Tersedia online di: http://ejournal.undip.ac.id/index.php/teknik

TEKNIK, 43 (1), 2022, 36-44

Analisis Performansi Penerapan *State-Estimator* pada *Hardware-In-The-Loop* Sistem *Ball and Beam*

Muhammad Zakiyullah Romdlony[1*], Fakih Irsyadi[2], Dien Rahmawati[1], Handika Y. Kristiawan[1]

[1]*Fakultas Teknik Elektro, Universitas Telkom*
Jl. Telekomunikasi 1, Bandung, Indonesia 40257
[2]*Sekolah Vokasi, Universitas Gadjah Mada,*
Catur Tunggal, Sleman, Yogyakarta, Indonesia 55281

Abstrak

Ball and Beam System (BBS) merupakan model kendalian yang seringkali digunakan untuk memodelkan berbagai sistem dinamis yang kompleks dan unstable. BBS merupakan underactuated system dimana sistem kompleks untuk dikendalikan sehingga ideal untuk penerapan berbagai jenis kendali, mulai dari kendali klasik, modern maupun cerdas. Mayoritas perancangan sistem kendali dilakukan secara simulasi. Metode ini kurang realistis karena dilakukan pada kondisi yang ideal sehingga hasilnya tidak dapat diimplementasikan secara langsung pada sistem riil. Salah satu metode lain yang dapat digunakan adalah simulasi Hardware in the loop (HIL). Penggunaan perangakat pengendali memungkinkan hasil perancangan dapat secara langsung digunakan untuk mengendalikan sistem riil. Paper ini mengusulkan perancangan kendali full state feedback untuk stabilisasi BBS menggunakan setup HIL simulator. Pengembangan yang dilakukan adalah penambahan state estimator, yang ditanamkan pada perangkat pengendali, untuk mengestimasi nilai posisi dan kecepatan bola sebagai masukan pengendali. Hasil pengujian menunjukkan bahwa perancangan state estimator pada simulasi HIL berhasil dilakukan. State estimator dapat mengestimasi output posisi BBS dengan waktu konvergensi yang cepat, sekitar 1,32 detik. Performansi yang dihasilkan sistem serupa dengan simulasi maupun implementasi sistem riil. Hal ini menunjukkan bahwa sistem yang diusulkan dapat merepresentasikan dinamika sistem pada full state feedback control.

Kata kunci*: ball and beam; state estimator; kendali full state feedback; hardware in the loop; kontrol posisi*

Abstract

[Title: Performance analysis of State-Estimator Implementation on Hardware-In-The-Loop of Ball and Beam System] *Ball and beam system (BBS) is a commonly used model to represent several complex and unstable systems. BBS is underactuated system. It is an ideal model to implement kind of control theory. Most of control design are done using simulation method. This method is not realistic because it is conducted on the ideal situation which implies that the result cannot be directly used to control the real system. Hardware in the loop (HIL) simulation is a method that can be used to solve these problems. The use of real controller makes the design experience more realistic and ready to implement. This paper proposes a design of full state feedback for BBS stabilization using HIL simulator. The contribution of this work is to design state estimator on the BBS setup to estimate position and velocity. The result shows that the proposed control and estimator are successfully implemented. The estimator can estimate the output position with time convergent about 1,32 second. The performance of the system is similar with simulation and real plant implementation. It shows that this method can represent the dynamic response of the system on full state feedback control.*

Keywords*: ball and beam; state estimator; full state feedback control; hardware in the loop; position control*

*) Penulis Korespondensi.
E-mail: zakiyullah@telkomuniversity.ac.id

doi: 10.14710/teknik.v43i1.41681

1. Pendahuluan

Ball and Beam System (BBS) merupakan salah satu model kendalian yang sangat penting pada sistem kendali. Sistem ini bersifat *non-linear* dan *unstable* dimana dapat digunakan untuk merepresentasikan berbagai sistem dinamis yang kompleks dan tidak stabil, seperti pada *segway*, misil militer, dan stabilisasi posisi horizontal pesawat terbang saat lepas landas, pendaratan dan turbulensi (Kim dkk., 2021) BBS tersusun dari sebuah motor yang terhubung pada suatu batang (*beam*) serta sebuah bola yang dapat secara bebas menggelinding di permukaan *beam*. Pengaturan posisi bola dapat dilakukan dengan mengatur posisi sudut motor yang akan menghasilkan simpangan sudut horizontal pada *beam*. Besarnya sudut simpangan dipengaruhi oleh *error* posisi bola yang diukur dari perbandingan antara referensi posisi yang diberikan dan posisi bola yang diukur melalui sensor. Tujuan pengendalian sistem ini adalah untuk dapat mempertahankan bola pada posisi tertentu diatas permukaan beam.

BBS merupakan sistem *underactuated* yang merupakan sistem dengan dua derajat kebebasan (posisi bola dan sudut beam) dan satu derajat aktuasi (Ali, 2020). Hal tersebut menyebabkan pengendalian sistem lebih kompleks dibandingkan dengan *fully-actuated* system (Chang dkk., 2013). Beberapa hal tersebut menjadikan BBS merupakan prototipe sistem yang ideal untuk penerapan berbagai strategi kontrol (Anjali & Mathew, 2017). Terdapat beberapa penelitian yang telah dilakukan terkait penerapan beberapa strategi kontrol, baik kontrol klasik, kontrol modern maupun kontrol cerdas, pada stabilisasi BBS. Melalui penelitiannnya, Latif, Muhammad, dan Naeem (2019) melakukan penerapan dua kendali secara terpisah, yaitu 2 DOF PID dan *Fuzzy logic*, dan membandingkan performansi dari keduanya pada stabilisasi sistem *ball and beam*. Saad dan Khalfallah (2017) melakukan penerapan kendali *Fuzzy PID*, yang merupakan penggabungan antara kendali *Fuzzy* dan PID, yang berhasil meningkatkan performansi kendali dalam menstabilkan BBM dibandingkan dengan kontrol PID konvensional. Anjali dan Mathew (2017) mengimplementasikan kontrol optimal, yaitu dengan menggabungkan dua kendali PID secara *cascade* untuk mengendalikan posisi bola dan posisi sudut motor servo, pada BBS. Hasil pengujian pada sistem riil menunjukkan bahwa, penerapan nilai parameter PID yang didapat dari proses *tuning* menggunakan algoritma genetika, output sistem dapat mengikuti perubahan referensi posisi yang diberikan. Pada penelitian Liqing dan Liu (2016) diterapkan kendali *Back Propagation* (BP) *Neural Network* yang menghasilkan respon stabilisasi lebih cepat dibandingkan dengan kendali *root locus*.

Mayoritas penelitian pada BBS dilakukan secara simulasi dan/atau implementasi pada sistem riil untuk memverifikasi hasil perancangan kendali. Kondisi yang serba ideal menjadikan perancangan sistem kendali dengan simulasi menjadi kurang realistis. Perlu dilakukan beberapa penyesuaian agar hasil perancangannya dapat diterapkan pada sistem riil secara langsung. *Hardware-in-the-Loop* (HIL) dapat dijadikan sebagai salah satu alternatif solusi untuk melakukan pengembangan dan pengujian suatu sistem kendali secara lebih realistis.

HIL merupakan salah satu teknik simulasi *embedded real-time* yang bertujuan untuk menguji suatu sistem sebelum mengimplementasikannya ke sistem riil (Salazar dkk., 2016). Simulasi HIL menggunakan *virtual plan/process* yang dapat digunakan untuk menguji sistem kendali yang telah dikembangkan pada berbagai sistem dinamik dan proses riil yang telah dimodelkan. Hingga saat ini, penelitian mengenai perancangan simulasi HIL masih terus dilakukan. Penerapan simulasi HIL digunakan pada berbagai bidang seperti pada sistem energi (Gambier, 2020), pada sistem otomotif (Soltani & Assadian, 2016) maupun agrikultur (Raikwar, 2019).

Pada paper ini dibahas perancangan kendali *full state feedback* pada BBS dengan menggunakan setup HIL simulator yang telah dikembangkan sebelumnya pada penelitian (Tumanggor dkk., 2019; Romdlony & Irsyadi, 2021). Pengembangan dilakukan dengan menambahkan *state-estimator*. Perancangan state-estimator dalam kendali *full state feedback* seringkali dilakukan untuk mengestimasi variabel-variabel pengontrolan yang tidak selalu tersedia (Siradjuddin dkk., 2018).

Pada sistem yang diusulkan, *state estimator* digunakan untuk mengestimasi posisi dan kecepatan bola berdasarkan sinyal *output* sistem yang selanjutnya digunakan sebagai umpan balik pada pengendalian *virtual plant* (BBS) yang dijalankan pada *software* simulasi (Simulink) menggunakan komputer. Algoritma pengendalian dan state-estimator akan ditanam pada perangkat pengendali (Arduino). Proses pertukaran data antara komputer dan perangkat pengendali dilakukan menggunakan DAQ (NI-usb 6008).

2. Bahan dan Metode

2.1 Pemodelan Sistem *Ball and Beam*

Sistem *ball and beam* merupakan suatu alat peraga sistem kesetimbangan sederhana yang terdiri dari sebuah batang panjang dimana salah satu ujungnya terhubung dengan konfigurasi motor dan *gearbox*. Kemiringan batang dapat diatur dengan merubah posisi motor untuk membuat bola, yang bergerak bebas dipermukaan batang, berada pada posisi yang diinginkan. Gambar 1 merupakan ilustrasi sederhana sistem *ball and beam.*Dimana m merupakan massa bola (kg), M adalah massa batang (kg), d adalah panjang tangan *offset* (m), g adalah percepatan gravitasi (m/s^2), J_b adalah momen inersia bola ($kg.m^2$), L adalah panjang batang (m), r adalah jarak bola dari titik nol (m), β

doi: 10.14710/teknik.v43i1.41681

adalah sudut batang (°), θ adalah sudut motor servo (°) dan R adalah jari-jari bola (m).

Penurunan persamaan matematis sistem *ball and beam* mengacu pada pendekatan persamaan *lagrange*. Simplifikasi persamaan *lagrangian* pada gerak bola dituliskan pada Persamaan 1 (Maalini, dkk., 2017).

$$\left(\frac{j_b}{R^2}+m\right)\ddot{r}+mg\sin\alpha-mr\dot{\alpha}^2=0 \quad (1)$$

Selanjutnya dilakukan linierisasi persamaan diatas, dengan menganggap $\sin\alpha \approx \alpha$ (untuk nilai $\alpha \approx 0$) dan hubungan antara posisi sudut motor dengan sudut *beam* dinyatakan pada Persamaan 2 maka fungsi transfer, hubungan antara jarak (posisi) bola terhadap posisi sudut motor, pada BBM dapat ditulisakan seperti pada Persamaan 3.

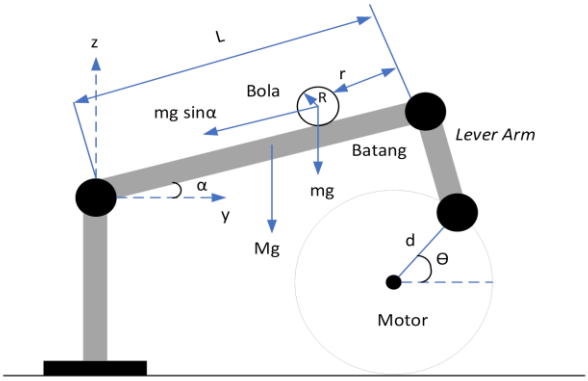
$$\alpha=\frac{d}{l}\theta \quad (2)$$

$$\frac{\mathrm{r(s)}}{\theta(\mathrm{s})}=\frac{mgd}{L\left(\frac{j_b}{R^2}+m\right)}\frac{1}{s^2} \quad (3)$$

Dengan asumsi bola yang digunakan untuk percobaan adalah bola berongga, $u(t)=\theta(t)$, $x_1(t)=r(t)$, $x_2(t)=\dot{r}(t)$ dan $y(t)=r(t)$, maka didapat matriks A, B, C dan D pada bentuk *state space* yang ditampilkan pada Persamaan 4.

$$A=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix};\ B=\begin{bmatrix}0\\ -\frac{3gd}{5L}\end{bmatrix};$$
$$C=\begin{bmatrix}1 & 0\end{bmatrix};\ D=0 \quad (4)$$

Dengan spesifikasi mekanik *plant* yang digunakan pada penelitian ini adalah $g=9{,}676\ m/s^2$, $d=0{,}1\ m$ dan $L=0{,}32\ m$. Maka didapat persamaan/pemodelan *state space* sistem seperti yang tertulis pada Persamaan 5.



Gambar 1. Ilustrasi sistem *ball and beam*.

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0\\ -1{,}81\end{bmatrix}u(t)$$
$$y=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \quad (5)$$

2.2. Perancangan Kendali Full State Feedback

Kendali *full-state feedback* bertujuan untuk mencari nilai penguatan (*gain) state feedback* (K) yang dapat menggeser lokasi *closed loop poles* untuk mencapai performansi yang diinginkan. Skema kendali *full-state feedback* dapat dilihat dari Gambar 2. Sebelum dilakukan perancangan, dilakukan uji keterkendalian pada sistem yang bertujuan untuk mengetahui apakah sistem dapat dikendalikan melalui sinyal kendali. Pengujian dilakukan dengan cara membuktikan jumlah orde dari sistem dan *rank* pada matriks keterkendalian sistem adalah sama.

Selanjutnya nilai penguatan state K dapat dicari dengan menggunakan Persamaan 6 yang juga disebut sebagai persamaan *Ackerman*.

$$K=\begin{bmatrix}0 & 1\end{bmatrix}M_C^{-1}\Phi_d(A) \quad (6)$$

Dimana M_c merupakan matriks keterkendalian dan Φ_d merupakan persamaan karakteristik *closed-loop* yang diinginkan.

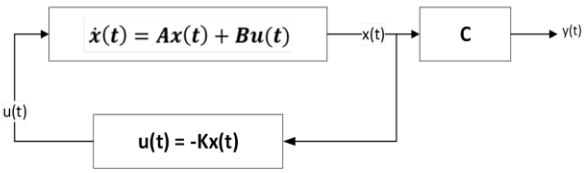
Pada penelitian ini, dipilih spesifikasi sistem yang diharapkan adalah sistem bersifat *underdamped* dengan maksimum *overshoot* 10% dan *settling time* (2%) 3 detik. Sehingga didapatkan nilai konstanta *gain feedback* K, seperti terlihat pada Persamaan 7.

$$K=\begin{bmatrix}-2{,}79 & -1{,}46\end{bmatrix} \quad (7)$$

2.3. Perancangan State Estimator

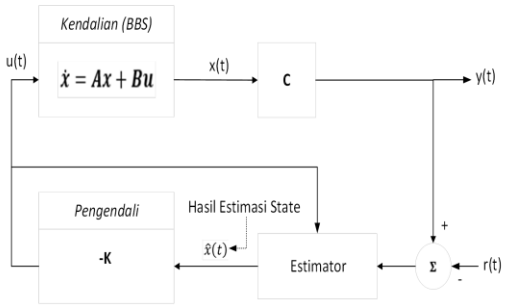
Hal terpenting sebelum melakukan perancangan *state estimator* adalah *observability* (keteramatan) suatu sistem. Syarat *observability* akan terpenuhi jika matriks *observability* (M_O) *linearly independent* atau rank matriks observability sama dengan jumlah orde sistem.

Gambar 3 menunjukkan representasi sederhana sistem kendali full state feedback dengan menggunakan *state estimator*. *State estimator* akan menghasilkan *estimated state* yang menyerupai *real state* apabila keduanya mempunyai kondisi *state* awal yang sama.

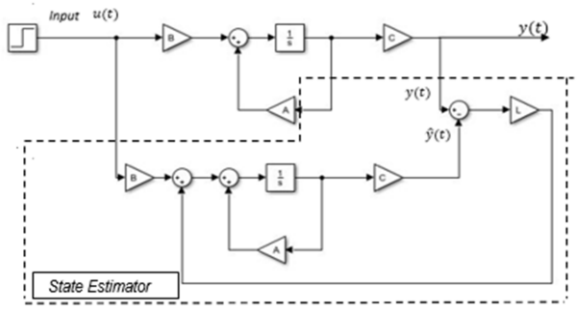


Gambar 2. Konfigurasi kendali *full-state feedback*

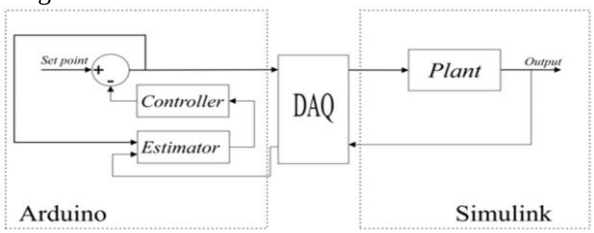
doi: 10.14710/teknik.v43i1.41681



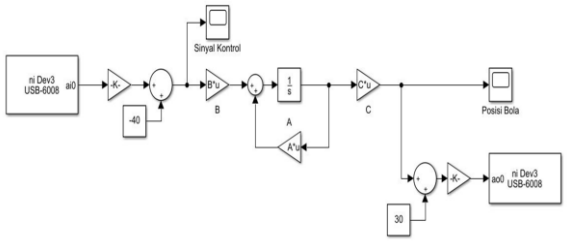
Gambar 3. Diagram blok sistem kendali *full-state feedback* dengan *state estimator*.



Gambar 4. Representasi *closed-loop state estimator*



Gambar 5. Diagram blok rancangan simulasi HIL



Gambar 6. Representasi *virtual plant*

Penentuan posisi awal *state* yang buruk dapat mengakibatkan *error* pada *estimate state* terus membesar atau mengecil dengan sangat lambat. Hal ini menyebabkan *error* pada *output system*.

Salah satu metode yang dapat mengatasi *error* dalam mendesain *state estimator* adalah menerapkan *closed-loop state estimator* yang memanfaatkan perbedaan *measured output* dengan *estimated output* untuk memperbaiki model *estimator* secara kontinyu. Gambar 4 merupakan representasi sistem dengan *closed-loop state estimator*. Persamaan *estimated state* ditunjukkan pada Persamaan 8.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - C\hat{x}(t)] \qquad (8)$$

Dimana L adalah penguatan proporsional yang digunakan untuk mengatur karakteristik sinyal *error*. Penguatan L dapat dicari dengan menggunakan persamaan *Ackermann*. Nilai L dapat dicari dengan Persamaan 9.

$$L = \Phi_d(A)(M_o)^{-1}\begin{bmatrix}0\\1\end{bmatrix} \qquad (9)$$

Dimana M_O adalah matriks keteramatan dan Φ_d persamaan karakteristik sistem.

Perancangan *state estimator* dilakukan berdasarkan spesifikasi performansi sistem yang diharapkan. Pada perancangan kendali di bagian sebelumnya, didapatkan persamaan karakteristik sistem yang diharapkan direpresentasikan pada Persamaan 10.

$$\Phi_d(s) = s^2 + 2,667s + 5,087 = 0 \qquad (10)$$

Sehingga kutub sistem yang diharapkan berada pada $s_{1,2} = -1,33 \pm j1,819$. Secara teori, kutub *estimator* dipilih 2 sampai 6 kali lebih cepat daripada kutub pada perancangan pengendali (Hauser, Sastry, & Kokotovic, 1992).

Hal ini bertujuan untuk memastikan proses peluruhan *error* pada estimator lebih cepat dibandingkan dengan dinamika respon sistem. Pada penelitian ini, kutub estimator yang dipilih adalah $s_{1,2} = -6,665 \pm j1,819$ atau 5 kali lebih besar dibandingkan dengan kutub pada perancangan kendali. Sehingga didapatkan konstanta penguatan estimator L seperti yang ditunjukkan pada Persamaan 11.

$$L = \begin{bmatrix}13,33\\47,73\end{bmatrix} \qquad (11)$$

2.4. Perancangan Simulasi HIL

HIL terdiri dari komponen pengendali ECU (*Electric Control Unit*) yang digunakan untuk mengendalikan *virtual plant* yang dijalankan menggunakan *software* simulasi pada komputer. *Virtual plant* merupakan model matematika kendalian yang dapat merepresentasikan dinamika sistem layaknya sistem riil (Romdlony & Irsyadi, 2021).

doi: 10.14710/teknik.v43i1.41681

2.4.1. Diagram Blok *Hardware* pada Simulasi HIL

Diagram blok dari skema simulasi HIL yang akan diterapkan pada penelitian ini ditunjukkan pada Gambar 5. Pada sistem ini, perangkat DAQ (NI USB-6008) berfungsi sebagai media komunikasi antara ECU (mikrokontroler Arduino) dengan *software* simulasi (Simulink) pada proses pengendalian. Modul DAQ membaca data variabel kendali dari *virtual plant* yang dikirimkan melalui komunikasi serial dan merubahnya menjadi representasi sinyal analog yang dapat dibaca, melalui pin ADC, oleh ECU sebagai sinyal umpan balik (*feedback*). Modul DAQ juga mengkonversi sinyal kendali (analog) hasil dari perhitungan kendali menjadi data yang nantinya dikirimkan melalui komunikasi serial ke *virtual plant* sebagai sinyal kendali sebagai *input* aktuator. Gambar 6 menunjukkan pemodelan *virtual plant* yang digunakan untuk simulasi HIL pada penelitian ini.

2.4.2. Diagram Alir Simulasi HIL

Bagian ini akan membahas mekanisme cara kerja simulasi HIL pada sistem *ball and beam* yang telah dirancang. Diagram alir mekanisme cara kerja simulasi HIL ditunjukkan pada Gambar 7.

Proses pengendalian *virtual plant* dan estimasi *state* dilakukan oleh Arduino dengan mengeksekusi algoritma yang telah ditanamkan dengan menggunakan *compiler* Arduino IDE. Pengendali membaca sinyal umpan balik dari *virtual plant*, manjalankan algoritma kendali dan mengirimkan sinyal kendali untuk men-*drive virtual plant* yang dijalankan pada *software* Simulink. Seluruh proses pertukaran data antara Arduino dengan *virtual plant* dilakukan melalui modul DAQ.

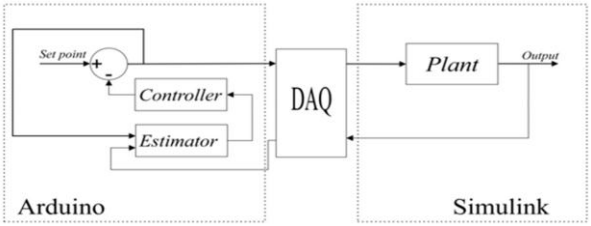
3. Hasil dan Pembahasan

Analisis dilakukan dengan cara membandingkan performansi kendali pada simulasi *software*, simulasi HIL dan implementasi pada *real plant*. Pengujian akan menerapkan sistem regulator sehingga *input* referensi akan selalu bernilai 0 serta pengujian akan dilakukan dengan posisi awal yang bervariasi. Pada pengujian sistem, referensi titik nol diatur pada titik tengah *beam* sehingga dinamika posisi bola bernilai antara -16 cm hingga 16 cm.

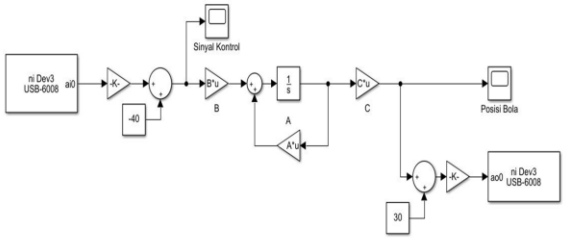
3.1. Pengujian I: Simulasi *Software* (Simulink)

Pengujian simulasi sistem pada *software* Simulink digunakan untuk melihat performansi ideal dari hasil perancangan. Gambar 8 merupakan pemodelan sistem *closed-loop ball and beam* pada Simulink.

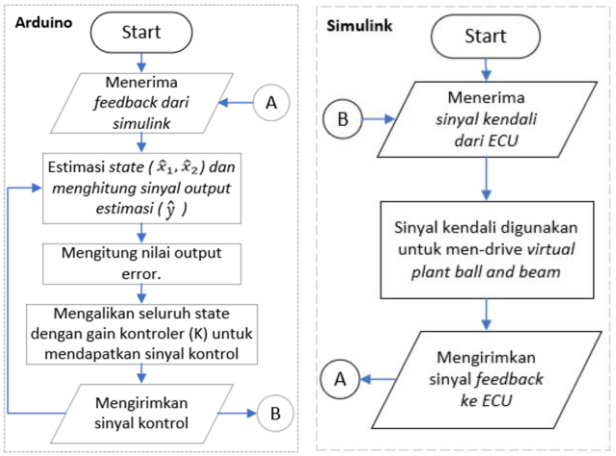
Pada pengujian ini, posisi bola akan dihubungkan dengan *output state estimator* sehingga seluruh *state* dapat diestimasi dan akan diumpan balikkan untuk menghitung sinyal kendali. Gambar 9 merupakan representasi grafik respon sistem pada simulasi dengan simpangan posisi awal 11 cm.



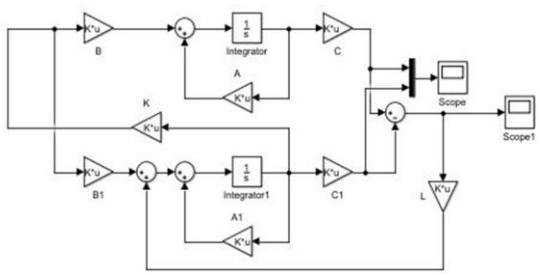
Gambar 5. Diagram blok rancangan simulasi HIL



Gambar 6. Representasi *virtual plant*

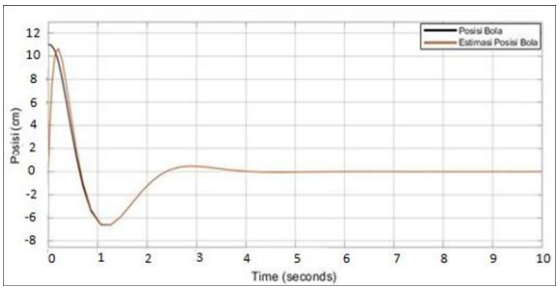


Gambar 7. Flowchart *cara kerja sistem.*

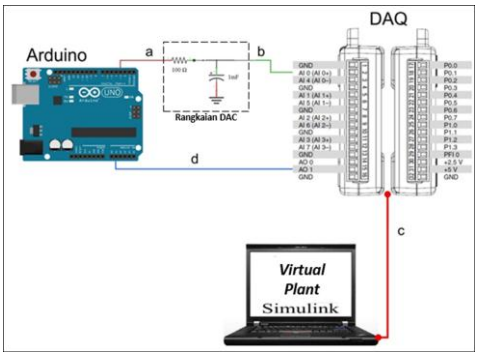


Gambar 8. Representasi *closed-loop BBS*

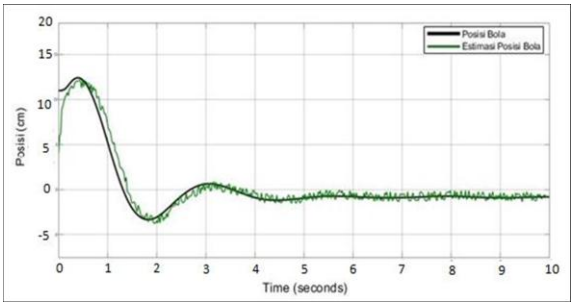
doi: 10.14710/teknik.v43i1.41681



Gambar 9. Grafik respon sistem hasil pengujian simulasi



Gambar 10. *Setup* simulasi HIL



Gambar 11. Grafik respons sistem pada simulasi HIL

Pada pengujian ini, posisi awal pada *state-estimator* diatur 0 cm. Didapat nilai *overshoot* 43,21% dan *settling time* 4,12 detik. Pemilihan kutub pada *state estimator* dipercepat 5 kali dari kutub pengendali berhasil membuat sistem memperbaiki *error* posisi estimasi dengan cepat, dengan waktu rata-rata sekitar 1,12 detik, tanpa mempengaruhi *dynamics respons* dari sistem. Tabel 1 menunjukkan hasil simulasi *closed loop sistem* pada beberapa posisi awal yang berbeda.

3.2. Pengujian II: Simulasi HIL

Pada pengujian ini, *virtual plant* mengirimkan sinyal *output* berupa posisi bola ke Arduino sebagai masukan *estimator*. Selanjutnya *output estimator* (nilai estimasi posisi dan kecepatan) dikirimkan ke pengendali untuk menghasilkan sinyal kendali. Sinyal kendali dikirimkan ke *virtual plant* untuk proses pengendalian. Seluruh proses estimasi dan perhitungan sinyal kendali dilakukan pada Arduino. Pengujian dilakukan dengan kondisi awal pada *virtual plant* yaitu 11 cm dan kondisi awal posisi estimasi diatur 0 cm. Gambar 10 menunjukkan *setup* simulasi HIL yang digunakan pada pengujian serta Gambar 11 menunjukkan grafik representasi respon sistem pada pengujian simulasi HIL.

Gambar 11 memperlihatkan bahwa adanya *noise* pada sinyal hasil estimasi posisi. Meskipun begitu, *trend* dari sinyal estimasi posisi berhasil mengikuti *trend* dari sinyal posisi *virtual plant* dengan spesifikasi maksimum *overshoot* 28,16% dan *settling time* 4,771 detik. Lama waktu peluruhan *error* posisi estimasi rata-rata sekitar 1,32 detik. Tabel 2 menunjukkan hasil simulasi HIL pada beberapa posisi awal yang berbeda.

3.3. Pengujian III: Implementasi pada *Real Plant*

Pengujian berikutnya adalah implementasi kendali pada *real plant* yang telah dirancang sebelumnya. Gambar 12 menunjukkan *real plant* yang digunakan dalam pengujian ini. Realisasi kendalian menggunakan motor servo untuk mempertahankan posisi bola dengan mengatur kemiringan beam. Posisi bola diukur berdasarkan jarak dari salah satu ujung batang dengan menggunakan sensor ultrasonik. Kontroler menggunakan Arduino UNO sedangkan *plant* berupa sistem mekanik ball and beam yang telah dirangkai dengan komponen elektronik penunjang proses pengendalian. Pengujian *closed loop* pada *real plant* dilakukan dengan bola pingpong yang merupakan representasi dari bola berongga. Pengujian akan dilakukan dengan posisi awal yang berbeda-beda dan nilai awal pada state estimator selalu 0 cm.

Gambar 13 menunjukkan hasil pengujian kendali pada *real plant*. Grafik hasil pengujian tersebut menunjukkan bahwa respon sistem memiliki nilai *overshoot* maksimum 14,27%, *settling time* 5,6 s dan *steady state error* 0,814 cm. Pada pengujian *real plant*, *error* sinyal estimasi posisi dapat diperbaiki rata-rata pada detik ke 1,25. Proses peluruhan nilai *error* estimasi posisi tidak mengganggu dinamika respon sistem. Tabel 3 merupakan hasil pengujian pada implementasi *real plant* untuk beberapa posisi awal yang berbeda.

doi: 10.14710/teknik.v43i1.41681

Tabel 1. Hasil percobaan simulasi pada beberapa nilai simpangan

Set Point (cm)	M_P (%)	e_{ss} (cm)	t_s (detik)	Waktu konvergensi error posisi (detik)
11	43.21	0.04	4,12	1,12
8	40.44	0.03	4,09	1,12
-8	40.44	-0.03	4,09	1,12
-14	45,84	-0.05	4,14	1,12

Tabel 2. Hasil percobaan simulasi HIL pada beberapa nilai simpangan

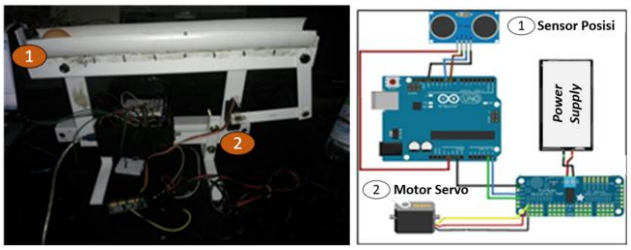
Set Point (cm)	M_P (%)	e_{ss} (cm)	t_s (detik)	Waktu konvergensi error posisi (detik)
11	28,16	-0,87	4,77	1,20
8	39,15	-0,71	5,79	1,25
-8	52,33	0,87	4,92	1,25
-14	37,88	-0,82	6,22	1,60

Tabel 3. Hasil percobaan implementasi *real plant* pada beberapa nilai simpangan

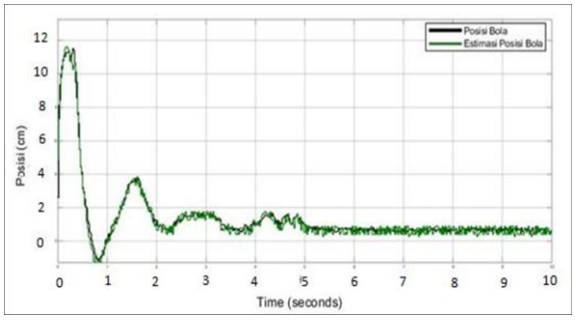
Set Point (cm)	M_P (%)	e_{ss} (cm)	t_s (detik)	Waktu konvergensi error posisi (detik)
11	14,27	0,58	5,2	1,02
8	0	0,23	5,01	1,22
-8	14,61	0,12	1,92	1,05
-14	73,24	0,76	6,74	1,72

Tabel 4. Perbandingan rataan hasil pengujian

Parameter	Simulasi	HIL	*Real Plant*
M_P (%)	42,48	39,15	25,53
e_{ss} (cm)	0,0375	0,866	0,423
T_s (detik)	4,11	5,426	4,7175
Waktu Konvergensi error (detik)	1,12	1,32	1,25



Gambar 12. Realisasi mekanika BBS dan skematik elektronika



Gambar 13. Grafik respons sistem pada implemenatasi *real plant*

3.4. Pembahasan

Tabel 4 menunjukkan perbandingan hasil rataan performansi sistem melalui simulasi, HIL dan *real plant* ketika diterapkan metode dan parameter kendali yang sama. Dari perbandingan rataan hasil pengujian pada Tabel 4, terlihat bahwa *overshoot* maksimum pada pengujian *real plant* paling rendah dengan selisih 13.62% dan 16,95% terhadap hasil simulasi dan *real plant.* Selisih nilai *overshoot* maksimum diduga disebabkan oleh kendala mekanik sistem dalam merespon sinyal masukan. Adanya friksi dan non-linearitas motor penggerak berpotensi meredam *overshoot* pada sistem dimana hal ini tidak dimodelkan pada skema simulasi dan HIL dikarenakan *plant* pada kedua skema tersebut berupa *virtual plant*.

Nilai *error steady state* pada simulasi HIL dan implementasi *real plant* lebih besar dibandingkan dengan hasil pengujian Simulasi. Pada simulasi HIL, nilai *error steady state* 0,8385 lebih besar dari hasil pengujian simulasi.

Settling time dan waktu peluruhan *error* sinyal estimasi posisi pada simulasi HIL memiliki waktu yang paling lama. Hal ini diduga karenakan terdapat *delay* akibat komunikasi data antara arduino dengan *virtual plant,* pada pengiriman sinyal kontrol dan umpan balik.

doi: 10.14710/teknik.v43i1.41681

4. Kesimpulan

Pada paper ini telah dilakukan penerapan *state-estimator* untuk mengestimasi posisi dan kecepatan bola pada simulasi HIL BBS. Hasil pengujian menunjukkan bahwa estimator yang dirancangan berhasil mengestimasi nilai posisi bola dengan baik. Profil data hasil estimasi posisi yang dihasilkan oleh estimator berhasil mereplikasi sinyal output BBS. Waktu konvergensi sinyal estimator sangat cepat, sekitar 1,32 detik, sehingga kesalahan estimasi tidak menggaggu proses pengendalian ketika nilai awal estimasi berbeda.

Performansi kendali pada simulasi HIL telah dikomparasi dengan performansi pada simulasi dan implementasi *real plant*. Ketiganya menghasilkan performansi yang tidak jauh berbeda untuk parameter pengendalian yang sama. Hasil ini menunjukkan sitem yang diusulkan dapat merepresentasikan *dynamic response* dari BBS pada pengendalian *full state feedback*.

Daftar Pustaka

Ali, S. S. (2020). Position control of ball and beam system using robust h∞ loop shaping controller. *Indonesian Journal of Electrical Engineering and Computer Science*, 19, 91. https://doi.org/10.11591/ijeecs.v19.i1.pp91-98

Anjali, T., & Mathew, S. S. (2017). Implementation of optimal control for ball and beam system. *Proceedings of IEEE International Conference on Emerging Technological Trends in Computing, Communications and Electrical Engineering, ICETT 2016*, 3–7. https://doi.org/10.1109/ICETT.2016.7873763

Chang, Y.-H., Chan, W.-S., & Chang, C.-W. (2013). T-S Fuzzy Model-Based Adaptive Dynamic Surface Control for Ball and Beam System. IEEE Transactions on Industrial Electronics, 60(6), 2251–2263. https://doi.org/10.1109/TIE.2012.2192891

Gambier, A. (2020). Real-time Control and Hardware-in-the-loop Simulation for Educational Purposes of Wind Energy Systems. IFAC-PapersOnLine, 53(2), 17344–17349. https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.2084.

Liqing, G. & Liu, Y. (2016). Design of BP neural network controller for ball-beam system. Proceeding of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 1087-1091. https://doi.org/10.1109/IMCEC.2016.7867379.

Hauser, J., Sastry, S., & Kokotovic, P. (1992). Nonlinear control via approximate input-output linearization: the ball and beam example. IEEE Transactions on Automatic Control, 37(3), 392–398. https://doi.org/10.1109/9.119645

Kim, Y., Kim, S. K., & Ahn, C. K. (2021). Variable Cut-Off Frequency Observer-Based Positioning for Ball-Beam Systems without Velocity and Current Feedback Considering Actuator Dynamics. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *68*(1), 396–405. https://doi.org/10.1109/TCSI.2020.3032128

Latif, S., Muhammad, E., & Naeem, U. (2019). Implementation of ball and beam system using classical and advanced control techniques. *2019 International Conference on Applied and Engineering Mathematics, ICAEM 2019 - Proceedings*, 74–79. https://doi.org/10.1109/ICAEM.2019.8853822

Maalini, P. V. M., Prabhakar, G., & Selvaperumal, S. (2017). Modelling and control of ball and beam system using PID controller. *Proceedings of 2016 International Conference on Advanced Communication Control and Computing Technologies, ICACCCT 2016*, (978), 322–326. https://doi.org/10.1109/ICACCCT.2016.7831655

Raikwar, S., Jijyabhau Wani, L., Arun Kumar, S., & Sreenivasulu Rao, M. (2019). Hardware-in-the-Loop test automation of embedded systems for agricultural tractors. *Measurement: Journal of the International Measurement Confederation*, *133*, 271–280. https://doi.org/10.1016/j.measurement.2018.10.014

Romdlony, M. Z., & Irsyadi, F. (2021). Journal of Mechatronics, Electrical Power, Hardware-in-the-loop simulation of DC motor as an instructional media for control system design and testing. 12, 81–86. https://dx.doi.org/10.14203/j.mev.2021.v12.81-86

Saad, M., & Khalfallah, M. (2017). Design and Implementation of an Embedded Ball-beam Controller Using PID Algorithm. *Universal Journal of Control and Automation*, *5*(4), 63–70. https://doi.org/10.13189/ujca.2017.050402

Salazar, G. D. B., Hurtado, D. A., & Sandoval, O. R. (2016). Hardware-in-the-loop simulation and digital control of double inverted pendulum. *ARPN Journal of Engineering and Applied Sciences*, *11*(2), 940–944.

Siradjuddin, I., Amalia, Z., Rohadi, E., Setiawan, B., Setiawan, A., Ika Putri, R., & Yudaningtyas, E. (2018). State-feedback control with a full-state estimator for a cart-inverted pendulum system. *International Journal of Engineering &*

Technology, *7*(4.44), 203. https://doi.org/10.14419/ijet.v7i4.44.26985

Soltani, A., & Assadian, F. (2016). A Hardware-in-the-Loop Facility for Integrated Vehicle Dynamics Control System Design and Validation. *IFAC-PapersOnLine*, *49*(21), 32–38. https://doi.org/10.1016/j.ifacol.2016.10.507

Tumanggor, L. C. E. Romdlony, M. Z. and Irsyadi, F. (2019). Rapid Prototyping Sistem Kendali PI Anti-Windup Menggunakan Simulator HIL (Hardware-In-The-Loop), *Proceeding of 11th Conference on Information Technology and Electrical Engineering (CITEE 2019)*, pp. 24–25.

doi: 10.14710/teknik.v43i1.41681