# Kontrol LQR-PID Hibrida untuk Robot Penyeimbang Otomatis: Simulasi dan Optimalisasi Dinamika Menggunakan Matlab Simscape


Jujur Satria Yudhatama
Teknik Mesin
Institut Teknologi Bandung
Bandung, Indonesia
13122114

Russell Christopher Yudhistira
Teknik Fisika
Institut Teknologi Bandung
Bandung, Indonesia
13322037

Faaiq Mastyaga
Teknik Fisika
Institut Teknologi Bandung
Bandung, Indonesia
13322052

Fathy Abdurrahman
Teknik Fisika
Institut Teknologi Bandung
Bandung, Indonesia
13322054

Valentino
Teknik Fisika
Institut Teknologi Bandung
Bandung, Indonesia
13322076

Zulfikar Firmanto
Teknik Fisika
Institut Teknologi Bandung
Bandung, Indonesia
13322098



***Abstract*—Robot penyeimbang otomatis (*self-balancing robot*) merupakan sistem yang secara intuitif tidak stabil sehingga memerlukan sistem kontrol yang efektif untuk menjaga keseimbangannya. Eksperimen ini bertujuan untuk merancang, memodelkan, dan mensimulasikan sistem kontrol untuk robot penyeimbang dua roda. Model dinamis robot diturunkan menggunakan hukum Newton dan dimodelkan dalam bentuk persamaan ruang keadaan (state-space) yang mencakup dinamika longitudinal (gerak maju dan sudut pitch) dan lateral (sudut yaw). Karena adanya keterkaitan (coupling) yang kuat antar dinamika, strategi kontrol dekomposisi diterapkan. Sistem kontrol longitudinal menggunakan kombinasi Linear-Quadratic Regulator (LQR) untuk stabilisasi optimal dan Proportional-Integral-Derivative (PID) untuk pelacakan referensi kecepatan, sementara sistem kontrol lateral menggunakan kontroler PID terpisah untuk mengatur sudut yaw. Verifikasi sistem dilakukan melalui simulasi dalam lingkungan terintegrasi MATLAB, menggunakan Simscape untuk model fisik robot dan Simulink untuk implementasi kontroler. Hasil simulasi menunjukkan bahwa arsitektur kontrol yang diusulkan berhasil menstabilkan robot dan memungkinkan pelacakan referensi kecepatan serta arah secara efektif. Selain itu, penelitian ini juga mengevaluasi pengaruh variasi parameter LQR (matriks Q dan R) terhadap respons sistem.**



***Keywords—state space, robot penyeimbang otomatis, LQR, PID, Kontrol, Simulasi Simscape***


## I. INTRODUCTION

Robot penyeimbang otomatis menjadi topik yang menarik untuk diselidiki stabilitas kontrolnya. Robot-robot ini dikembangkan berdasarkan model dinamis dan algoritma kontrol yang kompleks, serta umumnya dimanfaatkan untuk keperluan pendidikan, transportasi, dan penelitian. Keunggulan utama dari robot ini adalah kemampuannya untuk menjaga keseimbangan pada dua roda dan melakukan belokan tanpa meningkatkan kecepatan liniernya.

Karena sifatnya yang tidak stabil secara inheren, tantangan terbesar bagi robot ini adalah menjaga keseimbangannya. Upaya stabilisasi robot ini selalu dilakukan dengan menerapkan sistem kontrol dan menggunakan sensor. Metode kontrol klasik seperti PID atau LQR, bahkan kontroler modern seperti reinforcement learning, dapat digunakan untuk menyeimbangkan robot ini. Untuk menjaga keseimbangan robot, robot harus bergerak ke arah jatuhnya. Berdasarkan pergerakan ini, reaksi torsi motor dari roda akan memberikan torsi yang berlawanan dengan arah kemiringan sasis. Dengan demikian, robot akan stabil. Robot penyeimbang otomatis yang lebih tinggi cenderung lebih mudah diseimbangkan. Hal ini dikarenakan tinggi yang lebih besar menghasilkan momen inersia yang lebih besar, dan momen inersia yang lebih besar berarti resistansi yang lebih tinggi terhadap pergerakan. Oleh karena itu, ketika sasis ingin jatuh karena resistansi yang tinggi terhadap pergerakan (rotasi di sekitar sumbu roda), hal ini akan terjadi lebih lambat. Pengontrol robot secara konsisten bertujuan untuk menjaga pusat massa robot tepat di atas rodanya, karena gaya gravitasi membuat robot tidak stabil.

Dalam penelitian ini, model dinamis matematis dari robot yang disebutkan dikembangkan, kemudian perilaku sistem disimulasikan dalam lingkungan MATLAB Simulink. Akhirnya, robot disimulasikan dalam Simscape untuk mengevaluasi keseimbangan robot, pergerakan, dan navigasi otonom.

## II. PEMODELAN STATE SPACE

Model robot dapat diubah menjadi persamaan diferensial yang dibutuhkan untuk mendapatkan persamaan ruang keadaan. Berdasarkan tujuan, ingin dilakukan pengontrolan terhadap 3 variabel utama, yaitu sudut pitch $\varphi$, sudut roll $\phi$, dan sudut yaw $\psi$ serta turunannya. Fungsi persamaan ruang keadaan dapat dikontrol dengan mengatur nilai dan kutub tegangan pada motor DC.

### *A. Model Motor DC*

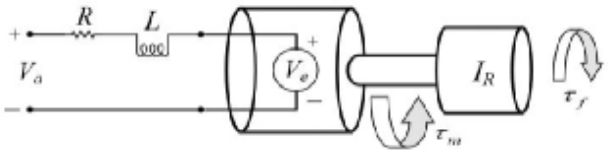



Fig. 1. Skematik Motor DC

Di dalam motor DC, terdapat loop tertutup yang terdiri dari sumber tegangan $V_a$, resistansi lilitan $R$, induktansi lilitan $L$, tegangan induksi / back EMF $V_e$. Berdasarkan Hukum Kirchoff II, didapat hubungan:

$$V_a - V_e = i_{total}R + L\frac{di}{dt} \tag{1}$$

Saat motor telah memasuki kondisi tunak, maka perubahan arus terhadap waktu akan menuju 0, sehingga persamaan (1) menjadi persamaan (2).

$$V_a - V_e = i_{total} R \tag{2}$$

Dalam perancangan motor DC, terdapat konstanta torsi motor $K_m$ dan konstanta balik EMF $K_e$ yang dapat dirumuskan sebagai persamaan (3).

$$\tau_m = K_m i_{total}, V_e = K_e \omega \tag{3}$$

Melalui metode substitusi antara persamaan (2) dan persamaan (3), didapat persamaan (4) yang menghubungkan tegangan terhadap torsi.

$$\tau_m = K_m i_{total} = K_m \frac{V_a - K_e \omega}{R} \tag{4}$$

*B. Model Roda*

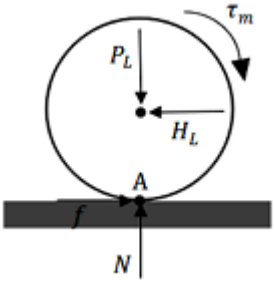



Fig. 2. Left Wheel Body Diagram

Nilai torsi pada roda memiliki hubungan dengan nilai percepatan roda. Hubungan ini dapat dijelaskan melalui Hukum Newton II.

$$\sum \tau = I\alpha$$

Berdasarkan teorema sumbu pararel, momen inersia pada roda yang menggelinding merupakan momen inersia roda terhadap pusat massa (rotasi) + momen inersia titik kontak roda terhadap tanah (translasi).

$$\tau_{m,L} - rH_L = (I_w + M_w r^2)\dot{\omega}$$

Untuk mengubah kecepatan rotasi roda, menjadi kecepatan linear, dapat menggunakan persamaan $\dot{x} = r\omega \rightarrow \frac{d\dot{x}}{dt} = \frac{dr\omega}{dt} \rightarrow \ddot{x} = r\dot{\omega}$. Subtitusikan persamaan kecepatan rotasi dan persamaan torsi ke dalam persamaan Hukum Newton II.

$$K_m \frac{V_{a,L} - K_e \frac{\dot{x}}{r}}{R} - rH_L = \frac{(I_w + M_w r^2)\ddot{x}}{r}$$

Dengan membagi kedua ruas dengan r, didapat persamaan diferensial roda kiri berupa persamaan (5) dan persamaan diferensial roda kanan berupa persamaan (6).

$$M_w \ddot{x} = \frac{K_m V_{a,L}}{Rr} - \frac{K_m K_e \dot{x}}{Rr^2} - H_L - \frac{I_w \ddot{x}}{r^2} \tag{5}$$

$$M_w \ddot{x} = \frac{K_m V_{a,R}}{Rr} - \frac{K_m K_e \dot{x}}{Rr^2} - H_R - \frac{I_w \ddot{x}}{r^2} \tag{6}$$

Jumlahkan persamaan (5) dan persamaan (6) sebagai persamaan (7).

$$2\left(M_w + \frac{I_w}{r^2}\right)\ddot{x} = \frac{K_m(V_{a,R} + V_{a,L})}{Rr} - \frac{2K_m K_e \dot{x}}{Rr^2} - H_R - H_L \tag{7}$$

*C. Model rangka Robot*

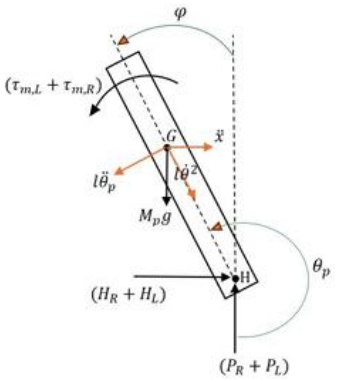



Fig. 3. Robot's Chassic Free Body Diagram

Akselerasi dari titik COM system (G) merupakan kombinasi dari $\ddot{x}(translasi)$, $l\dot{\theta_p}$ (sentrifugal), $l\ddot{\theta}_p$ (tangensial)

Pada arah sumbu X dengan titik pusat G, diterapkan hukum Newton II sehingga mendapat persamaan (8).

$$\sum F = M_p a$$

$$H_R + H_L = M_p\left(\ddot{x} + l\dot{\theta_p}\sin(\varphi) - l\ddot{\theta_p}\cos(\varphi)\right) \tag{8}$$

Subtitusikan gaya horizontal persaman (8) dengan persamaan (7) sebagai persamaan (9).

$$\frac{K_m}{Rr}\left(V_{a,L} + V_{a,R}\right) = \left(2M_w + \frac{2I_w}{r^2} + M_p\right)\ddot{x} + \frac{2K_m K_e}{Rr^2\dot{x}} - M_p l\ddot{\theta}_p \cos(\varphi) + M_p\, l\left(\dot{\theta}_p\right) sin\,(\varphi) \tag{9}$$

Misalkan sebuah variable $\beta$ dengan bentuk persamaan (10).

$$\beta = 2M_w + \frac{2I_w}{r^2} + M_p \tag{10}$$

Terapkan Hukum Newton II rotasi pada titik H yang merupakan titik rotasi.

$$\sum \tau = I\alpha$$

$$M_p\, glsin(\varphi) + \tau_{m,L} + \tau_{m,R} + lM_p\ \ddot{x}cos(\varphi) = \left(M_p\, l^2 + I_p\right)\ddot{\varphi}$$

$$\left(M_p\, l^2 + I_p\right)\ddot{\varphi} = M_p glsin(\varphi) - \frac{K_m}{R}\left(V_{a,L} + V_{a,R}\right) + \frac{2}{Rr}K_m\, K_e\dot{x} + lM_p\ddot{x}\, cos\,(\varphi) \tag{11}$$

Misalkan sebuah variable $\alpha$ dengan bentuk persamaan (12).

$$\alpha = 2M_p l^2\left(M_w + \frac{I_w}{r^2}\right) + \beta I_p \tag{12}$$

$$= \left(\beta - \frac{M_p^2\, l^2}{M_p\, l^2 + I_p}\right)\left(M_p l^2 + I_p\ \right)$$

Pada pemodelan ini, digunakan 2 asumsi bahwa nilai $\varphi$ sangat kecil dan $\dot{\theta_p}$ mendekati 0 sehingga persamaan trigonometrinya berlaku hubungan berikut.

$$\cos(\varphi) \approx 1, \sin(\varphi) \approx \varphi, dan\ \dot{\theta_p} \approx 0$$

Adanya kedua asumsi ini, persamaan (9) direduksi menjadi persamaan (13).

$$\frac{K_m}{Rr}\left(V_{a,L} + V_{a,R}\right) = (\beta)\ddot{x} + \frac{2K_m K_e}{Rr^2}\dot{x} - M_p l\ddot{\theta_p}$$

$$\ddot{\theta_p} = \frac{1}{M_p l}\left(\beta\ddot{x} + \frac{2K_m K_e}{Rr^2}\dot{x} - \frac{K_m}{Rr}\left(V_{a,L} + V_{a,R}\right)\right) \qquad (13)$$

Kemudian, asumsi ini juga digunakan untuk mereduksi persamaan (11) menjadi persamaan (14).

$$\left(M_p l^2 + I_p\right)\ddot{\varphi} = M_p g l\varphi - \frac{K_m}{R}\left(V_{a,L} + V_{a,R}\right) + \frac{2}{Rr}K_m K_e\dot{x} + lM_p\ddot{x} \qquad (14)$$

Berdasarkan diagram badan rangka robot, didapat hubungan sudut sebagai berikut.

$$\theta_p = 180 \circ + \varphi$$

$$\dot{\theta_p} = 0 + \dot{\varphi}$$

$$\ddot{\theta_p} = \ddot{\varphi}$$

Substitusikan persamaan (13) ke persamaan (14) dengan memanfaatkan hubungan sudut sehingga didapat persamaan percepatan linear robot sebagai persamaaan (15).

$$\ddot{x} = \frac{M_p^2 g l^2 \varphi}{\alpha} + \frac{2K_m K_e\dot{x}}{Rr^2\alpha}\left(M_p lr - \left(M_p l^2 + I_p\right)\right) + \frac{K_m\left(V_{a,L} + V_{a,R}\right)}{Rr\alpha}\left(M_p l^2 + I_p - M_p lr\right) \qquad (15)$$

Substitusikan persamaan (14) ke persamaan (15) untuk mendapatkan persamaan percepatan pitch robot sebagai persamaan (16).

$$\ddot{\varphi} = \frac{2K_m K_e}{\alpha Rr^2}\left(\beta r - M_p l\right)\dot{x} + \frac{M_p g l}{\alpha}\beta\varphi + \frac{K_m\left(V_{a,L} + V_{a,R}\right)}{Rr\alpha}\left(lM_p - r\beta\right) \qquad (16)$$

Berdasarkan tinjauan pustaka, didapat penurunan persamaan percepatan yaw robot di [1] sebagai persamaan final [2] (17).

$$\ddot{\psi} = \frac{K_m w}{Rr\left(I_\psi + \left(M_w + \frac{I_w}{r^2}\right)w^2\right)}\left(V_{a,L} - V_{a,R}\right) \qquad (17)$$

Setelah mendapatkan 3 persamaan diferensial variable yang ingin dikontrol, defenisikan 6 state variable sebagai berikut:

$$x_1 = x = posisi\ linear$$

$$x_2 = \dot{x} = kecepatan\ linear$$

$$x_3 = \varphi = sudut\ pitch$$

$$x_4 = \dot{\varphi} = kecepatan\ pitch$$

$$x_5 = \psi = sudut\ yaw$$

$$x_6 = \dot{\psi} = kecepatan\ yaw$$

Hasil persamaan keadaan (state space) dari robot tersebut dalam bentuk $\dot{x} = Ax + Bu$ adalah sebagai berikut.

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\\ \dot{x_3}\\ \dot{x_4}\\ \dot{x_5}\\ \dot{x_6}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & \frac{2K_m K_e}{Rr^2\alpha}\left(M_p lr - M_p l^2 - I_p\right) & \frac{M_p^2 g l^2}{\alpha} & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & \frac{2K_m K_e}{\alpha Rr^2}\left(\beta r - M_p l\right) & \frac{M_p g l}{\alpha}\beta & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix} + \begin{bmatrix}0 & 0\\ \frac{K_m}{Rr\alpha}\left(M_p l^2 + I_p - M_p lr\right) & \frac{K_m}{Rr\alpha}\left(M_p l^2 + I_p - M_p lr\right)\\ 0 & 0\\ \frac{K_m}{Rr\alpha}\left(lM_p - r\beta\right) & \frac{K_m}{Rr\alpha}\left(lM_p - r\beta\right)\\ 0 & 0\\ \frac{K_m w}{Rr\left(I_\psi + \left(M_w + \frac{I_w}{r^2}\right)w^2\right)} & \frac{-K_m w}{Rr\left(I_\psi + \left(M_w + \frac{I_w}{r^2}\right)w^2\right)}\end{bmatrix}\begin{bmatrix}V_{a,L}\\ V_{a,R}\end{bmatrix}$$

Persamaan luaran $y = Cx + Du$

$$y = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix}$$

Matriks keterkontrolan dari system ini memiliki rank = 6, sehingga system dapat dikontrol. Kemudian matriks keteramatan dari system ini memiliki rank = 5, sehingga system tidak dapat diamati.

## III. CONTROLLER DESIGN

Berdasarkan model ruang keadaan yang telah diturunkan pada bab sebelumnya, langkah selanjutnya adalah perancangan sistem kontrol. Pada tahap perancangan awal, dilakukan upaya untuk mengontrol ketiga variabel keluaran—kecepatan linear, sudut *pitch*, dan sudut *yaw*—secara simultan menggunakan metode LQR dengan integrator. Akan tetapi, hasil simulasi menunjukkan bahwa strategi ini kesulitan untuk mencapai kestabilan pada semua variabel secara bersamaan. Pengaturan parameter untuk menstabilkan satu variabel justru menyebabkan ketidakstabilan pada variabel lainnya.

Fenomena ini mengindikasikan adanya *coupling* (keterkaitan) yang kuat antara dinamika longitudinal (gerak maju dan keseimbangan *pitch*) dengan dinamika lateral (gerak berputar *yaw*). Mengontrol kedua dinamika ini dengan satu kontroler terpusat terbukti sulit dan tidak efektif.

Untuk mengatasi tantangan ini, diusulkan strategi dekomposisi kontrol, yakni sistem dibbagi menjadi dua subsistem yang dikontrol secara terpisah:

1. Subsistem longitudinal (keseimbangan dan gerak maju)

   Mengontrol kecepatan linear ($\dot{x}$) dan sudut *pitch* ($\varphi$). Pada awalnya, strategi yang dipertimbangkan adalah menggunakan LQR dengan aksi integral seperti pada gambar 4. Akan tetapi, untuk mendapatkan performa dengan *settling time* lebih cepat, aksi integral tersebut digantikan oleh kontroler PID. Dengan demikian, arsitektur ini menggabungkan keunggulan LQR dalam stabilisasi optimal dengan fleksibilitas PID dalam

membentuk pelacakan referensi kecepatan. Di sisi lain, sudut *pitch* diregulasi agar selalu kembali ke titik seimbang (nol). Arsitektur ini diperlihatkan pada gambar 5.

2. Subsistem lateral (arah gerak)

   Mengontrol sudut *yaw* ( $\psi$ ) dengan menggunakan pengontrol PID konvensional untuk melacak refrensi sudut *yaw*.

Kemudian, sinyal kontrol dari kedua kontroler ini digabungkan untuk menghasilkan tegangan aktual untuk masing-masing motor, umumnya dengan skema:

$$V_{a,L} = u_{lqr} - u_{pid}$$

$$V_{a,R} = u_{lqr} + u_{pid}$$

Dengan $u_{lqr}$ adalah keluaran dari kontroler LQR dan $u_{pid}$ adalah keluaran dari kontroler PID.

Untuk merancang kontroler LQR bagi subsistem longitudinal, model sistem direduksi dari model asli dengan 6 variabel keadaan. Variabel posisi (x) dan variabel yang berkaitan dengan *yaw* ( $\psi, \dot{\psi}$ ) diabaikan. Variabel posisi karena merupakan variabel yang tidak terukur secara langsung dan tidak esensial untuk regulasi kecepatan dan keseimbangan. Sementara itu, variabel *yaw* diabaikan karena dinamikanya ditangani oleh kontroler PID yang terpisah.

Dengan demikian, perancangan LQR hanya berfokus pada variabel kecepatan linear ($\dot{x}$), sudut *pitch* ($\varphi$), dan laju *pitch* ($\dot{\varphi}$). Persamaan ruang keadaan yang direduksi menjadi sebagai berikut.

$$\begin{bmatrix} \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{2K_m K_e}{Rr^2\alpha}(M_p lr - M_p l^2 - I_p) & \frac{M_p^2 g l^2}{\alpha} & 0 \\ 0 & 0 & 1 \\ \frac{2K_m K_e}{\alpha R r^2}(\beta r - M_p l) & \frac{M_p g l}{\alpha}\beta & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{K_m}{Rr\alpha}(M_p l^2 + I_p - M_p lr) & \frac{K_m}{Rr\alpha}(M_p l^2 + I_p - M_p lr) \\ 0 & 0 \\ \frac{K_m}{Rr\alpha}(lM_p - r\beta) & \frac{K_m}{Rr\alpha}(lM_p - r\beta) \end{bmatrix} \begin{bmatrix} V_{a,L} \\ V_{a,R} \end{bmatrix}$$

Selanjutnya, persamaan keluaran untuk perancangan aksi integral juga ditentukan. Sesuai dengan tujuan kontrol, hanya kecepatan linear ($\dot{x}$) yang akan melacak refrensi tidak nol, sedangkan sudut *pitch* ($\varphi$) hanya diregulasi menuju nol. Oleh karena itu, vektor keluaran $y$ untuk integrator didefinisikan hanya berisi variabel kecepatan linear, seperti yang ditunjukkan pada persamaan berikut.

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Meskipun sudut *pitch* tidak termasuk ke dalam vektor keluaran $y$, variabel ini tetap diregulasi secara aktif melalui mekanisme LQR dengan memberikan bobot yang sesuai pada matriks Q.

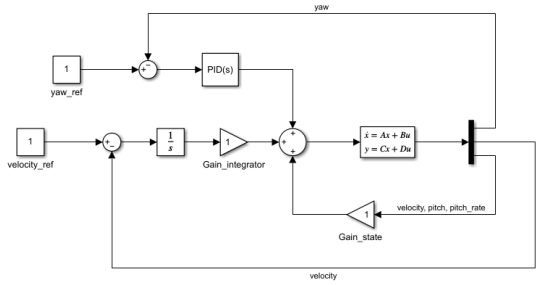



Fig. 4. Arsitektur Sistem Kontrol LQR-Integrator dan PID

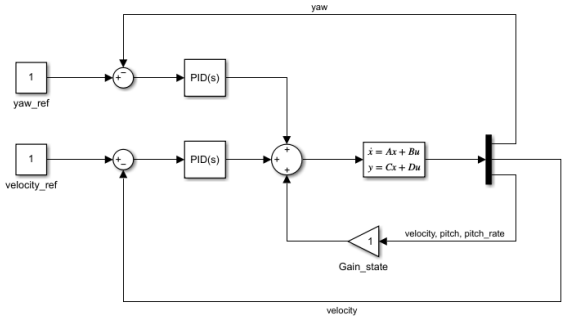



Fig. 5. Arsitektur Sistem Kontrol LQR-PID dan PID

TABEL I. menunjukkan informasi fisik terkait robot:

TABLE I. INFORMASI FISIK ROBOT

| Nama Parameter | Nilai | Satuan |
|---|---|---|
| Gravitasi konstan ($g$) | 9.81 | $m/s^2$ |
| Radius roda ($r_{wheel}$) | 0.0345 | $m$ |
| Jarak vertikal dari poros roda pusat ke pusat massa sassis ($l_{com}$) | 0.08177 | $m$ |
| Jarak antara pusat roda kiri dan kanan ($w$) | 0.251625 | $m$ |
| Massa sassis/*body* ($M_p$) | 1.2490 | $kg$ |
| Massa satu roda ($M_w$) | 0.0707 | $kg$ |
| Momen inersia sassis (*body*) terhadap sumbu roda ($I_{p\_body}$) | 0.0170 | $kg \cdot m^2$ |
| Momen inersia satu roda terhadap sumbu rotasinya ($I_w$) | 4.205e-5 | $kg \cdot m^2$ |
| Konstanta torsi motor ($K_m$) | 0.027 | $N \cdot m/A$ |
| Konstanta GGL balik motor ($K_e$) | 0.027 | $V/(rad/s)$ |
| Resistansi motor ($R_{motor}$) | 2 | $ohm$ |

## IV. ROBOT SIMULATION IN SIMSCAPE WITH MATLAB

### 4.1 Arsitektur Simulasi Keseluruhan

Simulasi Pengontrolan *self-balancing robot* ini dilakukan sepenuhnya pada lingkungan MATLAB, yakni menggunakan Simscape untuk pemodelan sistem fisik dan Simulink untuk pemodelan sistem kontrol.

Model 3D awalnya dibuat dalam format URDF untuk digunakan di Gazebo dengan tujuan melakukan simulasi fisik robot yang dikendalikan oleh pengontrol dari Simulink. Namun, simulasi melalui MATLAB menunjukkan performa yang sangat lambat dan tidak sinkron dengan waktu simulasi di Gazebo, sehingga pengontrol gagal mengendalikan *self-balancing robot* secara efektif.

Untuk mengatasi masalah ini, digunakan Simscape untuk membangun ulang model 3D robot langsung di lingkungan

MATLAB. Model URDF dari Gazebo kemudian dikonversi ke dalam format file .slx agar dapat digunakan sebagai model Simscape. Dengan demikian, seluruh proses simulasi—baik aspek fisika maupun logika kontrol—berjalan sepenuhnya di dalam ekosistem MATLAB/Simulink tanpa perlu komunikasi eksternal dengan Gazebo. Pendekatan ini secara signifikan mengurangi *overhead* komunikasi dan menghasilkan simulasi yang lebih cepat serta efisien. Berikut merupakan model 3D beserta arsitektur sistem keseluruhannya.

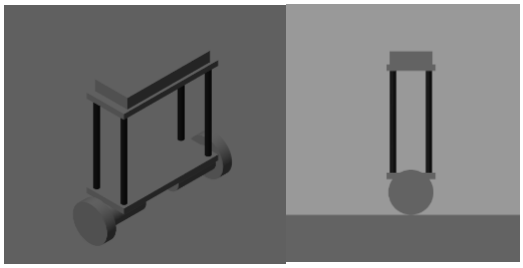


Fig. 6. Bentuk 3D dari Self-Balancing Robot

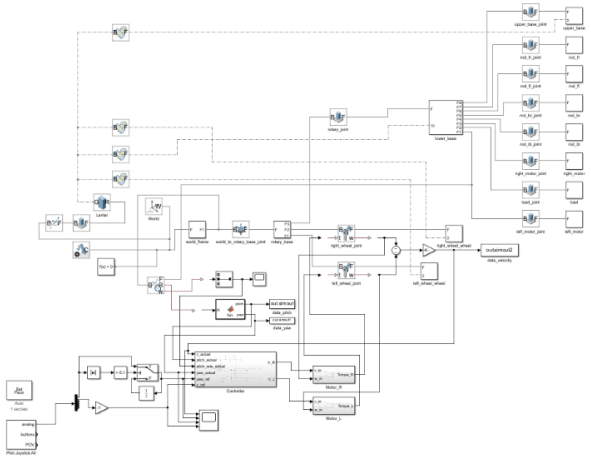


Fig. 7. Diagram Keseluruhan Simulink

Diagram di atas menunjukkan arsitektur lengkap dari sistem simulasi. Secara garis besar, alur sinyal pada sistem ini dapat dibagi menjadi empat bagian utama:

1. Input Sistem: Menerima perintah dari pengguna melalui *joystick*.
2. Blok Kontroler: Memproses data dari sensor dan input referensi untuk menghasilkan sinyal kendali (tegangan motor).
3. Plant (Model Fisik Robot): Merupakan representasi virtual dari robot yang dibangun menggunakan Simscape Multibody yang merespons sinyal kendali dari kontroler.
4. Umpan Balik (Feedback): Sensor virtual [ada model Simscape mengukur keadaan robot (termasuk kecepatan, sudut *pitch*, sudut *yaw*) dan mengirimkannya kembali ke blok kontroler.

## 4.2 Input Sistem: Pilot Joystick

Bagian input sistem, seperti yang ditunjukkan pada gambar 4.2, berfungsi sebagai antarmuka antara pengguna dan sistem kendali sekaligus sebagai pemberi *reference*.

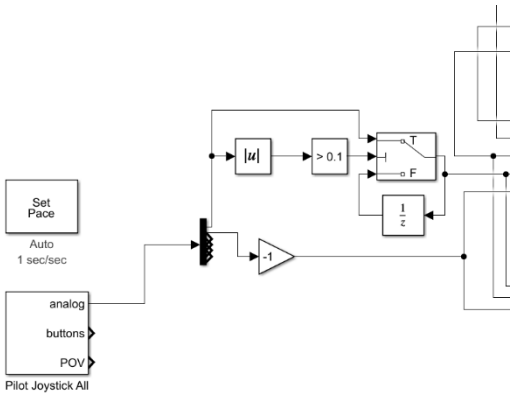



Fig. 8. Diagram Blok Bagian Input

Sistem akan menerima input dari *joystick* melalui port analog blok “Input Joystick All”. Sinyal dari port ini akan memberikan nilai refrensi untuk kecepatan dan sudut *yaw*. Pada pemberian referensi sudut *yaw*, nilainya akan kembali ke nol setiap kali input *joystick* kembali ke posisi normal. Oleh karena itu, diberikan *zero-order-hold* untuk membuat referensi sudut ditahan pada nilai terakhir yang diberikan oleh input *joystick* sebelum kembali ke posisi normal.

## 4.3 Blok Kontroler dan Subsistem Motor

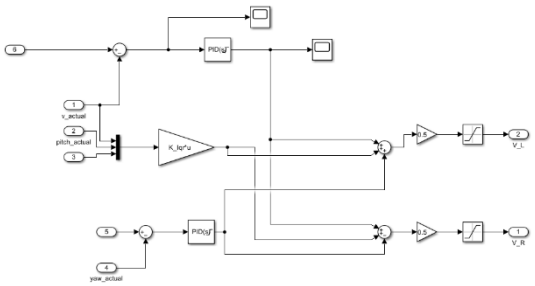


Fig. 9. Diagram Blok Kontroler

Blok kontroler merupakan pusat komputasi atau otak dari sistem kendali. Di dalamnya terimplementasi strategi kontrol hibrida yang menggabungkan *Linear-Quadratic Regulator* (LQR) untuk tugas stabilisasi dan dua kontroler PID untuk pelacakan referensi. LQR secara spesifik bertugas untuk menjaga keseimbangan robot dengan cara meminimalkan simpangan sudut kemiringan (*pitch*) dan laju perubahannya (*pitch rate*). Sementara itu, satu kontroler PID bertanggung jawab untuk mengatur kecepatan linear robot agar sesuai dengan referensi dari pengguna, dan satu PID lainnya mengatur pergerakan belok (*yaw*). Blok ini menerima masukan dari sensor virtual (kecepatan, *pitch*, dan *yaw* aktual) serta referensi dari *joystick*, kemudian menghasilkan sinyal keluaran berupa tegangan spesifik untuk motor kiri (VL) dan motor kanan (VR).

Sinyal tegangan dari kontroler kemudian diteruskan ke subsistem motor yang berfungsi sebagai aktuator atau otot

pada sistem. Setiap subsistem motor berisi model matematika dari sebuah motor DC yang secara realistis mengubah input tegangan menjadi torsi mekanis untuk memutar roda. Model ini juga telah memperhitungkan efek GGL Balik (*back-EMF*), di mana kecepatan putar roda akan mempengaruhi torsi yang dihasilkan, sehingga perilakunya lebih mendekati motor fisik di dunia nyata. Penting untuk dicatat bahwa seluruh parameter kunci yang digunakan dalam blok kontroler dan motor—seperti gain LQR (Klqr), gain PID (Kp, Ki, Kd), dan konstanta motor—didefinisikan dan dihitung dalam skrip inisialisasi MATLAB yang dijalankan sebelum simulasi.

### 4.4 Blok Pemodelan Robot

Bagian pemodelan fisik, atau yang biasa disebut *plant*, adalah representasi virtual dari *self-balancing robot* yang dibangun menggunakan *library* Simscape Multibody. Berbeda dengan Simulink konvensional yang berbasis aliran sinyal, Simscape memungkinkan perakitan model layaknya objek fisik di dunia nyata. Model ini terdiri dari beberapa komponen `Bodies` (Bodi) yang merepresentasikan sasis dan kedua roda. Setiap bodi ini diberikan properti fisis seperti massa dan momen inersia yang nilainya didefinisikan dalam skrip inisialisasi MATLAB. Bodi-bodi tersebut kemudian dihubungkan oleh `Joints` (Sendi) yang menentukan bagaimana mereka dapat bergerak relatif satu sama lain, contohnya adalah sendi putar (*revolute joint*) pada poros yang memungkinkan roda berotasi.

Model fisik ini berinteraksi secara dua arah dalam loop simulasi. Sebagai aktuasi, model ini menerima input berupa torsi mekanis dari subsistem motor yang kemudian diterapkan langsung pada sendi roda, menyebabkan keseluruhan model bergerak sesuai dengan hukum fisika Newton. Di sisi lain, untuk menyediakan umpan balik, pada model ini dipasang `Transform Sensor` virtual. Sensor-sensor ini bertugas mengukur keadaan sistem secara *real-time*, seperti sudut pada sendi utama untuk mendapatkan nilai *pitch* aktual dan kecepatan putar roda untuk mendapatkan nilai kecepatan linear. Data keluaran dari sensor-sensor inilah yang kemudian dikirim kembali ke blok kontroler untuk diproses, sehingga membentuk sebuah loop kendali tertutup yang lengkap dan realistis.

### 4.5 Variasi LQR

Untuk mendapatkan performa terbaik dari kontroler LQR, dilakukan serangkaian pengujian dengan memvariasikan nilai pembobotan pada matriks Q. Matriks Q, yaitu diag([v, pitch, pitch_rate]), menentukan seberapa besar "hukuman" yang diberikan untuk setiap deviasi *state* dari titik setimbangnya. Tujuan utama dari LQR di sini adalah untuk stabilisasi, sehingga prioritas diberikan untuk menjaga *pitch* dan *pitch rate* sekecil mungkin.

Setiap variasi kontroler diuji menggunakan skenario manuver yang sama untuk memastikan perbandingan yang adil. Skenario tersebut adalah sebagai berikut:

1. **t = 0-2s:** Robot dalam keadaan diam.
2. **t = 2-4s:** Robot berakselerasi maju.
3. **t = 4-6s:** Robot bergerak maju dan berbelok ke kanan secara simultan.
4. **t = 6-7s:** Robot melakukan pengereman/mundur mendadak untuk menguji respons terhadap gangguan besar.
5. **t > 7s:** Robot kembali ke keadaan diam, menguji kemampuan untuk meredam osilasi.

Beberapa pengujian juga menyertakan skenario kedua yaitu "lepas gas" atau pengereman tidak ekstrem untuk melihat perilaku robot saat kembali ke kondisi setimbang dari kecepatan tertentu. Berikut adalah hasil dari variasi dari LQR:

- **Variasi #1**

  $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

  R = 1

  Gain K = [-3.4226 156.3086 17.8584]

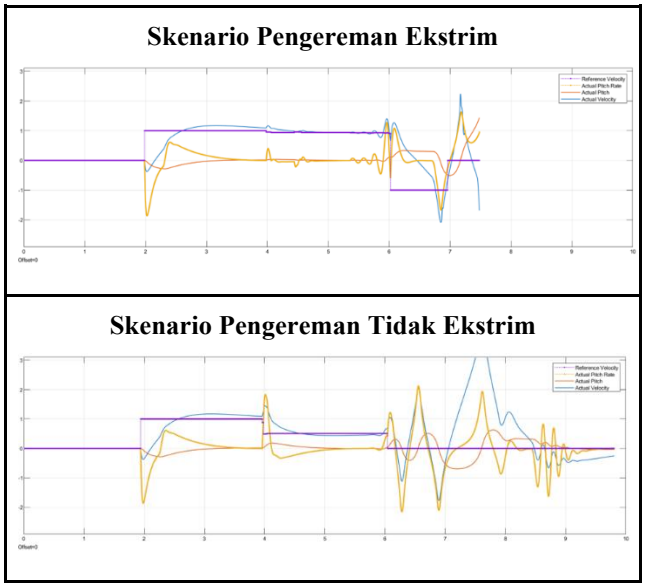


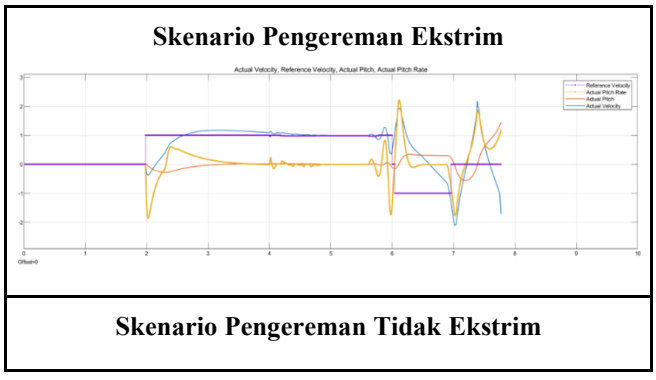

Fig. 10. Hasil Variasi LQR 1

- **Variasi #2**

  $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

  R = 1

  Gain K = [-3.4226 156.9159 17.8914]



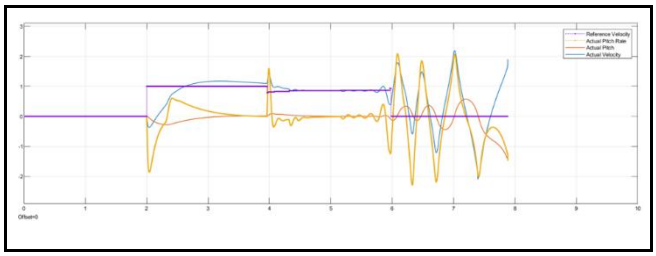


Fig. 11. Hasil Variasi LQR 2

- **Variasi #3**

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

R = 1

Gain K = [-3.4226 157.9715 20.6239]

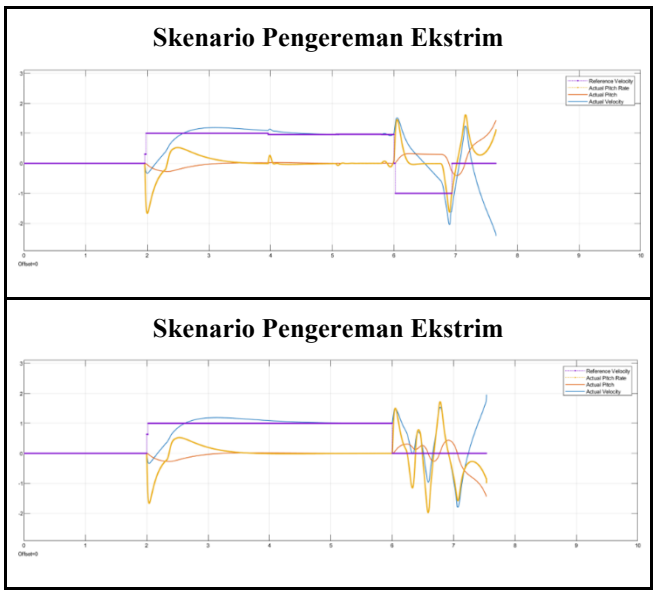


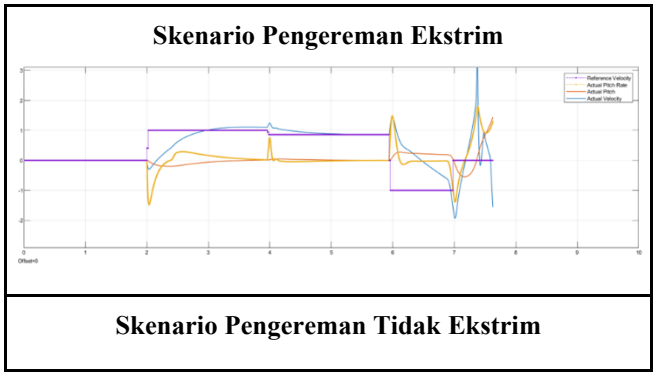

Fig. 12. Hasil Variasi LQR 3

- **Variasi #4**

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

R = 1

Gain K = [-11.6870 209.5571 24.4689]



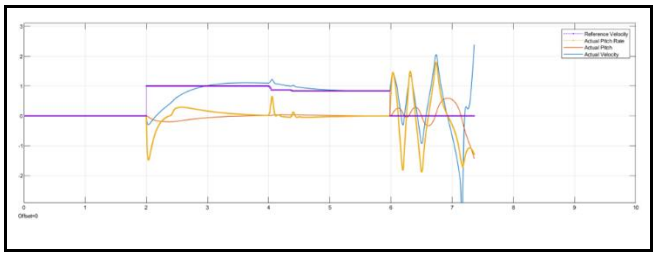



Fig. 13. Hasil Variasi LQR 4

- **Variasi #5**

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

R = 1

Gain K = [-3.4226 158.5639 20.6515]

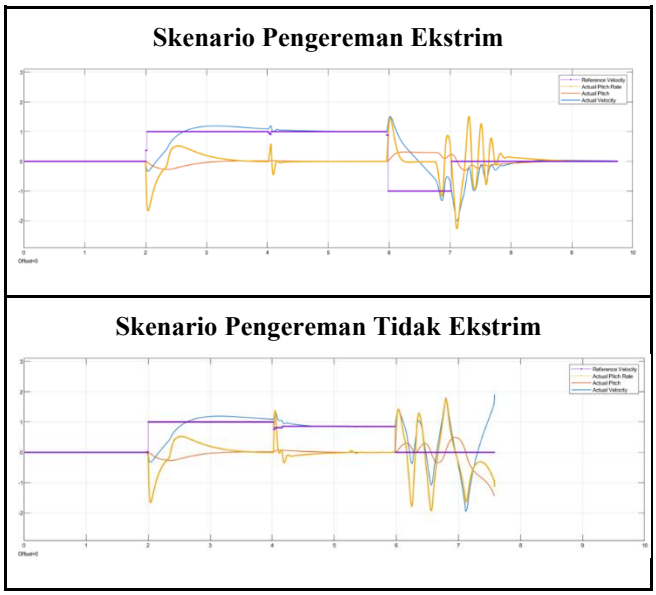


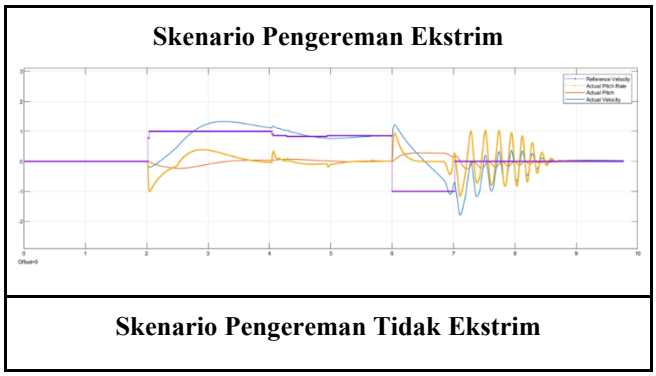

Fig. 14. Hasil Variasi LQR 5

- **Variasi #6**

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$

R = 1

Gain K = [-3.4226 172.4103 37.1212]



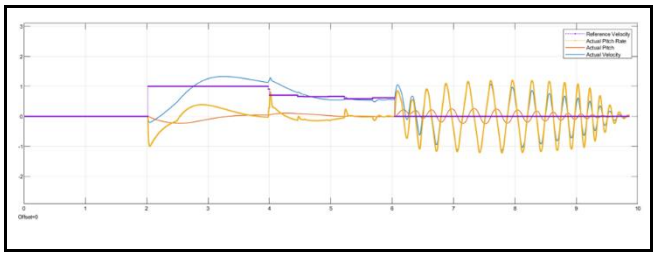


Fig. 15. Hasil Variasi LQR 6

- **Variasi #7 - #9**

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$

R = 1

Gain K = [-5.0937 189.1442 38.2728]

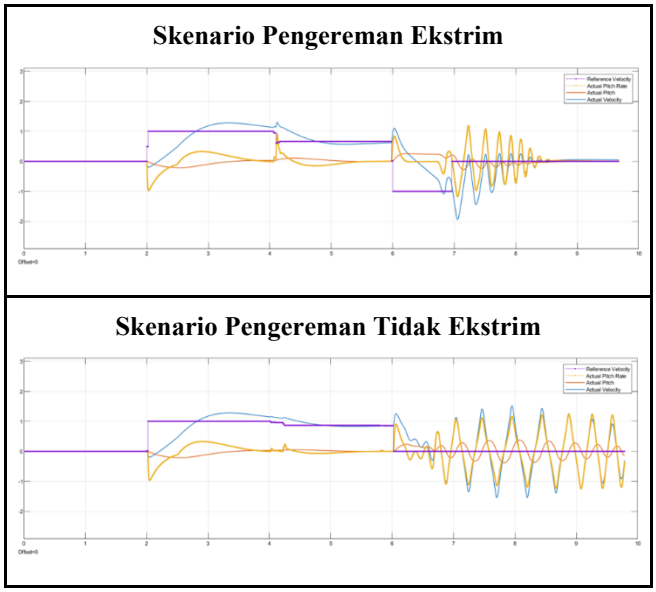



Fig. 16. Hasil Variasi LQR 7

- $Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$

R = 1

Gain K = [-11.687 245.3587 42.6556]

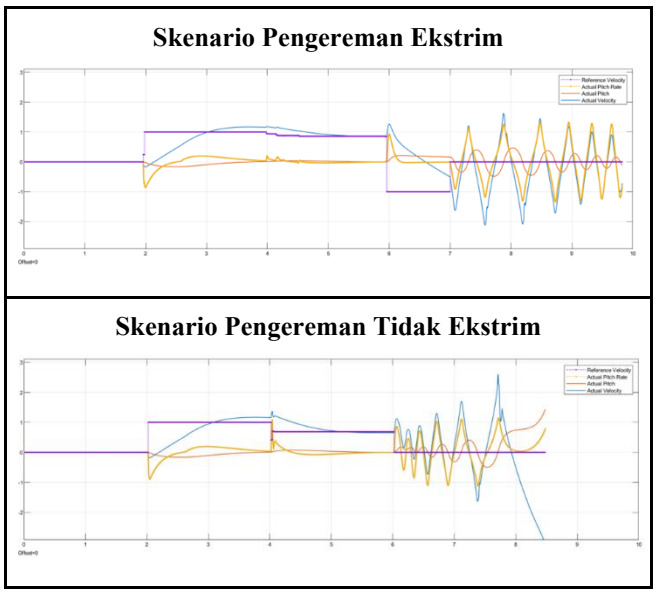



Fig. 17. Hasil Variasi LQR 8

- $Q = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$

R = 1

Gain K = [-3.2827 170.9287 37.0232]

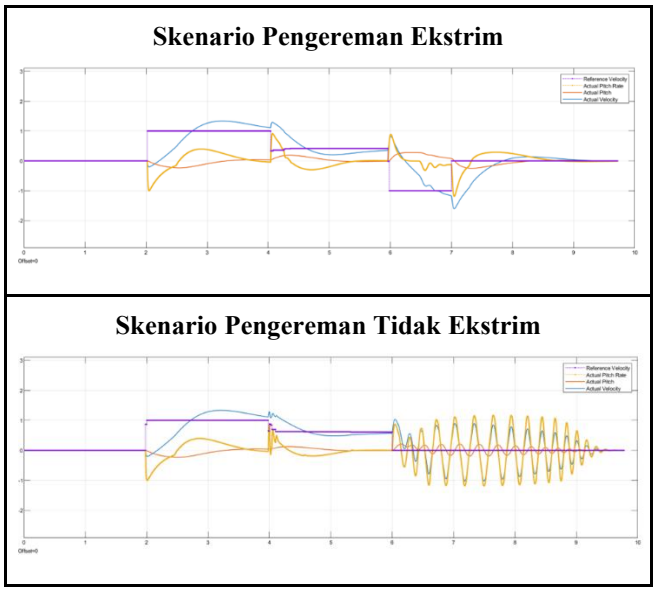



Fig. 18. Hasil Variasi LQR 9

- **Variasi #10**

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 500 \end{bmatrix}$$

R = 1

Gain K = [-3.4226 168.3392 29.2945]

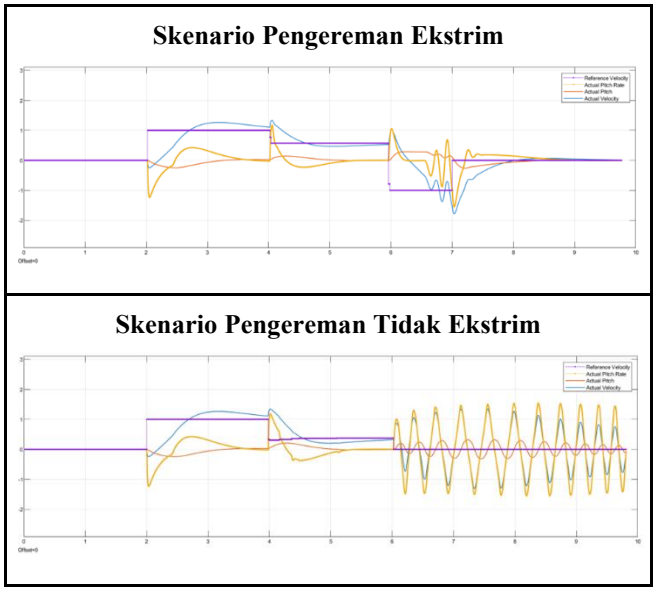


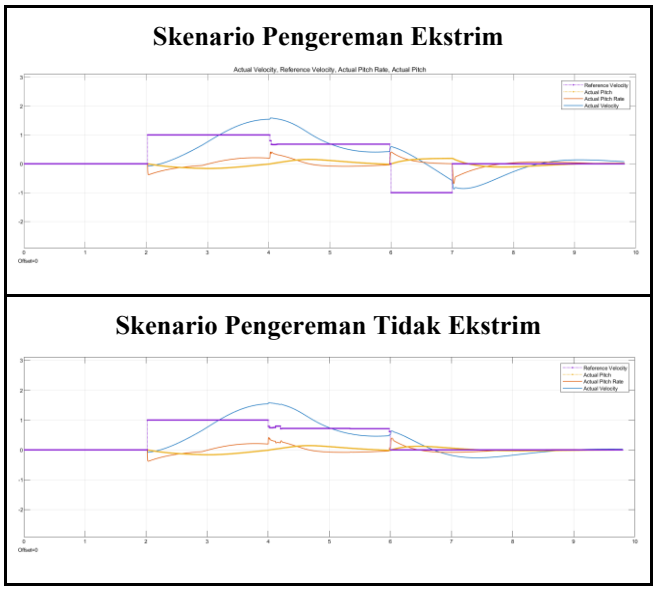

Fig. 19. Hasil Variasi LQR 10

- **Variasi #11**

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10000 & 0 \\ 0 & 0 & 10000 \end{bmatrix}$$

R = 1

Gain K = [-3.4226 232.7269 102.9224]

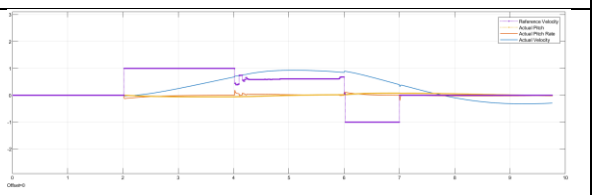
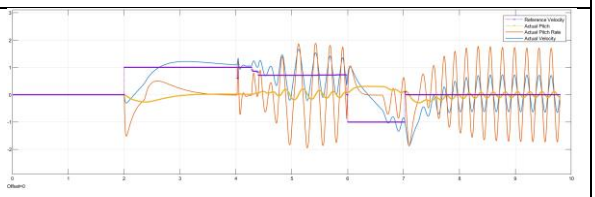
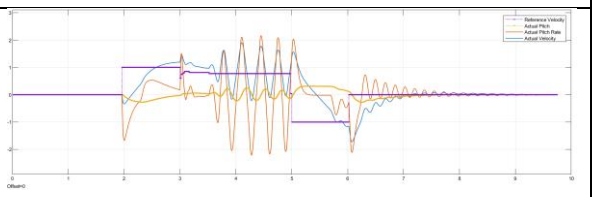
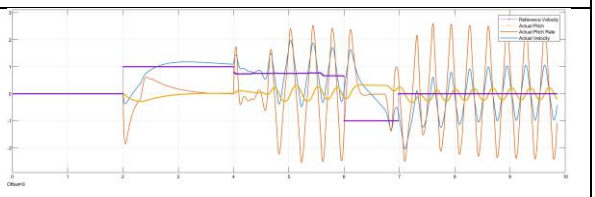



- **Variasi R**

  Variasi R dilakukan dengan kondisi pengereman ekstrim saja dengan parameter Q yang menghasilkan respon terbaik.

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$

TABLE II VARIASI R TERHADAP HASIL RESPON

| R | Hasil Respon |
|---|---|
| 0.01 |  |
| 5 |  |
| 10 |  |
| 100 |  |

Pada variasi 4, 7 sampai 9 dilakukan peningkatan pembobotan penalti terhadap state velocity. Dengan adanya peningkatan penalti, sistem ingin velocity agar secepat mungkin menuju set point. Perubahan ini menyebabkan peningkatan percepatan dan perlambatan namun terjadi overshoot velocity. Ketika terjadi percepatan atau perlambatan yang cepat, frame robot memberikan kompensasi dengan menghasilkan nilai state pitch yang besar. Untuk jenis robot self-balancing tidak ideal karena perlu mempertahankan orientasi frame robot agar tidak jatuh, dengan nilai pitch yang besar muncul resiko besar untuk robot terjatuh. Oleh karena itu, nilai pembobotan penalti terhadap state velocity sebaiknya kecil saja untuk memberikan ruang robot agar dapat menstabilkan orientasi frame. Respon dari variasi 6 mendemonstrasikan hal ini dengan baik ketika memprioritaskan pembobotan penalti terhadap pitch dan pitch rate dibandingkan velocity.

Pembobotan penalti terhadap state pitch, menyebabkan peningkatan respon sistem untuk mempertahankan state pitch tetap 0 atau frame tegak. Dapat dilihat pada variasi 5 dan 6 dengan ditingkatkannya penalti terhadap state pitch, deviasi pitch tidak terlalu besar dibandingkan variasi 1. Namun, dengan deviasi pitch yang kecil terjadi osilasi state velocity ketika dilakukan pengereman. Hal ini terjadi karena sistem harus tetap menjaga agar frame robot tetap tegak meskipun terdapat efek inersia ketika dilakukan pengereman.

Pembobotan penalti terhadap state pitch rate, menyebabkan perubahan pitch agar tetap 0 atau kemiringan frame tidak berubah. Pada variasi 1, 10, dan 6, dapat dilihat dengan peningkatan signifikan penalti terhadap pitch rate, frame robot dapat terjaga tegak. Pitch rate merupakan hal yang penting untuk robot jenis self-balancing, pitch rate terlalu besar dapat menyebabkan robot untuk terjatuh. Hal ini dapat dilihat seperti pada variasi 1 dimana penalti terhadap pitch rate kecil atau pitch rate yang besar diperbolehkan. Hal ini tentunya tidak baik ketika sedang akselerasi atau deselerasi karena adanya efek inersia.

Pembobotan pada R menyebabkan penalti terhadap sinyal control. Semakin besar nilai R maka sinyal kontrol yang boleh dikirim ke aktuator semakin kecil atau dapat disebut dengan kontrol yang “mahal”. Sinyal kontrol yang tidak cukup dapat menyebabkan robot self-balancing menjadi tidak stabil. Ini dapat dilihat ketika nilai R bernilai 5 hingga 100, terjadi osilasi ketika melakukan pengereman. Ketika kontrol “murah” atau bobot R kecil, maka robot akan mendapatkan respon yang diinginkan karena dapat memanfaatkan aktuator sepenuhnya. Dapat dilihat pada saat nilai R bernilai 0.01, respon pitch dan pitch rate sangat bagus namun respon velocity sangat lambat sesuai dengan pembobotan Q.

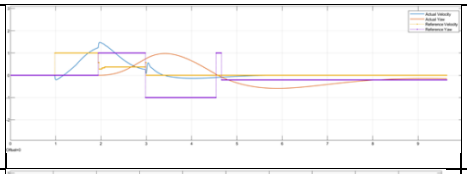
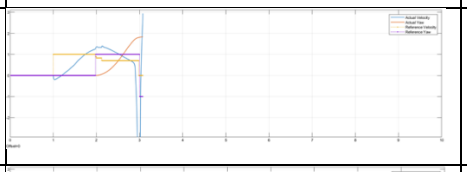
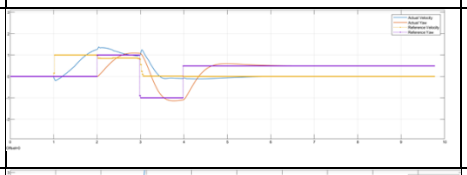
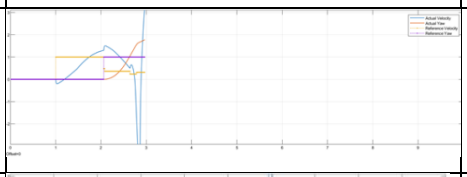
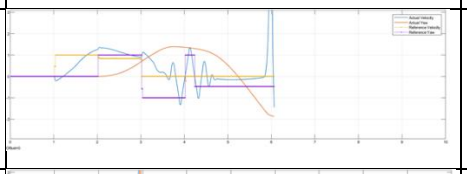
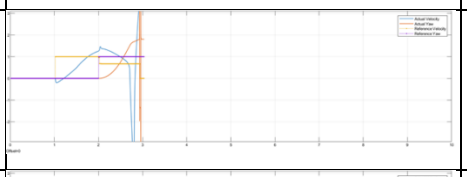
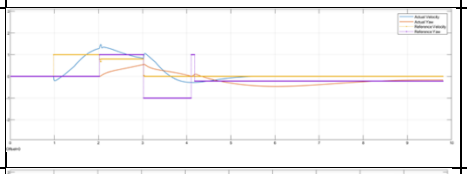
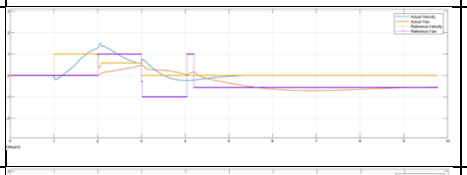
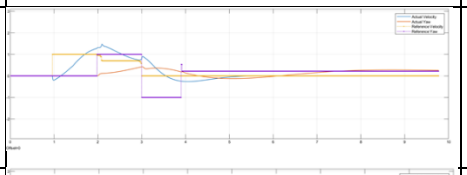
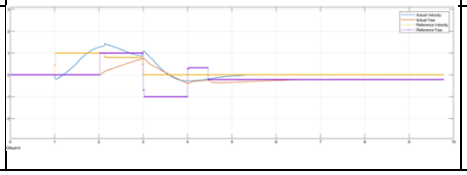
4.6 Variasi PID Yaw Controller

Selain kontroler LQR untuk stabilisasi, sistem ini juga menggunakan kontroler PID terpisah untuk mengatur pergerakan rotasi atau *yaw*. *Tuning* parameter PID (Proporsional, Integral, Derivatif) yang tepat sangat krusial untuk menghasilkan respons belok yang cepat, akurat, dan tidak mengganggu keseimbangan utama robot.

Serangkaian pengujian dilakukan dengan berbagai kombinasi parameter PID. Skenario pengujiannya adalah robot diam selama 1 detik, lalu maju selama 1 detik, kemudian maju dengan berbelok ke kanan secara simultan selama 1 detik, dan terakhir berbelok ke kiri selama 1 detik.

TABLE III. VARIASI PARAMETER TERHADAP HASIL RESPON

| No | Kp | Ki | Kd | Hasil Respon | |
|---|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | 0.2 | 0 | 0 |  |
| 2 | 1 | 0 | 0 |  |
| 3 | 4.8 | 4.8 | 1.2 |  |
| 4 | 0 | 1 | 0 |  |
| 5 | 0.2 | 0.2 | 0 |  |
| 6 | 1 | 1 | 0 |  |
| 7 | 1 | 1 | 1 |  |
| 8 | 2 | 2 | 2 |  |
| 9 | 5 | 5 | 5 |  |
| 10 | 5 | 5 | 3 |  |

Dari hasil berbagai variasi, terdapat beberapa variasi konstanta PID yang menyebabkan robot terjatuh. Dapat dilihat pada tabel diatas robot terjatuh ketika variasi hanya P saja atau I saja dengan nilai lebih dari atau sama dengan 1 tanpa damping dari konstanta D. Variasi PI tanpa damping konstanta D juga menyebabkan robot juga tetap terjatuh. Artinya, konstanta D sangat berperan untuk menjaga robot agar tidak terjatuh ketika berbelok. Oleh karena itu, harus digunakan variasi ketiga konstanta P, I dan D.

Pada variasi 3, 7 sampai 10, dilakukan variasi ketiga konstanta P, I, dan D. Dapat dilihat dari hasil respon bahwa robot tidak terjatuh ketika berbelok. Meskipun robot tidak terjatuh, variasi tersebut memiliki respon rise time, dan settling time yang berbeda. Konstanta D yang tinggi menyebabkan respon mengejar set point yang lambat seperti pada variasi 7, 8, 9. Artinya, konstanta D tidak boleh terlalu tinggi agar respon lebih cepat. Ini dapat dilihat pada variasi 10 dengan diturunkannya konstanta D agar tidak sama dengan konstanta P dan I, respon lebih cepat dibandingkan variasi 9. Variasi 3 memiliki respon terbaik dimana nilai-nilai konstanta PIDnya diperoleh melalui rekomendasi metode tuning ziegler-nichols.

## V. CONCLUSIONS

Penelitian ini telah berhasil mengembangkan dan memvalidasi sebuah arsitektur kontrol terdekomposisi untuk robot penyeimbang otomatis. Proses dimulai dengan penurunan model dinamis matematis dari sistem, yang kemudian direpresentasikan dalam bentuk ruang keadaan (state-space). Selanjutnya, dilakukan perancangan sistem kontrol dan evaluasi kinerjanya melalui simulasi di lingkungan MATLAB/Simulink dan Simscape.

Temuan utama dari penelitian ini adalah efektivitas strategi dekomposisi kontrol dalam mengatasi tantangan yang ditimbulkan oleh keterkaitan (coupling) antara dinamika longitudinal dan lateral. Upaya awal untuk menggunakan satu kontroler LQR-Integrator terpusat terbukti tidak mampu menstabilkan semua variabel secara simultan. Solusi yang diusulkan, yaitu membagi sistem menjadi dua subsistem—longitudinal (keseimbangan pitch dan kecepatan) yang dikontrol dengan LQR-PID dan lateral (yaw) yang dikontrol dengan PID—terbukti mampu mencapai kestabilan dan performa pelacakan yang diinginkan. Arsitektur LQR-PID untuk kontrol longitudinal memberikan keunggulan berupa stabilisasi optimal dari LQR sekaligus performa pelacakan referensi yang cepat dan fleksibel dari PID.

Dari sisi implementasi, migrasi dari platform simulasi terdistribusi (MATLAB/Gazebo) ke lingkungan terintegrasi penuh dalam MATLAB/Simscape secara signifikan meningkatkan efisiensi dan kecepatan simulasi, memungkinkan validasi kontroler yang lebih andal. Simulasi yang dilakukan dengan berbagai variasi parameter pada matriks bobot Q dan R mengonfirmasi bahwa penyesuaian pada kontroler LQR memiliki pengaruh langsung terhadap karakteristik respons sistem, seperti kecepatan mencapai kestabilan dan akurasi pelacakan, yang menunjukkan adanya trade-off antara performa dan penggunaan energi kontrol.

Secara keseluruhan, penelitian ini berhasil mendemonstrasikan perancangan, implementasi, dan validasi sistem kontrol yang robust untuk robot penyeimbang. Arsitektur kontrol hibrida dan terdekomposisi ini menawarkan solusi praktis untuk sistem mekatronika kompleks yang memiliki dinamika terkendali. Penelitian selanjutnya dapat difokuskan pada implementasi pada perangkat keras fisik, optimisasi parameter kontroler menggunakan algoritma cerdas, atau penambahan kemampuan navigasi otonom yang lebih canggih seperti pemetaan dan penghindaran rintangan.

## REFERENCES


[1] Asali, M.O., Hadary, F. and Sanjaya, B.W., 2017. Modeling, simulation, and optimal control for two-wheeled self-balancing robot. International Journal of Electrical and Computer Engineering, 7(4), p.2008.

[2] Alshahrani, H.N.A., 2020. Smart Two Wheels Balancing Robot (Doctoral dissertation, Flinders University, College of Science and Engineering.

[3] H. Khabazi and M. Shariyat, "Control and Simulation of Autonomous Navigation of a Self-Balancing Robot*," in *2024 12th RSI International Conference on Robotics and Mechatronics (ICRoM)*, Tehran, Iran, Islamic Republic of: IEEE, Dec. 2024, pp. 226–232. doi: 10.1109/ICRoM64545.2024.10903531.