

# SIMULASI KONTROL *ROLL, PITCH, DAN YAW* PADA *QUADROTOR* MENGGUNAKAN PID DAN LQR

Muhammad Rifky Santoso<sup>\*)</sup>, Wahyudi, dan Sudjadi

Departemen Teknik Elektro, Universitas Diponegoro  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)E-mail: mikotorifky8@gmail.com</sup>

## Abstrak

Quadrotor memiliki 4 buah propeller yang dapat bergerak dengan memvariasi putaran propeller. Dibutuhkan controller yang tepat agar dapat mengendalikan gerak quadrotor sehingga bisa mencapai posisi yang diinginkan. Salah satu solusinya dengan merancang simulasi kontrol gerak quadrotor menggunakan LQR dan PID. LQR memiliki 2 parameter kontrol yaitu matriks pembobot state (Q) dan matriks pembobot input (R) sedangkan PID memiliki 3 parameter kontrol yaitu proporsional (P), integral (I), dan derivative (D). Penentuan parameter kontrol LQR melalui trial error sebanyak 10 kali sedangkan penentuan parameter kontrol PID melalui metode tuning Ziegler-Nichols. Pengujian kontrol gerak quadrotor menggunakan LQR dan PID dilakukan pada gerakan roll, pitch, dan yaw untuk mengikuti sudut referensi yang diberikan. Hasil pengujian pada gerakan roll dan pitch menunjukkan spesifikasi respon dari LQR memiliki rise time sebesar 140,482 milidetik, overshoot sebesar 0,505%, dan setling time sebesar 158,897 milidetik untuk spesifikasi respon dari PID memiliki rise time sebesar  $109,81 \times 10^3$  milidetik, overshoot sebesar 80,909%, dan setling time sebesar  $3372 \times 10^3$  milidetik. Pada gerakan yaw menunjukkan spesifikasi respon dari LQR memiliki rise time sebesar 126,86 milidetik, overshoot sebesar 0,501%, dan setling time sebesar 169,883 milidetik untuk spesifikasi respon dari PID memiliki rise time sebesar  $43,88 \times 10^3$  milidetik, overshoot sebesar 77,679%, dan setling time sebesar  $1223 \times 10^3$  milidetik.

**Kata Kunci:** *roll, pitch, yaw, LQR, PID*

## Abstract

Quadrotor has 4 propellers that can move by varying the propeller rotation. It takes the right controller in order to control the quadrotor motion so as to achieve the desired position. One solution is by designing a quadrotor motion control simulation using LQR and PID. The LQR has two control parameters, the state weighted matrix Q and the input weighted matrix R whereas the PID has 3 parameter controls that are proportional (P), integral (I), and derivative (D). Determination of LQR control parameters through trial error 10 times while determining PID control parameters through Ziegler-Nichols tuning method. The quadrotor motion control test using LQR and PID is performed on roll, pitch, and yaw motion to follow the given reference angle. Test results on roll and pitch motions show the response specification of LQR has a rise time of 140.482 milliseconds, overshoot of 0.505%, and setling time of 158.897 milliseconds for response specification from PID has rise time  $109.81 \times 10^3$  millisecond, overshoot equal to 80.909%, and setling time of  $3372 \times 10^3$  milliseconds. In yaw motion shows the response specification of LQR has a rise time of 126.86 milliseconds, overshoot of 0.501%, and setling time of 169.883 milliseconds for response specification of PID has rise time of  $43.88 \times 10^3$  milliseconds, overshoot of 77.679%, and setling time of  $1223 \times 10^3$  milliseconds.

**Keywords:** *roll, pitch, yaw, LQR, PID*

## 1. Pendahuluan

Quadrotor merupakan jenis *multicopter* yang memiliki 4 buah *propeller* [1]. Kecepatan putar dari *propeller* dapat divariasikan sehingga timbul gerakan *roll, pitch, dan yaw* [2]. Beberapa penelitian terdahulu telah mencoba menerapkan berbagai metode kontrol gerak *quadrotor* antara lain ; Hoffmann [3] menggunakan EKF (*Extended Kalman Filter*) untuk memperkirakan pengukuran kecepatan dan posisi oleh GPS sedangkan untuk menstabilkan *altitude*

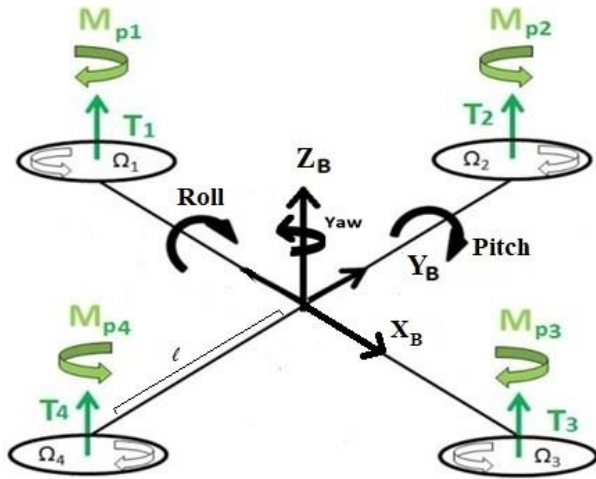
dan *attitude* menggunakan LQR (*Linier Quadratic Regulator*), Gibiansky [4] menggunakan PID. Pada tugas akhir ini bertujuan merancang simulasi kontrol gerakan *roll, pitch, dan yaw* menggunakan LQR dan PID.

## 2. Metode

### 2.1. Sistem Quadrotor

Quadrotor dapat bergerak ke segala arah dikarenakan gaya yang ditimbulkan dari putaran 4 buah *propeller* [5].

Propeller pada quadrotor terdiri atas 2 pasang propeller dengan arah putar yang berlawanan seperti yang ditunjukkan Gambar 1 [5].



Gambar 1 Mekanisme gerak quadrotor.

Pasangan propeller pertama terdiri dari propeller 1 dan 3 sedangkan pasangan propeller kedua terdiri dari propeller 2 dan 4. Gerakan pitch muncul dengan meningkatkan kecepatan putar propeller 1 ( $\Omega_1$ ) dan menurunkan kecepatan putar propeller 3 ( $\Omega_3$ ), begitu juga sebaliknya. Gerakan roll muncul dengan meningkatkan kecepatan putar propeller 2 ( $\Omega_2$ ) dan menurunkan kecepatan putar propeller 4 ( $\Omega_4$ ), begitu juga sebaliknya. Gerakan yaw muncul dengan meningkatkan kecepatan putar pasangan propeller pertama dan menurunkan kecepatan putar pasangan propeller kedua, begitu juga sebaliknya. Gerakan roll menimbulkan perubahan sudut  $\phi$  (phi), gerakan pitch menimbulkan perubahan sudut  $\theta$  (theta), dan gerakan yaw menimbulkan perubahan sudut  $\psi$  (psi) [6]. Model matematis yang menunjukkan gerak rotasi quadrotor terdapat pada persamaan (1), (2), dan (3) [7].

$$\ddot{\phi}(t) = \frac{(\Omega_2^2(t) - \Omega_4^2(t))K_T l}{I_{xx}} \quad (1)$$

$$\ddot{\theta}(t) = \frac{(\Omega_1^2(t) - \Omega_3^2(t))K_T l}{I_{yy}} \quad (2)$$

$$\ddot{\psi}(t) = \frac{(\Omega_1^2(t) - \Omega_2^2(t) + \Omega_3^2(t) - \Omega_4^2(t))K_M}{I_{zz}} \quad (3)$$

$K_T$  adalah konstanta gaya aerodinamis,  $K_M$  adalah konstanta momen aerodinamis,  $l$  adalah panjang lengan quadrotor,  $I_{xx}$  adalah momen inersia di sekitar sumbu  $X_B$  ketika objek berotasi sekitar sumbu  $X_B$ ,  $I_{yy}$  adalah momen inersia di sekitar sumbu  $Y_B$  ketika objek berotasi sekitar sumbu  $Y_B$ ,  $I_{zz}$  adalah momen inersia di sekitar sumbu  $Z_B$  ketika objek berotasi sekitar sumbu  $Z_B$ . Nilai dari sudut  $\phi$  ditentukan dari operasi integral pada percepatan sudut phi ( $\ddot{\phi}$ ) sedangkan nilai dari sudut  $\theta$  ditentukan dari operasi integral pada percepatan sudut theta ( $\ddot{\theta}$ ) begitu juga untuk nilai dari sudut  $\psi$  ditentukan dari operasi integral pada percepatan sudut psi ( $\ddot{\psi}$ ) [8].

$K_T$ ,  $K_M$ ,  $l$ ,  $I_{xx}$ ,  $I_{yy}$ , dan  $I_{zz}$  merupakan parameter fisik quadrotor yang ditentukan berdasarkan Tabel 1 [5].

Tabel 1 Parameter fisik quadrotor.

Parameter	Nilai	Satuan
$I_{xx}$	$7,5 \times 10^{-3}$	$\text{Kg.m}^2$
$I_{yy}$	$7,5 \times 10^{-3}$	$\text{Kg.m}^2$
$I_{zz}$	$1,3 \times 10^{-2}$	$\text{Kg.m}^2$
$l$	0,23	m
$K_T$	$3,1 \times 10^{-5}$	$\text{Kg.m.rad}^{-2}$
$K_M$	$7,5 \times 10^{-7}$	$\text{Kg.m}^2.\text{rad}^{-2}$

## 2.2. Controller

### 2.2.1. Kontrol LQR (Linear Quadratic Regulator)

Metode optimasi dengan LQR bertujuan menentukan sinyal kontrol yang akan memindahkan keadaan suatu sistem dari kondisi awal  $X(t_0)$  menuju ke suatu kondisi akhir  $X(t)$  yang akan meminimumkan cost function [9]. Cost function ( $J_{LQR}$ ) yang dimaksud adalah waktu integral dari bentuk kuadratis pada vektor  $X$  dan vektor  $U$  [10].

$$J_{LQR} = \int_{t_0}^{t_f} (X^T Q X + U^T R U) dt \quad (4)$$

Prinsip penggunaan metode LQR adalah memperoleh sinyal kontrol optimal dari state feedback sesuai persamaan (5) [10].

$$U = -K_{LQR} X \quad (5)$$

$$K_{LQR} = R^{-1} B^T P \quad (6)$$

$K_{LQR}$  adalah gain state feedback,  $P$  adalah solusi persamaan aljabar Riccati. Solusi persamaan aljabar Riccati dihitung berdasarkan persamaan (7) [10].

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (7)$$

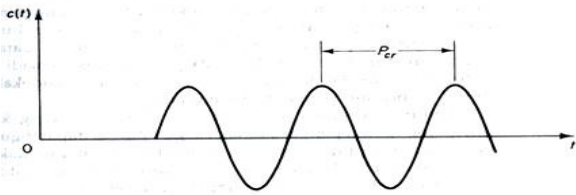
### 2.2.2. Kontrol PID dengan Metode Tuning Ziegler-Nichols

Secara umum kontrol PID memiliki persamaan matematis sebagai berikut [10] :

$$U(t) = K_p(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}) \quad (8)$$

Penerapan metode tuning Ziegler-Nichols bertujuan menentukan nilai parameter  $K_p$ ,  $T_i$ , dan  $T_d$  pada persamaan (8). Metode tuning Ziegler-Nichols dapat diterapkan melalui 2 cara yaitu analisis respon kurva s dan analisis respon osilasi. Penerapan analisis respon kurva s dengan memberikan input step pada sebuah plant tanpa umpan balik (open loop). Jika kurva respon terhadap input step berbentuk mirip huruf s maka analisis kurva respon dapat diterapkan dan sebaliknya. Karena quadrotor tidak menghasilkan kurva respon berbentuk mirip huruf s maka analisis respon osilasi yang diterapkan. Penerapan analisis respon osilasi hanya menggunakan aksi kontrol proporsional pada sistem close loop. Nilai  $K_p$  harus dinaikkan dari 0 sampai nilai kritisnya sehingga menyebabkan output sistem berupa

sinyal yang berosilasi dengan amplitudo yang konstan. Analisis respon osilasi membutuhkan parameter penguatan kritis ( $K_{cr}$ ) dan periode kritis ( $P_{cr}$ ). Nilai  $K_{cr}$  adalah nilai kritis dari  $K_p$  saat sistem mulai menghasilkan *output* berupa sinyal yang berosilasi dengan amplitudo yang konstan. Nilai  $P_{cr}$  didapatkan dengan mengukur periode sinyal osilasi dari puncak ke puncak seperti pada Gambar 2. Parameter  $K_{cr}$  dan  $P_{cr}$  digunakan untuk menghitung nilai  $K_p$ ,  $T_i$ , dan  $T_d$  berdasarkan Tabel 2 [10].



Gambar 2 Pengukuran nilai  $P_{cr}$  dari sinyal yang berosilasi.

Tabel 2 Penentuan nilai  $K_p$ ,  $T_i$ , dan  $T_d$  pada metode *tuning Ziegler-Nichols* berdasarkan respon osilasi.

Tipe Kontrol	$K_p$	$T_i$	$T_d$
P	$0,5K_{cr}$	~	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

2.2. Perancangan Sistem

2.2.1. Perancangan Kontrol LQR pada Quadrotor

Dalam perancangan LQR perlu dihitung nilai  $K_{LQR}$  sehingga matriks  $Q$  dan matriks  $R$  perlu ditentukan terlebih dahulu secara *trial error* dengan tahapan sebagai berikut :

1. Membuat 10 macam variasi dari matriks  $Q$  seperti pada Tabel 3 sedangkan matriks  $R$  ditetapkan sebagai matriks identitas. Matriks  $Q$  yang menyebabkan sistem memiliki respon dengan *rise time* paling kecil akan dipilih.
2. Membuat 10 macam variasi dari matriks  $R$  seperti pada Tabel 5 sedangkan matriks  $Q$  ditetapkan dari variasi matriks  $Q$  yang menyebabkan sistem memiliki respon dengan *rise time* paling kecil. Matriks  $R$  yang menyebabkan sistem memiliki respon dengan *rise time* paling kecil akan dipilih.

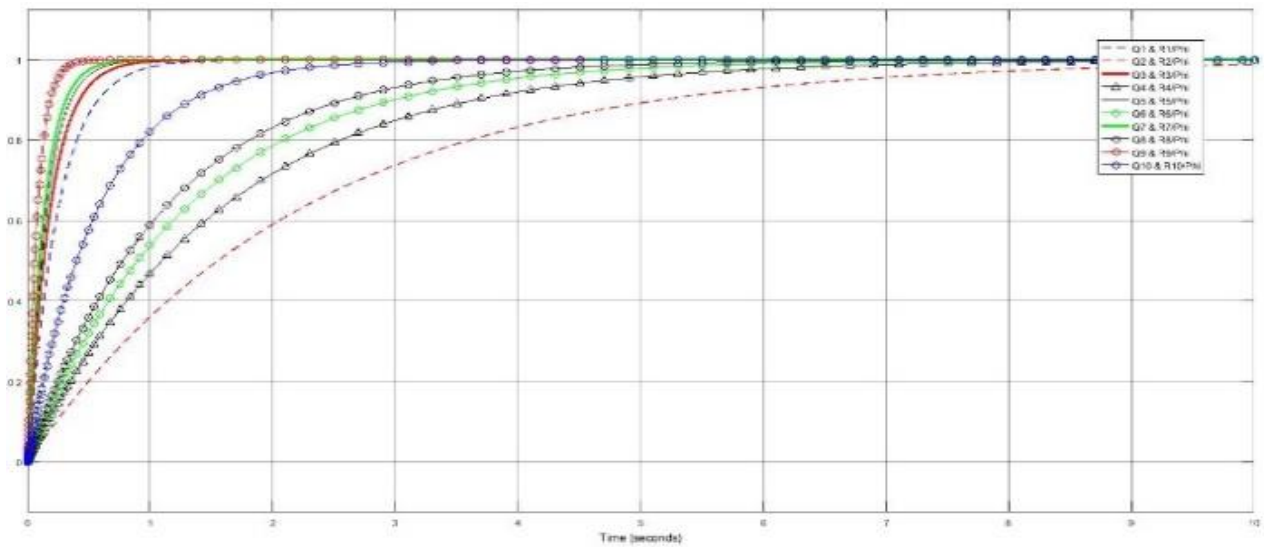
Tabel 3 Variasi matriks  $Q$ .

No	Matriks $Q$
1	$\begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

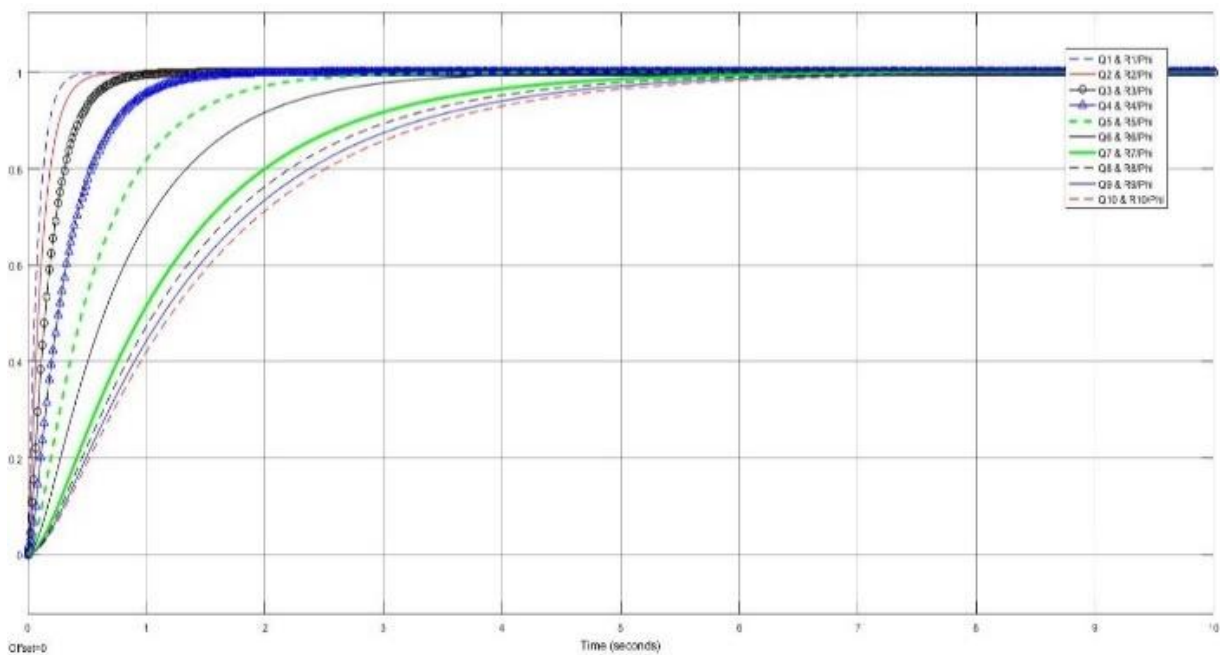
Tabel 3 (lanjutan).

No	Matriks $Q$
2	$\begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$
3	$\begin{bmatrix} 40 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
4	$\begin{bmatrix} 40 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$
5	$\begin{bmatrix} 60 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
6	$\begin{bmatrix} 60 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$
7	$\begin{bmatrix} 80 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 80 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
8	$\begin{bmatrix} 80 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 80 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$
9	$\begin{bmatrix} 300 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 300 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
10	$\begin{bmatrix} 300 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 300 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$

Hasil respon sistem terhadap variasi matriks  $Q$  untuk masukan unit *step* ditunjukkan oleh Gambar 3 sedangkan spesifikasi respon berupa *rise time* ditunjukkan oleh Tabel 4.



Gambar 3. Respon LQR terhadap variasi matriks Q.



Gambar 4 .Respon LQR terhadap variasi matriks R.

Tabel 4 Rise time dari respon variasi matriks Q.

No	Variasi Matriks Q	Rise Time (milidetik)		
		Roll	Pitch	Yaw
1	Q1	534,042	534,042	483,066
2	Q2	4614	4614	4610
3	Q3	392,163	392,163	342,602
4	Q4	3389	3389	3384
5	Q5	328,152	328,152	279,122
6	Q6	2783	2783	2777
7	Q7	290,463	290,463	242,824
8	Q8	2415	2415	2409
9	Q9	170,783	170,783	127,67
10	Q10	1253	1253	1247

Tabel 4 menunjukkan bahwa *rise time* yang paling kecil disebabkan oleh variasi dari matriks Q ke 9 (Q9). Tabel 4 menunjukkan jika nilai elemen diagonal ke 1, 3, dan 5 semakin besar sedangkan nilai elemen diagonal ke 2, 4, dan 6 semakin kecil pada matriks Q maka *rise time* semakin kecil. Proses *trial error* dilanjutkan dengan membuat 10 macam variasi dari matriks R sedangkan matriks Q dipilih dari variasi matriks Q ke 9

Tabel 5 Variasi matriks R.

No	Matriks R
1	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
2	$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
3	$\begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$
4	$\begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$
5	$\begin{bmatrix} 10000 & 0 & 0 \\ 0 & 10000 & 0 \\ 0 & 0 & 10000 \end{bmatrix}$
6	$\begin{bmatrix} 40000 & 0 & 0 \\ 0 & 40000 & 0 \\ 0 & 0 & 40000 \end{bmatrix}$
7	$\begin{bmatrix} 200000 & 0 & 0 \\ 0 & 200000 & 0 \\ 0 & 0 & 200000 \end{bmatrix}$
8	$\begin{bmatrix} 300000 & 0 & 0 \\ 0 & 300000 & 0 \\ 0 & 0 & 300000 \end{bmatrix}$
9	$\begin{bmatrix} 400000 & 0 & 0 \\ 0 & 400000 & 0 \\ 0 & 0 & 400000 \end{bmatrix}$
10	$\begin{bmatrix} 500000 & 0 & 0 \\ 0 & 500000 & 0 \\ 0 & 0 & 500000 \end{bmatrix}$

Hasil respon sistem pada gerakan roll, pitch, dan yaw terhadap variasi matriks R untuk masukan unit step ditunjukkan oleh Gambar 4 sedangkan spesifikasi respon berupa rise time ditunjukkan oleh Tabel 6.

Tabel 6 Rise time dari respon variasi matriks R.

No	Variasi Matriks R	Rise Time (milidetik)		
		Roll	Pitch	Yaw
1	R1	170,323	170,323	126,813
2	R2	242,024	242,024	144,229
3	R3	390,079	390,079	209,482
4	R4	668,92	668,92	340,405
5	R5	1176	1176	596,511
6	R6	1658	1658	824,491
7	R7	2476	2476	1232
8	R8	2733	2733	1362
9	R9	2934	2934	1462
10	R10	3099	3099	1548

Tabel 5 menunjukkan bahwa rise time yang paling kecil disebabkan oleh variasi dari matriks R ke 1 (R1). Tabel 5 menunjukkan elemen diagonal pada matriks R juga memberikan pengaruh ketika nilainya semakin kecil maka rise time semakin kecil. Berdasarkan hasil trial error dapat disimpulkan bahwa variasi matriks Q ke 9 dan matriks R ke 10 menyebabkan sistem memiliki respon cukup baik. Langkah selanjutnya menghitung nilai P yang hasilnya ditunjukkan pada persamaan (11).

$$P = \begin{bmatrix} 44,3968 & 0,0439 & 0 & 0 & 0 & 0 \\ 0,0439 & 0,0011 & 0 & 0 & 0 & 0 \\ 0 & 0 & 44,3968 & 0,0439 & 0 & 0 \\ 0 & 0 & 0,0439 & 0,0011 & 0 & 0 \\ 0 & 0 & 0 & 0 & 43,3061 & 0,0175 \\ 0 & 0 & 0 & 0 & 0,0175 & 0,0004 \end{bmatrix} \quad (11)$$

Nilai P digunakan untuk menghitung  $K_{LQR}$  yang hasilnya ditunjukkan persamaan (12).

$$K_{LQR} = \begin{bmatrix} 1346,1 & 33 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1346,1 & 33 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1346,1 & 32,2 \end{bmatrix} \quad (12)$$

2.2.2 Perancangan Kontrol PID pada Quadrotor

a. Perancangan PID Roll

Langkah awal dalam merancang PID untuk mengontrol gerakan roll adalah menemukan fungsi alih yang mewakilinya. Persamaan (1) mewakili gerakan roll. PID roll akan menghasilkan sinyal kontrol untuk menentukan gaya yang ditimbulkan dari selisih antara  $\Omega_1$  dengan  $\Omega_3$  maka

$$u_1 = (\Omega_1^2(t) - \Omega_3^2(t))K_T \quad (13)$$

Persamaan (13) disubstitusi ke persamaan (1) sehingga

$$\ddot{\phi}(t) = \frac{u_1 l}{I_{xx}} \quad (14)$$

Substitusi nilai parameter fisik quadrotor dan transformasi Laplace diterapkan pada persamaan (14) dengan mengasumsikan semua kondisi awal sama dengan nol.

$$s^2 \Phi(s) = \frac{0,23U_1(s)}{7,5 \times 10^{-3}} \quad (15)$$

Fungsi alih untuk unit roll adalah :

$$\frac{\Phi(s)}{U_1(s)} = \frac{30,67}{s^2} \quad (16)$$

Ketika disimulasikan pada Simulink selama  $10^3$  detik dan nilai  $K_{cr}$  dinaikkan secara bertahap sampai mencapai nilai 0,000005 selalu menghasilkan output berupa sinyal yang berosilasi sedangkan ketika simulasi dijalankan selama  $10^8$  detik menunjukkan hasil simulasi yang terlihat pada Gambar 5.

Gambar 5 menunjukkan terjadi peningkatan amplitudo pada hasil simulasi ketika nilai  $K_{cr}$  sebesar 0,000004 sedangkan ketika nilai  $K_{cr}$  sebesar 0,000005 tidak terjadi peningkatan amplitudo sehingga nilai  $K_{cr}$  yang dipilih adalah 0,000005. Pada sinyal osilasi muncul puncak pertama pada detik ke 260 sedangkan puncak kedua muncul pada detik ke 760 sehingga nilai  $P_{cr}$  adalah :

$$P_{cr} = 760 - 260 = 500 \quad (17)$$

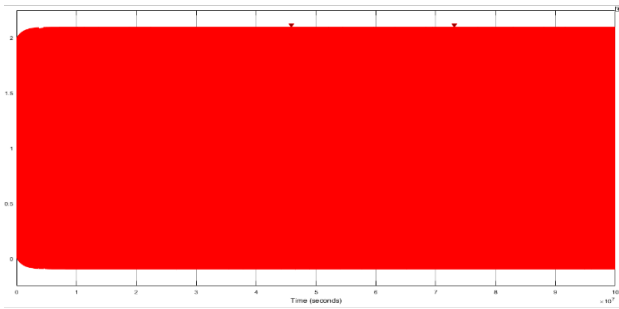
Langkah selanjutnya adalah menghitung nilai  $K_p$ ,  $T_i$ , dan  $T_d$  untuk PID roll berdasarkan Tabel 2.

$$K_p = 0,6 \times 0,000005 = 0,000003 \quad (18)$$

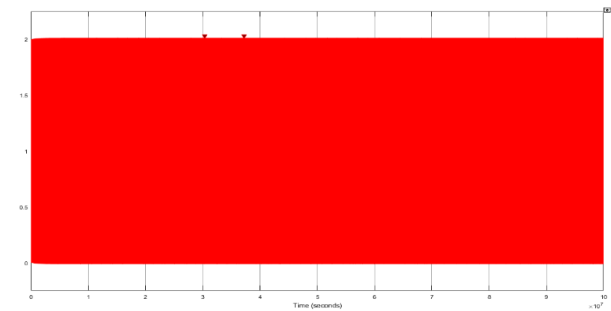
$$T_i = 0,5 \times 500 = 250 \quad (19)$$

$$T_d = 0,125 \times 500 = 62,5 \quad (20)$$

Nilai  $K_p$ ,  $T_i$ , dan  $T_d$  digunakan untuk melengkapi perancangan PID roll sebagaimana ditunjukkan oleh Gambar 6.

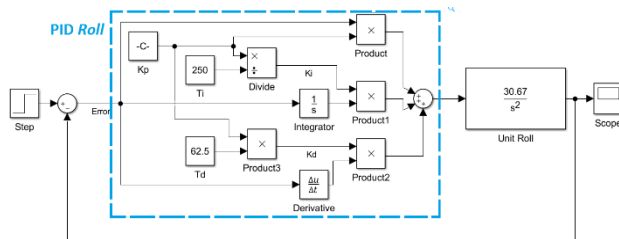


(a) Respon sistem *close loop* pada unit *roll* ketika nilai  $K_{cr}$  sama dengan 0,000004 selama  $10^8$  detik.



(b) Respon sistem *close loop* pada unit *roll* ketika nilai  $K_{cr}$  sama dengan 0,000005 selama  $10^8$  detik.

**Gambar 5. Respon sistem *close loop* dengan controller proporsional pada unit *roll* selama  $10^8$  detik.**



**Gambar 6. Hasil perancangan PID *roll*.**

*b. Perancangan PID Pitch*

Langkah awal dalam merancang PID untuk mengontrol gerakan *pitch* adalah menemukan fungsi alih yang mewakilinya. Persamaan (2) mewakili gerakan *pitch*. PID *pitch* akan menghasilkan sinyal kontrol untuk menentukan gaya yang ditimbulkan dari selisih antara  $\Omega_2$  dengan  $\Omega_4$  maka

$$u_2 = (\Omega_2^2(t) - \Omega_4^2(t))K_T \quad (21)$$

Persamaan (21) disubstitusi ke persamaan (2) sehingga

$$\ddot{\theta}(t) = \frac{u_2 l}{I_{yy}} \quad (22)$$

Substitusi nilai parameter fisik *quadrotor* dan transformasi *Laplace* diterapkan pada persamaan (22) dengan mengasumsikan semua kondisi awal sama dengan nol.

$$s^2 \theta(s) = \frac{0,23 U_2(s)}{7,5 \times 10^{-3}} \quad (23)$$

Fungsi alih untuk unit *pitch* adalah :

$$\frac{\theta(s)}{U_2(s)} = \frac{30,67}{s^2} \quad (24)$$

Ketika disimulasikan pada *Simulink* selama  $10^3$  detik dan nilai  $K_{cr}$  dinaikkan secara bertahap sampai mencapai nilai 0,000005 selalu menghasilkan *output* berupa sinyal yang berosilasi sedangkan ketika simulasi dijalankan selama  $10^8$  detik menunjukkan hasil simulasi yang identik dengan unit *roll* di mana terjadi peningkatan amplitudo pada hasil simulasi ketika nilai  $K_{cr}$  sebesar 0,000004 sedangkan ketika nilai  $K_{cr}$  sebesar 0,000005 tidak terjadi peningkatan amplitudo sehingga nilai  $K_{cr}$  yang dipilih adalah 0,000005. Pada sinyal osilasi muncul puncak pertama pada detik ke 260 sedangkan puncak kedua muncul pada detik ke 760 sehingga nilai  $P_{cr}$  adalah :

$$P_{cr} = 760 - 260 = 500 \quad (25)$$

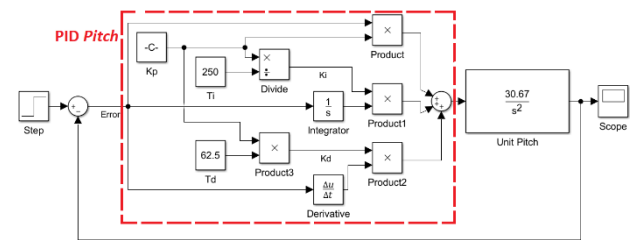
Langkah selanjutnya adalah menghitung nilai  $K_p$ ,  $T_i$ , dan  $T_d$  untuk PID *roll* berdasarkan Tabel 2.

$$K_p = 0,6 \times 0,000005 = 0,000003 \quad (26)$$

$$T_i = 0,5 \times 500 = 250 \quad (27)$$

$$T_d = 0,125 \times 500 = 62,5 \quad (28)$$

Nilai  $K_p$ ,  $T_i$ , dan  $T_d$  digunakan untuk melengkapi perancangan PID *pitch* sebagaimana ditunjukkan oleh Gambar 7.



**Gambar 7 Hasil perancangan PID *pitch*.**

*c. Perancangan PID Yaw*

Langkah awal dalam merancang PID untuk mengontrol gerakan *yaw* adalah menemukan fungsi alih yang mewakilinya. Persamaan (3) mewakili gerakan *yaw*. PID *yaw* akan menghasilkan sinyal kontrol untuk menentukan gaya yang ditimbulkan dari selisih kecepatan putar pasangan *propeller* pertama dengan kedua maka

$$u_3 = (\Omega_1^2(t) - \Omega_2^2(t) + \Omega_3^2(t) - \Omega_4^2(t))K_M \quad (29)$$

Persamaan (29) disubstitusi ke persamaan (3) sehingga

$$\ddot{\psi}(t) = \frac{u_3}{I_{zz}} \quad (30)$$

Substitusi nilai parameter fisik *quadrotor* dan transformasi *Laplace* diterapkan pada persamaan (30) dengan mengasumsikan semua kondisi awal sama dengan nol.

$$s^2 \Psi(s) = \frac{u_3(s)}{1,3 \times 10^{-2}} \quad (31)$$

Fungsi alih untuk unit *yaw* adalah :

$$\frac{\Psi(s)}{U_3(s)} = \frac{76,92}{s^2} \quad (32)$$

Ketika disimulasikan pada *Simulink* selama 500 detik dan nilai  $K_{cr}$  dinaikkan secara bertahap sampai mencapai nilai 0,000013 selalu menghasilkan *output* berupa sinyal yang berosilasi sedangkan ketika simulasi dijalankan selama  $10^8$  detik menunjukkan hasil yang sedikit berbeda dengan unit *roll* di mana terjadi peningkatan amplitudo pada hasil simulasi ketika nilai  $K_{cr}$  sebesar 0,000012 sedangkan ketika nilai  $K_{cr}$  sebesar 0,000013 tidak terjadi peningkatan amplitudo pada hasil simulasinya sehingga nilai  $K_{cr}$  yang dipilih adalah 0,000013. Nilai  $P_{cr}$  untuk PID *yaw* ditentukan dengan mengukur periode sinyal osilasi dari puncak ke puncak. Ketika nilai  $K_{cr}$  sebesar 0,000013 disimulasikan selama 500 detik menunjukkan hasil simulasi berupa sinyal osilasi dengan puncak pertama muncul pada detik ke 100 sedangkan puncak kedua muncul pada detik ke 300 sehingga nilai  $P_{cr}$  untuk PID *yaw* ditentukan sebagai berikut :

$$P_{cr} = 300 - 100 = 200 \quad (33)$$

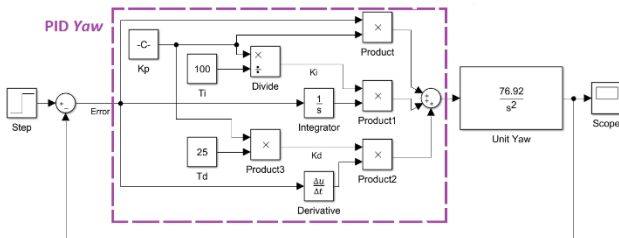
Langkah selanjutnya adalah menghitung nilai  $K_p$ ,  $T_i$ , dan  $T_d$  berdasarkan Tabel 2.

$$K_p = 0,6 \times 0,000013 = 0,0000078 \quad (34)$$

$$T_i = 0,5 \times 200 = 100 \quad (35)$$

$$T_d = 0,125 \times 200 = 25 \quad (36)$$

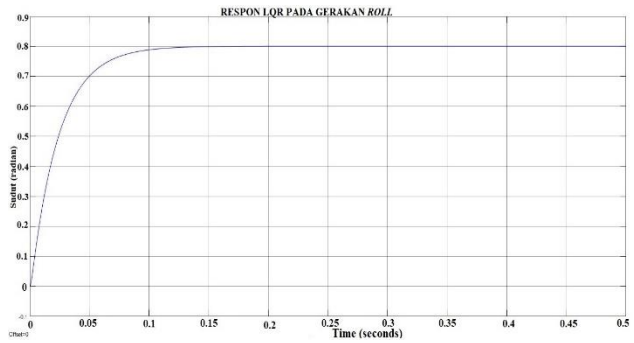
Nilai  $K_p$ ,  $T_i$ , dan  $T_d$  digunakan untuk melengkapi perancangan PID *yaw* sebagaimana ditunjukkan oleh Gambar 8.



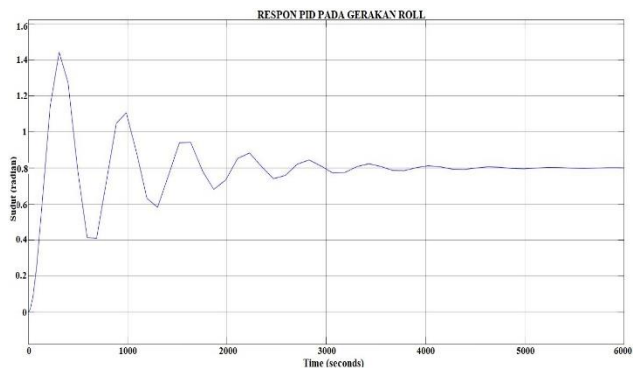
Gambar 8. Hasil perancangan PID *yaw*.

### 3. Hasil dan Analisis

Pengujian dilakukan dengan memberikan masukan *step* untuk merubah sudut dari 0 radian menjadi 0,8 radian. Pengujian pertama dilakukan untuk mengontrol gerakan *roll*. Respon LQR ditunjukkan pada Gambar 9 sedangkan respon PID ditunjukkan pada Gambar 10. Spesifikasi respon dari LQR dan PID pada gerakan *roll* ditunjukkan pada Tabel 7.



Gambar 9. Respon LQR pada gerakan *roll* dan *pitch*.



Gambar 10. Respon PID pada gerakan *roll* dan *pitch*.

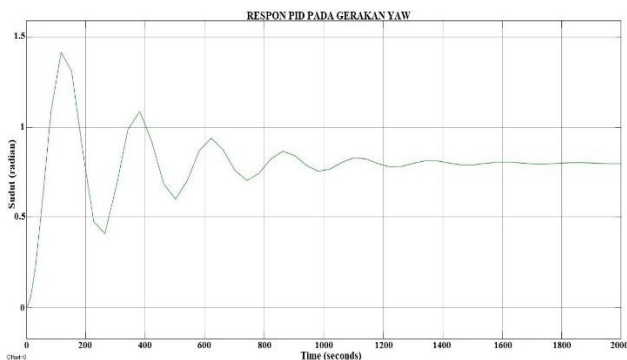
Tabel 7. Spesifikasi respon dari LQR dan PID pada gerakan *roll* dan *pitch*.

Spesifikasi Respon	LQR	PID
Rise time (milidetik)	50,611	$109,81 \times 10^3$
Overshoot (%)	0,501	80,909
Settling time (milidetik)	70,544	$3372 \times 10$

Pengujian kedua dilakukan untuk mengontrol gerakan *pitch*. Respon LQR dan PID menunjukkan hasil pengujian gerakan *pitch* identik dengan hasil pengujian gerakan *roll*. Pengujian ketiga dilakukan untuk mengontrol gerakan *yaw* menggunakan LQR dan PID. Respon LQR pada gerakan *yaw* ditunjukkan pada Gambar 11 sedangkan respon PID ditunjukkan pada Gambar 12. Spesifikasi respon dari LQR dan PID pada gerakan *yaw* ditunjukkan pada Tabel 8.



Gambar 11. Respon LQR pada gerakan *yaw*.



Gambar 12. Respon PID pada gerakan yaw.

Tabel 8. Spesifikasi respon dari LQR dan PID pada gerakan yaw.

Spesifikasi Respon	LQR	PID
Rise time (milidetik)	50,637	$43,88 \times 10^3$
Overshoot (%)	0,501	77,679
Setting time (milidetik)	70,625	$1223 \times 10^3$

#### 4. Kesimpulan

Kontrol LQR memberikan respon yang baik jika dibandingkan dengan PID. Pada pengembangan berikutnya disarankan membahas gerak translasi *quadrotor*. Disarankan pula menerapkan metode optimasi seperti *genetic algorithm* untuk menentukan parameter kontrol LQR dan PID.

#### Referensi

- [1]. T. Bresciani, "Modelling , Identification and Control of a Quadrotor Helicopter," *English*, vol. 4, no. October, p. 213, 2008.
- [2]. F. T. Industri, "TUNING PARAMETER LINEAR QUADRATIC TRACKING TRACKING USING GENETIC ALGORITHM FOR," 2016.
- [3]. G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Eng. Pract.*, vol. 19, no. 9, pp. 1023–1036, 2011.
- [4]. A. Gibiansky, "Quadcopter Dynamics , Simulation , and Control Introduction Quadcopter Dynamics," pp. 1–18, 2012.
- [5]. J. Domingues, "Quadrotor prototype," *Inst. Super. Tec. Univ. Tec. ....*, no. October, p. 129, 2009.
- [6]. H. Arrosida, "Perancangan Metode Kontrol LQR ( Linear Quadratic Regulator ) Sebagai Solusi Optimal Pengendalian Gerak Quadrotor," vol. 1, no. November, pp. 109–122, 2016.
- [7]. S. Bouabdallah, A. Noth, R. Siegwart, and R. Siegwan, "PID vs LQT control techniques applied to an indoor micro quadrotor," *2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2451–2456, 2004.
- [8]. L. R. García Carrillo, A. E. Dzul López, R. Lozano, C. Pégard, and SpringerLink (Online service), "Quad Rotorcraft Control Vision-Based Hovering and Navigation," *Adv. Ind. Control.*, p. XIX, 179 117 illus., 74 illus. in color., 2013.
- [9]. D. Informatics, A. Afis, A. Arisekola, and A. I. Street, "THE USE OF MATLAB IN THE SOLUTION OF LINEAR QUADRATIC REGULATOR ( LQR ) PROBLEMS," vol. 5, no. 4, pp. 15–32, 2014.
- [10]. S. Liberty, *Modern control engineering*, vol. 17, no. 3. 1972.