat Pham, 2008) (Narayan, 2014).

Penelitian ini berupaya untuk mengembangkan sistem pengendali terdistribusi yang ditujukan untuk optimasi performansi pengendalian level fluida dan meningkatkan kecepatan kendali dengan menerapkan beberapa kondisi *real* yang terjadi di industri. Persoalan yang muncul adalah bagaimana mendesain algoritma yang tepat pada unit pengendali utama (MCU) dan koordinasi yang dilakukan dengan unit pengendali lokal (LCU) (Prakash, dkk, 2021). Jika hanya mengandalkan kendali PID konvensional maka masalah *set-point kick* akan muncul, terutama ketika sinyal yang digunakan sinyal step. Isu *set-point kick* dibahas detail oleh (Ogata, 2010), dengan menempatkan parameter *Derivative* yang mendapat *feedback*. Peristiwa *set-point kick* perlu diatasi karena berdampak pada respons aktuatur yang memiliki karakteristik responsif terhadap nilai referensi. Hal ini sangat berbahaya jika terjadi perubahan *setpoint*, maka level air melebihi batas dan terjadi peluapan, dan aktuatur cepat rusak akibat *setpoint kick*.

2. METODE PENELITIAN

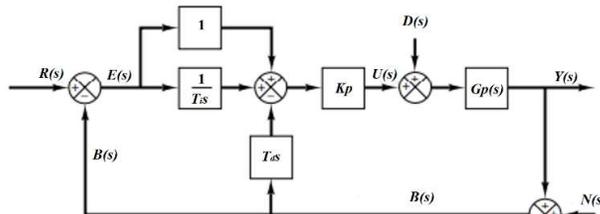
2.1 Sistem Kendali Terdistribusi

Sistem kendali terdistribusi (*Distributed Control System*) adalah sekumpulan sistem kendali lokal yang tersebar yang diintegrasikan dalam sistem kendali kerjasama (*cooperating control*) untuk mengeliminasi kompleksitas jika hanya menggunakan sistem kendali tunggal. Sistem kendali kerjasama sering juga disebut MCU, sedangkan sistem kendali lokal yang tersebar di beberapa titik disebut sebagai LCU (**Aboelhasan, dkk, 2020**) (**van Schuppen, dkk, 2011**). MCU berfungsi sebagai kendali utama sistem DCS dan memberikan nilai yang dibutuhkan pada masing-masing LCU, serta berfungsi mentransmisikan data jika akan ditampilkan pada display *Human Machine Interface* (HMI). Diagram blok sistem DCS secara umum ditunjukkan pada Gambar 1.



Gambar 1. Diagram Blok Sistem DCS

Sistem kendali di industri umumnya menggunakan kendali PID karena konsep sederhana dan mudah diimplementasikan. Namun, sering ditemukan masalah pada kendali PID konvensional yakni perubahan respons yang berubah-ubah secara mendadak akibat perubahan *setpoint*, yang dikenal dengan istilah *setpoint kick*. Perubahan semacam ini tidak diinginkan di dalam sistem kendali karena menurunkan kualitas performansi. Oleh karena itu, penelitian yang dilakukan oleh (**Ogata, 2010**) mengatasi persoalan di atas dengan melakukan perubahan fungsi derivatif hanya pada bagian umpan-balik (*feedback*) bukan pada bagian *error*, sehingga diferensiasi hanya terjadi pada sinyal umpan-balik dan tidak pada sinyal referensi. Konfigurasi semacam ini disebut juga kendali tipe PI-D, yang ditunjukkan pada Gambar 2.



Gambar 2. Diagram Blok Kendali PI-D

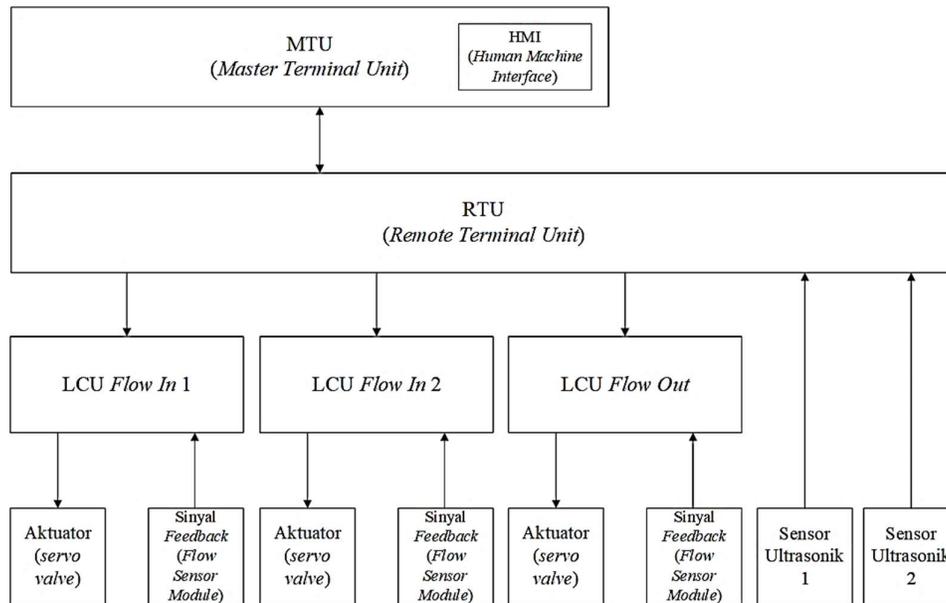
Berdasarkan Gambar 2, persamaan sistem kendali dapat ditulis menjadi:

$$u(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int e(t) dt - K_p T_d \frac{d(PV(t))}{dt} \quad (1)$$

Terlihat dari Persamaan (1) kendali PI-D memiliki respons yang lebih tidak reaktif terhadap perubahan *setpoint* dibandingkan kendali konvensional PID, sehingga jika ada perubahan *setpoint* secara mendadak maka respons berubah secara perlahan.

Metode penelitian untuk mendapatkan dan menguji kendali PI-D dilakukan dengan membuat konfigurasi sistem *plant* level air terdistribusi. Konfigurasi perangkat keras yang menjelaskan lebih lengkap bagian-bagian sistem pada penelitian ini ditunjukkan pada

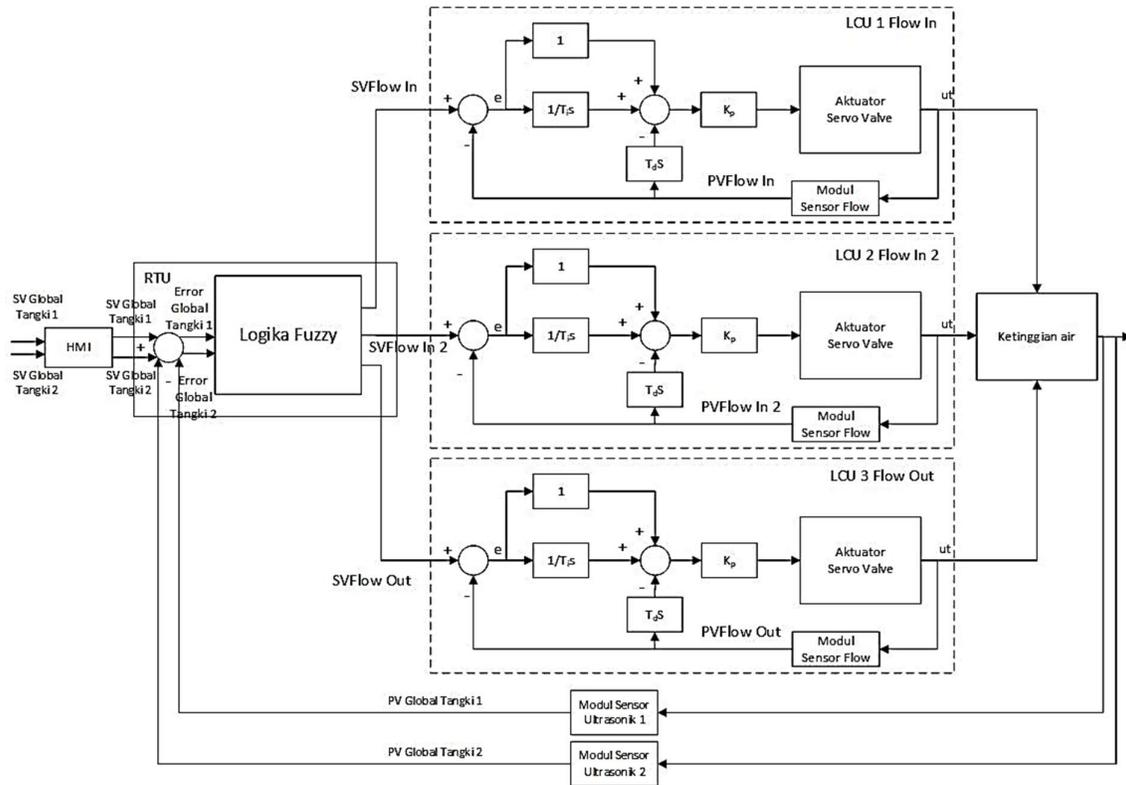
Gambar 3. Terdapat tiga komponen utama pada sistem, yaitu HMI, *Remote Terminal Unit* / *Master Terminal Unit* (RTU/MTU) dan *Local Control Unit* (LCU). HMI berfungsi sebagai antarmuka operator untuk melakukan pemantauan (*monitoring*) level air secara *real-time*, sekaligus digunakan untuk mengubah nilai *setpoint* global pada tangki (dalam sistem ini digunakan 2 tangki yang dikendalikan) dan memasukkan nilai parameter kendali PI-D secara manual. RTU/MTU berfungsi memberikan nilai *setpoint* pada masing-masing LCU dan menerima 2 *input Fuzzy*, berupa *error global*/hasil pembacaan sensor ultrasonik. LCU berfungsi mengendalikan aliran air baik yang masuk ke tangki 1 dan 2, maupun yang keluar dari tangki 2.



Gambar 3. Diagram Blok Sistem yang Diajukan

Terdapat prinsip kerja sistem ini adalah level air yang dijadikan *setpoint* ditentukan melalui HMI, yang dinyatakan sebagai SV global. Selanjutnya dibandingkan dengan hasil pembacaan sensor ultrasonik pada masing-masing tangki sebagai umpan-balik, yang dinyatakan sebagai PV global, menghasilkan nilai *error global* sebagai *input* untuk *Fuzzy*. Kontrol *Fuzzy* digunakan pada MCU untuk memproses *input* berupa *error global* menghasilkan *setpoint* (*setpoint* lokal) untuk masing-masing LCU dengan nilai tertentu berdasarkan aturan-aturan (*rules*) yang dibuat. *Setpoint* lokal ini direpresentasikan sebagai SVFlow In 1, SVFlow In 2, dan SVFlow Out. Masing-masing LCU akan dikendalikan oleh kendali PI-D agar dapat mengatur aliran air yang masuk ke dalam dan keluar tangki serta mengatasi fenomena *setpoint kick*. Deskripsi lengkap ditunjukkan pada Gambar 4. Secara sederhana, terdapat 2 metode pada *plant* kendali level air

ini, yaitu metode PI-D untuk mengatasi *setpoint kick* dan metode *Fuzzy* untuk optimasi performansi *error steady state*.

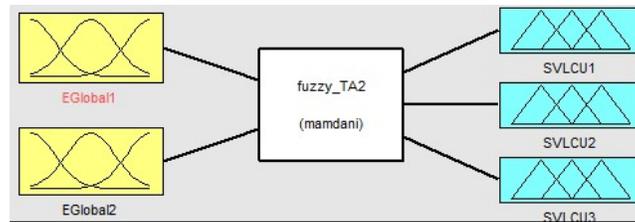


Gambar 4. Diagram Blok Kendali

2.2 Desain Kontrol Fuzzy

Desain algoritma *Fuzzy* dilakukan menggunakan MATLAB dengan metode inferensi Mamdani. Terdapat 2 *input*, yakni *Error Global 1* (selisih SV Global 1 dan PV Global 1) dan *Error Global 2* (selisih SV Global 2 dan PV Global 2). *Output* dari kontrol *fuzzy* adalah SV LCU 1, SV LCU 2, dan SV LCU 3 seperti ditunjukkan pada

Gambar 5.

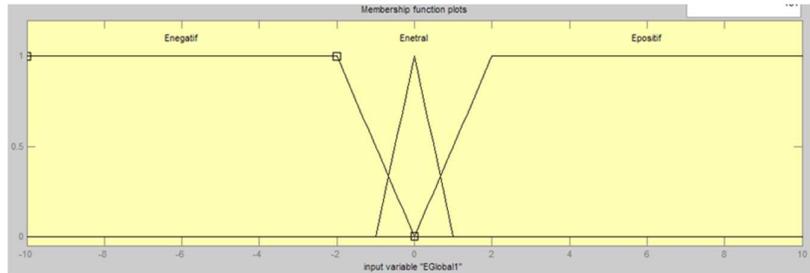


Gambar 5. Desain Logika Fuzzy

Proses fuzzifikasi berfungsi mengubah besaran *crisp* menjadi besaran *Fuzzy*. Berikut ditentukan fungsi keanggotaan dari masing-masing *input* dan *output*:

1. *Error Global*

Fungsi keanggotaan *input Error Global* ditunjukkan Gambar 6. *Input Error Global* terdiri dari dua *input*, yaitu *Input Error Global 1* dan *Error Global 2* yang masing-masing memiliki tiga fungsi keanggotaan yaitu Enegatif yang artinya *error global* bernilai negatif, Enetral yang bermakna *error global* = 0, dan Epositif yang berarti *error global* bernilai positif seperti terlihat pada Tabel 1.



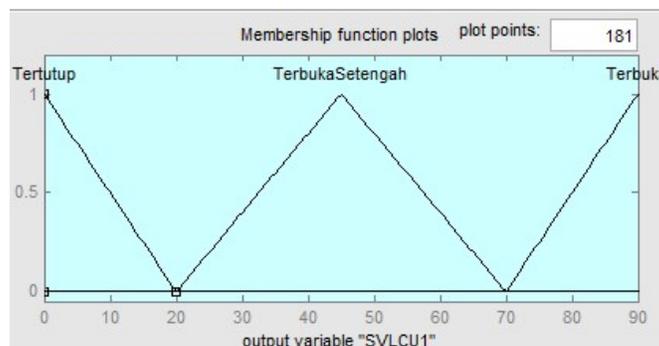
Gambar 6. Fungsi Keanggotaan *Input Error Global*

Tabel 1. Fungsi Keanggotaan *Input Error Global*

Enegatif		Enetral		Epositif	
1	$X \leq -2$	1	$X = 0$	1	$X \geq 2$
0	$X \geq 0$	$\frac{X + 1}{1}$	$-1 \leq X \leq 0$	$\frac{X}{2}$	$0 \leq X \leq 2$
$\frac{-X}{2}$	$-2 \leq X \leq 0$	0	$X \leq -1$ atau $X \geq 1$	0	$X \leq 0$
		$\frac{1 - X}{1}$	$\leq X \leq 1$		

2. SV LCU

Output SV LCU memiliki tiga fungsi keanggotaan berbentuk segitiga dan segitiga terbuka dapat dilihat pada Gambar 7. Kondisi tertutup yang bernilai 0, terbuka setengah yang bernilai 45 dan terbuka yang bernilai 90. Nilai-nilai ini berkorelasi terhadap bukaan servo. Masing-masing batasan ditunjukkan pada Tabel 2.



Gambar 7. Fungsi Keanggotaan *Output* SV LCU

Tabel 2. Nilai Batas Fungsi Keanggotaan *Output* SV LCU

Tertutup		Terbuka Setengah		Terbuka	
1	X=0	1	X = 45	1	X = 90
0	$X \geq 20$	$\frac{X - 20}{25}$	$20 \leq X \leq 45$	$\frac{X - 70}{20}$	$70 \leq X \leq 90$
$\frac{20 - X}{20}$	$0 \leq X \leq 20$	0	$X \leq 20$ atau $X \geq 70$	0	$X \leq 70$
		$\frac{70 - X}{25}$	$45 \leq X \leq 70$		

Selanjutnya dilakukan proses menentukan aturan-aturan (*rules*) Fuzzy yang dibuat adalah sebagai berikut:

1. If (EGlobal1 is Epositif) and (EGlobal2 is Epositif) then (SVLCU1 is Terbuka) (SVLCU2 is TerbukaSetengah) (SVLCU3 is Tertutup)
2. If (EGlobal1 is Epositif) and (EGlobal2 is Enetral) then (SVLCU1 is Terbuka) (SVLCU2 is TerbukaSetengah) (SVLCU3 is TerbukaSetengah)
3. If (EGlobal1 is Enetral) and (EGlobal2 is Enetral) then (SVLCU1 is Tertutup) (SVLCU2 is Tertutup) (SVLCU3 is Tertutup)
4. If (EGlobal1 is Enetral) and (EGlobal2 is Enegatif) then (SVLCU1 is Tertutup) (SVLCU2 is Tertutup) (SVLCU3 is Terbuka)
5. If (EGlobal1 is Enetral) and (EGlobal2 is Epositif) then (SVLCU1 is Tertutup) (SVLCU2 is Terbuka) (SVLCU3 is Tertutup)
6. If (EGlobal1 is Enegatif) and (EGlobal2 is Enetral) then (SVLCU1 is TerbukaSetengah) (SVLCU2 is Terbuka) (SVLCU3 is TerbukaSetengah)
7. If (EGlobal1 is Epositif) and (EGlobal2 is Enegatif) then (SVLCU1 is Terbuka) (SVLCU2 is TerbukaSetengah) (SVLCU3 is Terbuka)
8. If (EGlobal1 is Enegatif) and (EGlobal2 is Epositif) then (SVLCU1 is TerbukaSetengah) (SVLCU2 is Terbuka) (SVLCU3 is Tertutup)
9. If (EGlobal1 is Enegatif) and (EGlobal2 is Enegatif) then (SVLCU1 is TerbukaSetengah) (SVLCU2 is Terbuka) (SVLCU3 is Terbuka)

dari *rules* di atas, kemudian dilakukan proses *defuzzifikasi* untuk mengubah besaran Fuzzy menjadi *crisp*. Proses ini dilakukan dengan menggunakan metode *Center of Area* (COA) diskrit dengan formula pada Persamaan 2.

$$COA \text{ diskrit} = \frac{\sum_{j=1}^n \mu_c(Z_j) Z_j}{\sum_{j=1}^n \mu_c(Z_j)} \quad (2)$$

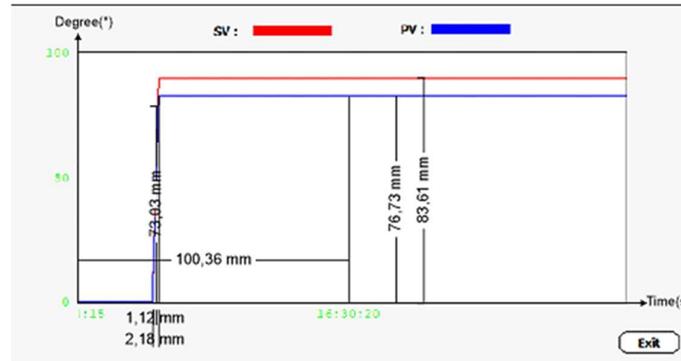
dimana n adalah jumlah dari kuantifikasi level *output*, i adalah nilai perulangan data ke- n , μ_c adalah nilai anggota dalam c , dan Z_j adalah jumlah *output* kontrol pada kuantifikasi level j .

3. HASIL DAN PEMBAHASAN

3.1 Respons Kendali LCU

Pengujian respons kendali pada masing-masing LCU bertujuan untuk melihat parameter-parameter respons *transient* sudah sesuai dengan kriteria yang ditetapkan. Pengujian dilakukan dengan membandingkan hasil respons *transient* masing-masing parameter pada HMI dengan *serial plotter*. Pada respons kendali PID untuk LCU 1 yang terlihat pada *serial plotter* dengan parameter tuning yaitu $K_p = 0,5$; $T_i = 6,5$; dan $T_d = 1,5$ dengan hasil respons

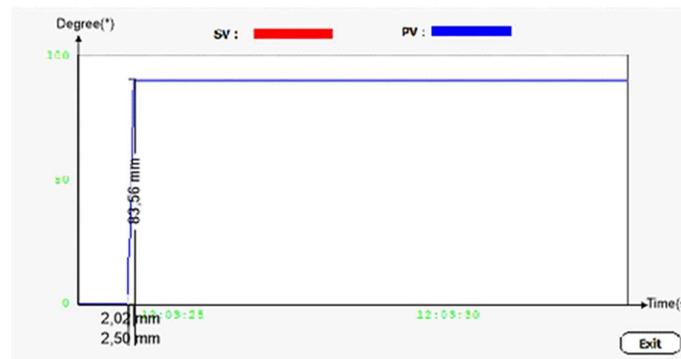
transient yang didapat adalah *rise time* sebesar 0,4 detik, % *overshoot* 0%, *settling time* 0,22 detik, dan tidak ada *error steady state*. Namun, ketika dimasukkan ke dalam tampilan HMI, maka terdapat perbedaan yang dominan pada *error steady state* yang meningkat seperti tampak pada Gambar 8. Hal ini terjadi akibat adanya faktor *setpoint kick*, dimana komponen derivatif mengakibatkan melonjaknya *error* yang terjadi.



Gambar 8. Respons Kendali PID LCU 1 pada HMI

Selanjutnya membandingkan respons *transient* kendali PI-D LCU 1 pada *serial plotter* dan HMI dengan parameter kendali yang sama, yakni $K_p = 0,5$, $T_i = 6,5$, dan $T_d = 1,5$. Respons sistem ditunjukkan oleh Gambar 9, dengan nilai *rise time* 0,4 detik, *overshoot* 0%, *settling time* 0,21 detik, dan tidak ada *error steady state*. Pada tampilan HMI, respons kendali PI-D menghasilkan respons yang hampir sama, dengan tidak adanya *error steady state*. Perbandingan respons *transient* kendali LCU 1 dapat dilihat pada Tabel 3. Dengan cara yang sama, maka diperoleh juga perbandingan respons *transient* kendali LCU 2 dan LCU 3, yang berturut-turut ditunjukkan pada

Tabel 4 dan Tabel 5.



Gambar 9. Respons Kendali PI-D LCU 1 pada HMI

Tabel 3. Tabel Perbandingan Respons *Transient* Kendali LCU 1

	<i>Rise Time (s)</i>	<i>Overshoot (%)</i>	<i>Settling Time (s)</i>	<i>Error Steady State (degree)</i>
Spesifikasi Penelitian	< 10	< 10	< 10	0
Respons PID (<i>serial plotter</i>)	0,4	0	0,22	0
Respons PID (HMI)	0,11	0	0,06	6,88
Respons PI-D (<i>serial plotter</i>)	0,4	0	0,21	0
Respons PI-D (HMI)	0,12	0	0,1	0

Tabel 4. Tabel Perbandingan Respons *Transient* Kendali LCU 2

	<i>Rise Time (s)</i>	<i>Overshoot (%)</i>	<i>Settling Time (s)</i>	<i>Error Steady State (degree)</i>
Spesifikasi Penelitian	< 10	< 10	< 10	0
Respons PID (<i>serial plotter</i>)	0,42	0	3,46	0
Respons PID (HMI)	0,11	0	0,1	19,31
Respons PI-D (<i>serial plotter</i>)	0,41	0	0,22	0
Respons PI-D (HMI)	0,12	0	0,09	0

Tabel 5. Tabel Perbandingan Respons *Transient* Kendali LCU 3

	<i>Rise Time (s)</i>	<i>Overshoot (%)</i>	<i>Settling Time (s)</i>	<i>Error Steady State (degree)</i>
Spesifikasi Penelitian	< 10	< 10	< 10	0
Respons PID (<i>serial plotter</i>)	0,38	0	1,76	0
Respons PID (HMI)	0,12	0	0,08	9,45
Respons PI-D (<i>serial plotter</i>)	0,41	0	0,22	0
Respons PI-D (HMI)	0,13	0	0,09	0

Jika diamati pada Tabel 3,

Tabel 4, dan Tabel 5 pada sistem yang menggunakan kendali PID konvensional menunjukkan nilai *rise time* dan *settling time* yang lebih baik, namun memiliki kekurangan yang signifikan, yaitu besarnya *error steady state*, dimana parameter ini diinginkan bernilai nol sesuai dengan

kriteria yang ditentukan diawal. Kebalikan dari respons kendali PID konvensional, kendali PI-D menunjukkan performansi yang lebih baik meski ada perbedaan yang kecil pada nilai *rise time* dan *settling time*. Hal ini menunjukkan bahwa kendali PI-D lebih berhasil mengatasi persoalan *setpoint kick* dibandingkan dengan kendali PID konvensional dengan ditandai *error steady state* yang bernilai nol, sehingga tidak terdapat perbedaan antara SV dengan PV.

3.3 Pengujian Algoritma *Fuzzy*

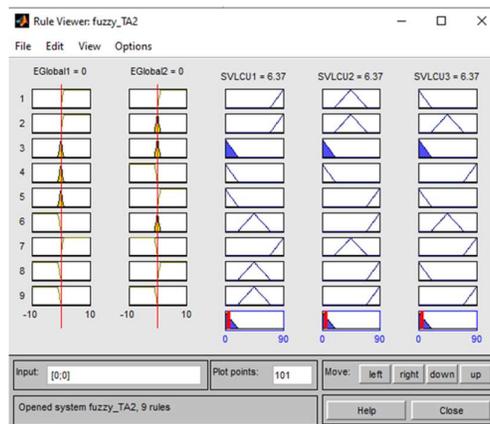
Pengujian algoritma logika *Fuzzy* dilakukan dengan membandingkan hasil logika *Fuzzy* berupa *Setpoint Value* untuk masing – masing LCU pada *Serial monitor* Arduino dan MATLAB. Pengujian dan pembuktian dilakukan pada satu aturan *Fuzzy*, hal ini ditujukan untuk membuktikan bahwa pendefinisian algoritma *Fuzzy* ke dalam pemrograman Arduino sudah betul, dengan tahap sebagai berikut:

Pengujian saat *Error Global 1* bernilai nol dan *Error Global 2* bernilai 0.
Pada *rule* menunjukkan:

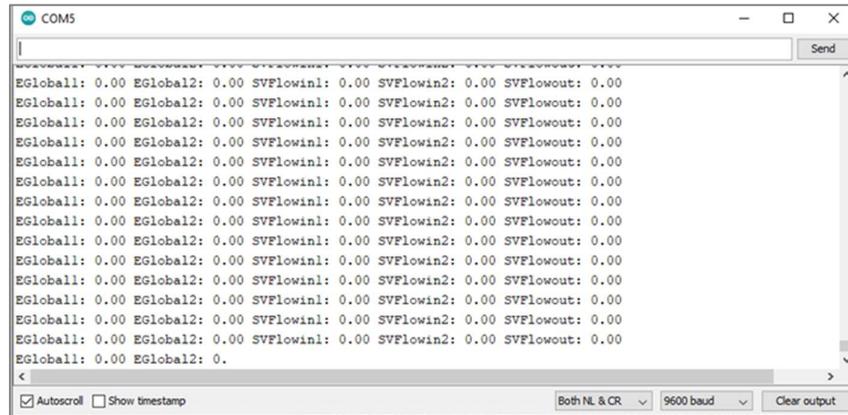
“If (EGlobal1 is Enetral) and (EGlobal2 is Enetral) then (SVLCU1 is Tertutup) (SVLCU2 is Tertutup) (SVLCU3 is Tertutup)”

Nilai *Error Global 1* didapatkan dengan memperoleh selisih nilai *Setpoint Global* pada tangki level 1 dengan *Process Value Global 1* tangki level 1 sedangkan *Error Global 2* didapatkan dengan memperoleh selisih nilai *Setpoint Global* pada tangki level 2 dengan *Process Value Global* tangki level 2. Kedua *Process Value* adalah hasil dari pembacaan sensor ultrasonik pada kedua tangki. Enetral adalah ketika *Error Global* menunjukkan nilai 0 dan keadaan tertutup saat *Setpoint Value* untuk motor servo menunjukkan nilai 0 derajat. Maka dapat dilihat hasil pada MATLAB seperti yang ditunjukkan oleh Gambar 10. Hasil pengujian logika *Fuzzy* pada Arduino di tunjukkan pada tampilan serial monitor pada Gambar 11.

Hasil *Fuzzy* menunjukkan bahwa, nilai *Setpoint Value* di semua LCU pada MATLAB bernilai 6,37 sedangkan pada Serial monitor bernilai nol. Maka dapat disimpulkan bahwa hasilnya sesuai antara desain logika *Fuzzy* pada MATLAB dengan hasil pada *serial monitor*. pengujian algoritma



Gambar 10. Pengujian saat *Error Global 1* dan *Error Global 2* Bernilai Nol pada MATLAB

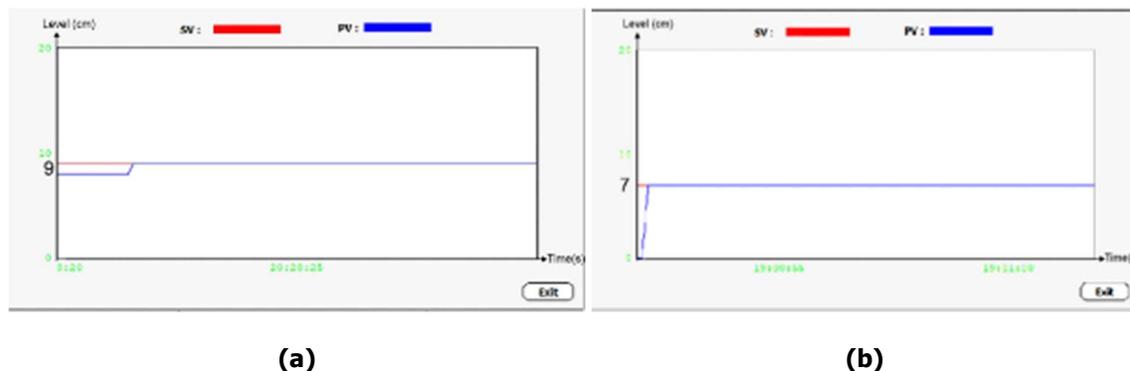


Gambar 11. Pengujian saat *Error Global 1* dan *Error Global 2* Bernilai Nol pada Serial Monitor

3.4 Pengujian Keseluruhan Sistem Level Air

Pengujian dilakukan dengan menentukan SV Global 1 dan SV Global 2 pada kedua tangki atau ketinggian air yang diinginkan. Pada percobaan pertama menggunakan ketinggian air yang diinginkan pada tangki 1 adalah 9 cm sedangkan ketinggian air yang diinginkan pada tangki 2 adalah 7 cm. Nilai tersebut di-*input* melalui program dan memastikan apakah pada HMI sudah berhasil menampilkan nilai SV Global pada kedua tangki yang diinginkan.

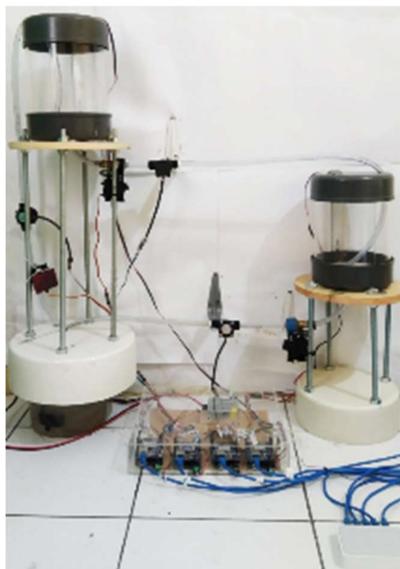
Pada bagian SV LCU terdapat nilai yang dikirimkan dan ditampilkan pada HMI. Nilai tersebut adalah hasil dari logika *Fuzzy* yang terdapat pada MTU. Nilai SV LCU 1 yaitu 90 derajat, SV LCU 2 yaitu 45 derajat, dan SV LCU 3 yaitu 0 derajat saat *Error Global 1* dan *Error Global 2* menunjukkan *Error* Positif atau nilai ketinggian air yang terbaca (*PV Global*) masih kurang dari nilai *SV Global*. Ini sesuai dengan *rules* pada logika *Fuzzy* dimana saat *Error Global 1* yaitu *Error* positif dan *Error Global 2* yaitu *Error* positif, maka SV LCU 1 adalah *valve* terbuka penuh, SV LCU 2 adalah *valve* terbuka setengah, dan SV LCU 3 adalah *valve* tertutup. Dan setelah beberapa saat kemudian, kedua SV Global dapat dicapai ketinggiannya. Gambar 12 (a) dan (b) menampilkan hasil akhir level air pada plant yang sesuai dengan nilai *setpoint* yang diberikan yakni 9 cm untuk tangki 1 dan 7 cm untuk tangki 2.



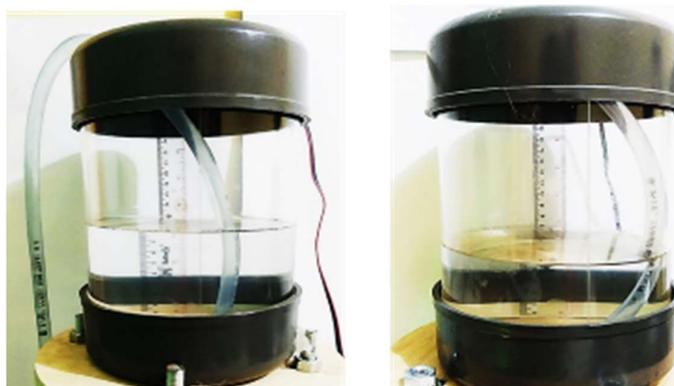
**Gambar 12. (a) Grafik Respons Saat PV Global 1 Mencapai SV Global 1 Bernilai 9 cm.
(b) Grafik Respons Saat PV Global 2 Mencapai SV Global 2 Bernilai 7 cm**

Untuk memastikan bahwa ketinggian air yang dicapai sesuai atau tidak maka dilakukan perbandingan dengan pengukuran pada tangki sebenarnya. Adapun realisasi dari sistem ini

dapat dilihat pada Gambar 13, sedangkan Gambar 14 menunjukkan hasil pengujian ketinggian air pada tangki yang sebenarnya.

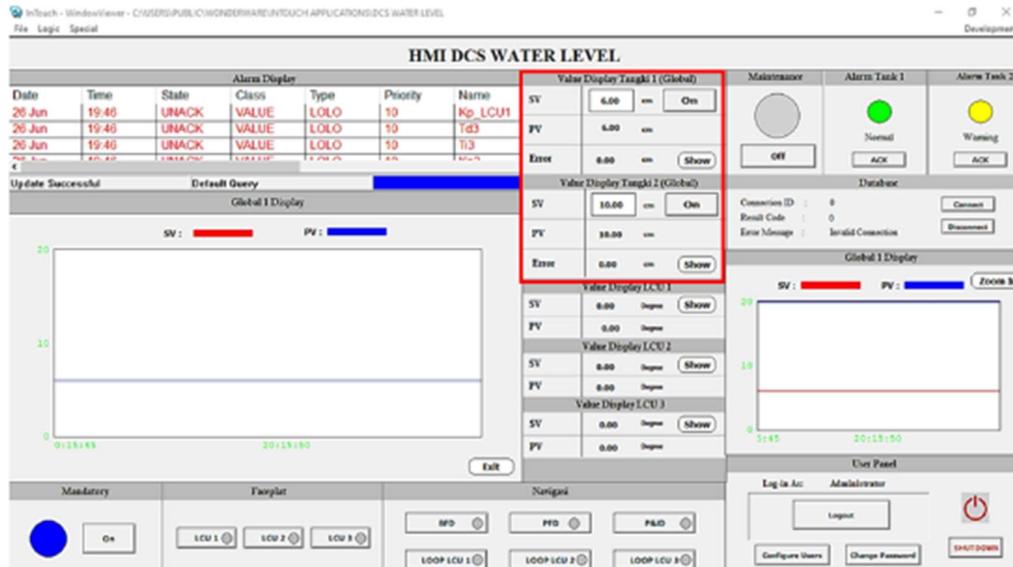


Gambar 13. Realisasi Sistem Secara Keseluruhan

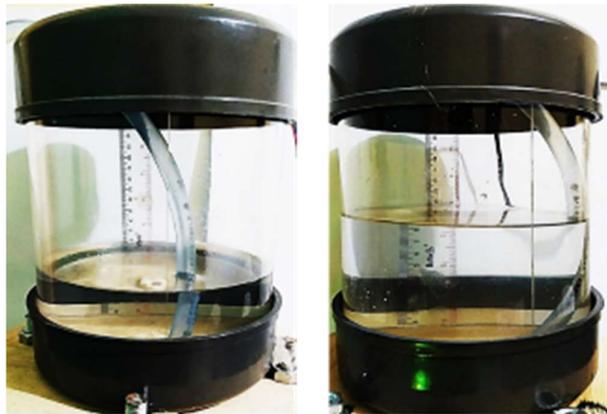


Gambar 14. Hasil Pengukuran Real *Setpoint* (Kiri: Level Tangki 1; Kanan: Level Tangki 2)

Selanjutnya dilakukan percobaan dengan mengubah nilai *SV Global* 1 dan 2 secara manual dengan cara memasukkan nilai *SV* yang diinginkan langsung dari HMI. *SV Global* 1 bernilai 6 cm dan *SV Global* 2 bernilai 10 cm, adapun tampilan HMI dapat dilihat pada Gambar 15. Tampilan HMI terdiri dari *Alarms display*, *user panel*, *mandatory*, *values display*, *database*, dan *output response display (Global display)*.



Gambar 15. HMI dengan Plotting SV Global 1 Bernilai 6 cm dan SV Global 2 bernilai 10 cm



Gambar 16. Hasil Pengukuran Real *Setpoint* (Kiri: Level Tangki 1; Kanan: Level Tangki 2)

Pada menu Global 1 Display (Gambar 15) dapat dilihat bahwa ketinggian air pada kedua tangki yang diinginkan berhasil dicapai dengan *Error Global 1* dan *2* yaitu bernilai nol. Nilai SV pada setiap LCU bernilai 0 derajat saat *Error Global* bernilai nol atau Error Netral. Ini sesuai dengan *rules* pada logika *Fuzzy* dimana saat *Error Global 1* dan *Error Global 2* bernilai nol atau *Error* Netral maka nilai SV ketiga LCU menjadi *valve* menutup atau bukaan servo sebesar 0 derajat. Gambar 16 menunjukkan respons pengisian air pada tangki yang sebenarnya.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan algoritma kendali pada sistem tangki yang terdistribusi dengan menggunakan metode kendali PI-D dan kontrol *Fuzzy*. Kendali PI-D menunjukkan respons yang tidak reaktif dari perubahan *setpoint* sehingga dapat mengatasi fenomena *setpoint kick*. Hal ini ditunjukkan dengan *rise time* kendali PI-D lebih lama dibanding PID konvensional. Disamping itu, respons kendali PI-D mampu membuat *error steady state* bernilai nol, yang sesuai dengan kriteria yang ditetapkan. Sedangkan kontrol *Fuzzy* yang digunakan mampu memproses nilai *Error Global 1* dan *Error Global 2* menjadi nilai *setpoint* (SV) untuk

masing-masing LCU sesuai dengan *rules* dan dibuktikan melalui perbandingan pada MATLAB dan *serial plotter*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada P3M POLBAN yang telah mendanai penelitian ini melalui skema Penelitian Mandiri DIPA Politeknik Negeri Bandung Tahun Anggaran 2021 (No. Kontrak : 105.110/PL1.R7/PG.00.03/2021).

DAFTAR RUJUKAN

- Aboelhasan, A., Abdelgeliel, M., Zakzouk, E. E., & Galea, M. (2020). Design and implementation of model predictive control based pid controller for industrial applications. *Energies*, *13*(24). <https://doi.org/10.3390/en13246594>
- Ahmad, A., Redhu, V., Gupta, U., Year, nd, & Devi, L. (2012). Liquid Level Control By Using Fuzzy Logic Controller. *International Journal of Advances in Engineering & Technology*, *4*.
- Aydogmus, Z. (2015). A real-time robust fuzzy-based level control using programmable logic controller. *Elektronika Ir Elektrotehnika*, *21*(1), 13–17. <https://doi.org/10.5755/j01.eee.21.1.7812>
- Chen, G., & Tat Pham, T. (2008). *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems Introduction to*.
- Chen, L. (2021). Principle and Simulation PID Controller of Liquid Level System. *Journal of Physics: Conference Series*, *1757*(1). <https://doi.org/10.1088/1742-6596/1757/1/012187>
- Komenda, J., Masopust, T., & van Schuppen, J. H. (2010). *Supervisory Control Synthesis of Discrete-Event Systems using Coordination Scheme*. <https://doi.org/10.1016/j.automatica.2011.07.008>
- Li, L., & Zhang, Y. (2021). Distributed preview control for large-scale systems with time-varying delay. *ISA Transactions*, *109*. <https://doi.org/10.1016/j.isatra.2020.09.005>
- Luyben, W. L. (2020a). Liquid level control: Simplicity and complexity. *Journal of Process Control*, *86*. <https://doi.org/10.1016/j.jprocont.2019.12.008>
- Luyben, W. L. (2020b). Control of a two-pressure distillation column. *Journal of Process Control*, *92*. <https://doi.org/10.1016/j.jprocont.2020.06.015>
- Narayan, S. (2014). *Improving Network Performance: An Evaluation of TCP/UDP on Networks A Doctor of Computing Thesis*.

- Negash Getu, B., Khaimah, A., & al Khaimah, R. (2016). Water Level Controlling System Using Pid Controller. *International Journal of Applied Engineering Research*, 11(23). <http://www.ripublication.com>
- Ogata, Katsuhiko. (2010). *Modern control engineering*. Prentice-Hall.
- Pambakian, N. (2011). LQG coordination control. *IEEE/ACM Transactions on Computational Biology and Bioinformatics - TCBB*.
- Pasha, S. A., & Ayub, A. (2021). Zero-dynamics attacks on networked control systems. *Journal of Process Control*, 105. <https://doi.org/10.1016/j.jprocont.2021.07.010>
- Prakash, S., Nagarajan, S., & Thangavelusamy, D. (2021). Comparison and analysis of hybrid optimization algorithms for water level control of drum boiler system. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2020.12.251>
- van Schuppen, J. H., Boutin, O., Kempker, P. L., Komenda, J., Masopust, T., Pambakian, N., & Ran, A. C. M. (2011). Control of Distributed Systems: Tutorial and Overview. *European Journal of Control*, 17, 5–6. <https://doi.org/10.3166/ejc.17.579-602>
- Wang, L., & Wang, X. V. (2018). *Cloud-Based Cyber-Physical Systems in Manufacturing*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-67693-7>
- Xu, D., Zhu, F., Zhou, Z., & Yan, X. (2020). Distributed fault detection and estimation in cyber-physical systems subject to actuator faults. *ISA Transactions*, 104. <https://doi.org/10.1016/j.isatra.2019.12.002>
- Yordanova, S. (2015). Intelligent Approaches to Real Time Level Control. *International Journal of Intelligent Systems and Applications*, 7(10). <https://doi.org/10.5815/ijisa.2015.10.03>