

Perbandingan Algoritme RLS dan GM pada Multi Loop Quadrotor PD *Auto-tuning Loop Shaping*

Atikah Surriani¹, Adha Imam Cahyadi², Hanung Adi Nugroho³

Jurusan Teknik Elektro dan Teknologi Informasi
Fakultas Teknik Universitas Gadjah Mada
Jalan Grafika 2, Yogyakarta 55281

E-mail: ¹atikah.surriani.sie13@mail.ugm.ac.id, ²masimam@jteti.gadjahmada.edu, ³adinugroho@ugm.ac.id

Abstract—In this paper, PD auto-tuning controller based on loop shaping is designed to achieve stable output of the quadrotor's attitude and altitude under unknown parameters. We compare the results of two estimation algorithms namely Recursive least Squares (RLS) algorithm and Gradient Method (GM). We present the responses of each algorithm toward attitude and altitude of quadrotor and analyze it using Integral of Time Absolute Error (ITAE) to measure the effectiveness adaptive law performance. Regarding to the ITAE values, it shows that RLS algorithm has error relative to time value lower than GM algorithm, however yaw angle has higher value.

Abstrak— Pada paper ini, pengendali PD *auto-tuning loop shaping* dirancang untuk mendapatkan keluaran yang stabil dari *attitude* dan *altitude* quadrotor pada parameter yang tidak diketahui. Kami membandingkan hasil dari dua algoritme estimasi yaitu algoritme *Recursive Least Square* (RLS) dan *Gradient Method* (GM). Kami menampilkan tanggapan-tanggapan dari setiap algoritme terhadap *attitude* dan *altitude* (ketinggian) quadrotor dan menganalisis tanggapan tersebut dengan menggunakan *Integral of Time Absolute Error* (ITAE) untuk memperlihatkan kinerja dari algoritme. Berdasarkan nilai ITAE, menunjukkan bahwa algoritme RLS memiliki kesalahan relative terhadap waktu yang lebih rendah dibandingkan GM, kecuali pada sudut *yaw* yang memiliki nilai yang lebih tinggi.

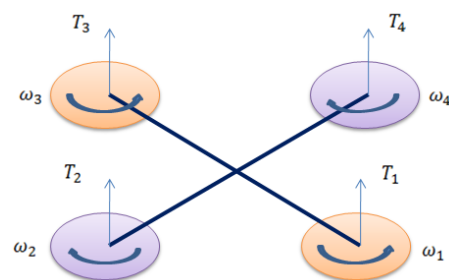
Kata kunci - Quadrotor, *Auto-tuning*, *Loop Shaping*, *Recursive least Squares*, *Gradient Method*, *Kendali Attitude & Altitude*.

I. PENDAHULUAN

Wahana udara tanpa awak atau yang dikenal dengan *Unmanned Aerial Vehicle* (UAV), menjadi objek penelitian yang populer dalam beberapa dekade ini. Hal ini dikarenakan UAV dapat diimplementasikan dalam berbagai bidang diantaranya keperluan pertanian untuk mengetahui data komposisi tanah dan air, militer [1], kebutuhan sipil [2], meteorologi, dan SAR [3]. Terdapat beberapa jenis UAV diantaranya adalah *fixed wing* yaitu wahana udara tanpa awak yang memiliki prinsip kerja yang sama dengan pesawat penumpang yang memiliki sayap dan ekor, kemudian *rotor craft* yaitu wahana udara tanpa awak yang bergerak dengan *multi* rotor, diantaranya helikopter yang memiliki satu rotor utama dan rotor, quadcopter atau quadrotor dengan empat rotor, *hexagon* dengan enam rotor, serta *pentagon* dengan delapan rotor [1].

UAV yang memiliki banyak rotor seperti quadrotor banyak menjadi objek penelitian dikarenakan memiliki beberapa kelebihan, diantaranya kemampuan *Vertical Take Off and Landing* (VTOL) [1][4] disamping *fixed wing aircraft* yang memerlukan landasan udara untuk terbang atau mendarat, kemampuan *hovering* atau melayang tetap diudara diatas target, dapat terbang di area tertutup, berbahaya, dan tidak dapat dijangkau manusia dan memiliki kemampuan untuk bermanuver baik [5].

Kelebihan-kelebihan quadrotor tersebut dikarenakan konfigurasi dari quadrotor yang bergerak dengan menggunakan 4 rotor yang terdiri dari sepasang rotor T_2 - T_4 berputar searah jarum jam dan sepasang rotor T_2 - T_4 berputar berlawanan arah jarum jam sebagaimana ditunjukkan pada Gambar 1, dan memiliki 6 derajat kebebasan (*Degree of Freedom*).



Gambar 1. Multi Rotor dari Quadrotor

Konfigurasi tersebut memperlihatkan bahwa quadrotor merupakan sistem yang *Multiple Input Multiple Output* (MIMO). Dengan 4 masukan dan 6 keluaran sehingga terjadi-nya *coupling* pada quadrotor tidak dapat dihindarkan [6], dan menjadikan quadrotor sebagai sistem yang *under-actuated*. Sehingga quadrotor adalah sistem yang tidak linier, rentan terhadap ketidakstabilan, dan sulit dikendalikan [7].

Secara umum quadrotor dikendalikan oleh pengendali PID. Namun metode pengendali PID tersebut membutuhkan banyak keterlibatan operator dalam menentukan nilai *gain* dari masing-masing koefisien K_p , K_i , dan K_d [8]. Pada dinamika pergerakan quadrotor dan medan lingkungan terbang quadrotor yang berubah-ubah, pengendali PID konvensional memiliki kemampuan terbatas. Kemampuan *auto-tuning* atau *self-tuning*

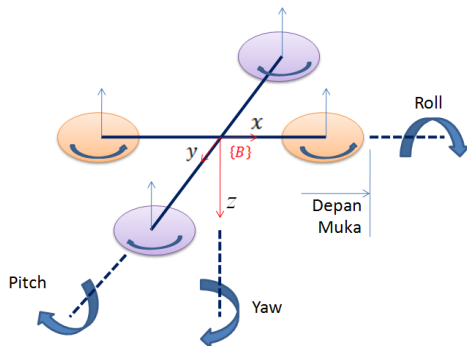
menjadi salah satu solusi dalam mengatasi masalah pada kondisi *coupling*, sistem *under-actuated*, dan tidak linier pada quadrotor.

Pada penelitian sebelumnya, dilakukan perancangan system pengendali PD *auto-tuning loop shaping* dengan menggunakan algoritme *Recursive Least Squares (RLS)* [9]. Penelitian ini mengendalikan ketinggian dari quadrotor yang didekati dengan fungsi alih orde 2. Fungsi alih orde 2 tersebut memiliki efek redaman *aerodynamic* yang rendah, sehingga cukup dikendalikan menggunakan pengendali PD lebih sederhana dibandingkan pengendali PID.

Penelitian ini merupakan pengembangan dari penelitian [9], yaitu penelitian mengenai pengendali PD *auto-tuning loop shaping* dengan menggunakan algoritme RLS. Pengembangan dari penelitian tersebut adalah dengan menambahkan peubah yang akan dikendalikan yaitu *roll*, *pitch* dan *yaw* atau *attitude* quadrotor dan juga ketinggian dari quadrotor. Perancangan sistem kendali ini juga mempertimbangkan *trajectory* atau lintasan dari quadrotor. Keluaran quadrotor pada sumbu-x dan sumbu-y dipengaruhi oleh sudut *roll* dan *pitch* quadrotor, sehingga model sistem terbagi menjadi *outer loop* dan *inner loop*. Pada *inner loop* sistem diimplementasikan pengendali *auto-tuning loop shaping*. Model sistem pengendali tersebut berdasarkan model Corke [3] yang telah dianalisis *controllability* dan *observability*-nya dari model tersebut dalam penelitian [10]. Selain *Recursive Least Squares*, kami mengimplementasikan *Gradient Method (GM)* pada penala pengendali, dan menganalisis tanggapan sistem terhadap kedua algoritme tersebut dengan menggunakan *Integral of Time Absolute Error (ITAE)*.

II. MODEL SISTEM QUADROTOR

Dasar model sistem quadrotor yang dideskripsikan oleh Corke [3]. Konfigurasi sistem quadrotor yang dimodelkan pada Gambar 2, quadrotor memiliki rotor pada masing-masing ujung silangnya. Koordinat dari *body frame* quadrotor memiliki sumbu-z yang menghadap kebawah yang mengikuti aturan *aerospace*, sehingga ketika quadrotor terbang menjauhi bumi, koordinat pada sumbu-z akan bernilai negatif.



Gambar 2. Konfigurasi Sistem Quadrotor

Model ini menggunakan 12 keadaan $\eta = [x\ y\ z\ \psi_R\ \psi_P\ \psi_Y\ \dot{x}\ \dot{y}\ \dot{z}\ \dot{p}\ \dot{q}\ \dot{r}]$ dengan $\eta_1 = [x\ y\ z]$ adalah koordinat quadrotor terhadap sumbu-x, sumbu-y,

dan sumbu-z pada kerangka global sedangkan $\eta_2 = [\psi_R\ \psi_P\ \psi_Y]$ adalah tiga sudut Euler yang membentuk pergerakan *roll*, *pitch*, *yaw* pada masing-masing sumbu-x, sumbu-y, dan sumbu-z, kecepatan linearnya berdasarkan posisi pada masing-masing sumbu-x, sumbu-y, dan sumbu-z $\eta_3 = [\dot{x}\ \dot{y}\ \dot{z}]$, kemudian kecepatan sudutnya berdasarkan sudut Euler pada masing-masing sumbu-x, sumbu-y, dan sumbu-z adalah $\eta_4 = [\dot{p}\ \dot{q}\ \dot{r}]$.

Peubah keadaan dan masukan sistem ditentukan sebagai berikut,

$$x = [\eta_1^T\ \eta_2^T\ \eta_3^T\ \eta_4^T]^T \tag{1}$$

$$u = [\omega_1^2\ \omega_2^2\ \omega_3^2\ \omega_4^2]^T \tag{2}$$

sehingga,

$$x = [\eta_1, \dots, \eta_{12}]^T \tag{3}$$

$$u = [u_1, \dots, u_4]^T \tag{4}$$

Dengan koordinat posisi $\eta_1 = [x\ y\ z] \in \mathbb{R}^3$, gaya yang bekerja pada quadrotor didapat berdasarkan hukum Newton kedua,

$$F = m\dot{v} + (\omega \times mv) \tag{5}$$

dengan m adalah massa quadrotor, $\omega = \eta_4 = [\dot{p}\ \dot{q}\ \dot{r}]$ adalah kecepatan sudut dan $v = \eta_3 = [\dot{x}\ \dot{y}\ \dot{z}]$ adalah kecepatan linier. Dari Gambar 2. Menunjukkan bahwa gaya yang bekerja pada quadrotor adalah gaya tarik gravitasi bumi dan gaya dorong total dari quadrotor,

$$F = F_g - F_{dorong} \tag{6}$$

Maka persamaan (6) menjadi

$$F = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - {}^W R_B \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{7}$$

Dengan g adalah percepatan gravitasi. Persamaan (7) disubstitusikan kedalam persamaan (5), maka didapat

$$\dot{v} = \frac{1}{m} \left[mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - {}^W R_B \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \right] - \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{8}$$

${}^W R_B$ adalah matrik rotasi dari *body frame* ke kerangka global,

$${}^W R_B = \begin{bmatrix} c_{\psi_P} c_{\psi_Y} & s_{\psi_R} s_{\psi_P} c_{\psi_Y} - c_{\psi_R} s_{\psi_Y} & c_{\psi_R} s_{\psi_P} c_{\psi_Y} + s_{\psi_R} c_{\psi_Y} \\ c_{\psi_P} s_{\psi_Y} & s_{\psi_R} s_{\psi_P} s_{\psi_Y} + c_{\psi_R} c_{\psi_Y} & c_{\psi_R} s_{\psi_P} s_{\psi_Y} - s_{\psi_R} c_{\psi_Y} \\ s_{\psi_P} & s_{\psi_R} c_{\psi_P} & c_{\psi_R} c_{\psi_P} \end{bmatrix} \tag{9}$$

Dengan $c_{\psi_R} = \cos \psi_R$, $s_{\psi_R} = \sin \psi_R$, $c_{\psi_P} = \cos \psi_P$, $s_{\psi_P} = \sin \psi_P$, $c_{\psi_Y} = \cos \psi_Y$, $s_{\psi_Y} = \sin \psi_Y$. Maka percepatan \dot{v} dapat ditulis sebagai,

$$\dot{v} = \frac{1}{m} \begin{bmatrix} -T(c_{\psi_R} s_{\psi_P} c_{\psi_Y} + s_{\psi_R} s_{\psi_Y}) \\ -T(c_{\psi_R} s_{\psi_P} s_{\psi_Y} - s_{\psi_R} c_{\psi_Y}) \\ mg - T(c_{\psi_R} c_{\psi_P}) \end{bmatrix} - {}^A R_B \begin{bmatrix} (\dot{q}\dot{z} - \dot{r}\dot{y}) \\ \dot{r}\dot{x} - \dot{p}\dot{z} \\ \dot{p}\dot{y} - \dot{q}\dot{x} \end{bmatrix} \tag{10}$$

Pada analisis rotasi, jika torsi dari η_2 , adalah

$$\tau = \begin{bmatrix} \tau_R \\ \tau_P \\ \tau_Y \end{bmatrix} \tag{11}$$

dan moment inersia dari quadrotor adalah,

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}. \quad (12)$$

Menggunakan hukum rotasi *rigid body*, maka torsi dinyatakan sebagai berikut,

$$\boldsymbol{\tau} = \mathbf{I}\dot{\boldsymbol{\omega}} + (\boldsymbol{\omega} + \mathbf{I}\boldsymbol{\omega}) \quad (13)$$

Persamaan (13) dapat dijabarkan menjadi,

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \begin{bmatrix} \tau_R \\ \tau_P \\ \tau_Y \end{bmatrix} - \left[\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times \begin{bmatrix} I_x \dot{p} \\ I_y \dot{q} \\ I_z \dot{r} \end{bmatrix} \right] \quad (14)$$

Maka, hubungan torsi $\boldsymbol{\tau}$ dengan kecepatan rotor dari quadrotor dapat direpresentasikan dengan menggunakan matriks Λ , yaitu

$$\Lambda = \begin{bmatrix} 0 & -db & 0 & db \\ db & 0 & -db & 0 \\ k & -k & k & -k \\ b & b & b & b \end{bmatrix}. \quad (15)$$

Sehingga didapat hubungan sebagai berikut,

$$\tau_R = db(u_4 - u_2) \quad (16)$$

$$\tau_P = db(u_1 - u_3) \quad (17)$$

$$\tau_Y = k(u_1 - u_2 + u_3 - u_4) \quad (18)$$

$$T = b(u_1 + u_2 + u_3 + u_4) \quad (19)$$

Konstanta b adalah konstanta daya angkat relative terhadap massa jenis udara, sedangkan d merupakan jarak antara motor terhadap titik pusat massa quadrotor dan k adalah konstanta pergeseran.

Model dinamis dari quadrotor berdasarkan translasi posisi $\eta_1 = [x \ y \ z]^T$ adalah kecepatan linier

$$\mathbf{v} = \boldsymbol{\eta}_3 = [\dot{x} \ \dot{y} \ \dot{z}]^T \quad (20)$$

dan percepatan linier $\dot{\mathbf{v}}$ diturunkan dari persamaan (10) dengan mengasumsikan bahwa $\boldsymbol{\omega} = \boldsymbol{\eta}_4 = [\dot{p} \ \dot{q} \ \dot{r}]^T \approx 0$ dan $\mathbf{v} = \boldsymbol{\eta}_3 = [\dot{x} \ \dot{y} \ \dot{z}]^T \approx 0$, sehingga

$$\dot{\mathbf{v}} = \dot{\boldsymbol{\eta}}_3 = [\ddot{x} \ \ddot{y} \ \ddot{z}]^T. \quad (21)$$

Persamaan (10) dapat ditulis sebagai,

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \frac{T}{m} \begin{bmatrix} (c_{\psi_R} s_{\psi_P} c_{\psi_Y} + s_{\psi_R} s_{\psi_Y}) \\ (c_{\psi_R} s_{\psi_P} s_{\psi_Y} - s_{\psi_R} c_{\psi_Y}) \\ (c_{\psi_R} c_{\psi_P}) \end{bmatrix}. \quad (22)$$

Model dinamis quadrotor berdasarkan rotasi $\eta_2 = [\psi_R \ \psi_P \ \psi_Y]^T$ adalah kecepatan sudut

$$\boldsymbol{\omega} = \boldsymbol{\eta}_4 = [\dot{p} \ \dot{q} \ \dot{r}]^T \quad (23)$$

dan percepatan sudut berdasarkan persamaan (14) maka $\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\eta}}_4 = [\ddot{p} \ \ddot{q} \ \ddot{r}]^T$ didapat,

$$\ddot{p} = \frac{db}{I_x} (u_4 - u_2) - \frac{(I_z - I_y)}{I_x} \dot{q}\dot{r} \quad (24)$$

$$\ddot{q} = \frac{db}{I_y} (u_1 - u_3) - \frac{(I_x - I_z)}{I_y} \dot{p}\dot{r} \quad (25)$$

$$\ddot{r} = \frac{k}{I_z} (u_1 - u_2 + u_3 - u_4) - \frac{(I_y - I_x)}{I_z} \dot{p}\dot{q}. \quad (26)$$

Linearisasi sistem dan parameter yang diatur menggunakan parameter yang digunakan dalam penelitian yang telah dilakukan sebelumnya [9][10] menggunakan metode linearisasi Jacobi. Dalam metode Jacobi, matriks \mathbf{A} dan \mathbf{B} menjadi derivatif parsial dari model non linear.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_{12}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{12}}{\partial x_1} & \dots & \frac{\partial f_{12}}{\partial x_{12}} \end{bmatrix} \quad (27)$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_{12}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{12}}{\partial u_1} & \dots & \frac{\partial f_{12}}{\partial u_{12}} \end{bmatrix}. \quad (28)$$

Representasi dari persamaan keadaan dan keluaran sistem ditunjukkan sebagai berikut,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (29)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (30)$$

Setelah model non linear disubstitusi pada titik kesetimbangan di dapat matriks \mathbf{A} , \mathbf{B} , \mathbf{C} dan \mathbf{D} sebagai berikut,

$$\mathbf{A}_{12 \times 12} = \begin{bmatrix} \mathbf{O}_{6 \times 6} & \mathbf{I}_{6 \times 6} \\ \mathbf{G}_{6 \times 6} & \mathbf{O}_{6 \times 6} \end{bmatrix} \quad (31)$$

$$\mathbf{B}_{12 \times 4} = \begin{bmatrix} \mathbf{O}_{8 \times 4} \\ \mathbf{M}_{4 \times 4} \end{bmatrix} \quad (32)$$

$$\mathbf{C} = [\mathbf{I}_{1 \times 12}] \quad (33)$$

dan

$$\mathbf{D} = \mathbf{0} \quad (34)$$

dengan matriks \mathbf{O} merupakan matrik nol dan \mathbf{I} adalah matriks identitas. Matriks \mathbf{G} dan \mathbf{M} dapat dituliskan sebagai,

$$\mathbf{G}_{6 \times 6} = \begin{bmatrix} 0 & 0 & 0 & 0 & -g & 0 \\ 0 & 0 & 0 & g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (35)$$

$$\mathbf{M}_{4 \times 4} = \begin{bmatrix} -\mu_1 & -\mu_1 & -\mu_1 & -\mu_1 \\ 0 & -\mu_2 & 0 & \mu_2 \\ \mu_3 & 0 & -\mu_3 & 0 \\ \mu_4 & -\mu_4 & \mu_4 & -\mu_4 \end{bmatrix}. \quad (36)$$

Konstanta yang digunakan dalam penelitian ini adalah sebagai berikut,

$$g = 9,81 \text{ m/s}^2$$

$$\mu_1 = 6,56 \times 10^{-5} \text{ m}$$

$$\mu_2 = 1,001 \times 10^{-3} \text{ m}$$

$$\mu_3 = 1,001 \times 10^{-3} \text{ m}$$

$$\mu_4 = 1,392 \times 10^{-6} \text{ m}.$$

Persamaan keadaan serta persamaan keluaran dari sistem adalah,

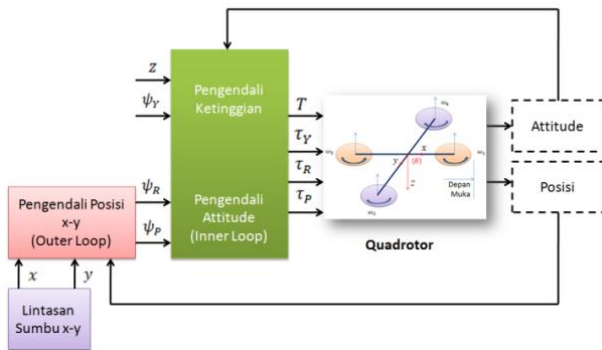
$$\dot{\mathbf{x}} = \mathbf{A}_{12 \times 12}\mathbf{x} + \mathbf{B}_{12 \times 4}\mathbf{u} \quad (37)$$

$$\mathbf{y} = \mathbf{C}_{1 \times 12}\mathbf{x}. \quad (38)$$

III. RANCANGAN SISTEM KENDALI

A. Sistem Kendali *Multi Loop*

Sistem pengendali ini terbagi menjadi *inner loop* dan *outer loop* sebagaimana model quadrotor [3]. Diagram blok rancangan sistem kendali *multi loop* ditunjukkan oleh Gambar 3.



Gambar 3. Diagram blok rancangan sistem kendali

Berdasarkan Gambar 3, pada *outer loop* mengendalikan posisi lintasan dalam sumbu-x dan sumbu-y dari quadrotor dengan pengendali PD. Masukan dari *outer loop* adalah posisi $[x \ y]$ quadrotor pada sumbu-x dan sumbu-y, keluaran dari *outer loop* tersebut adalah sudut *roll* dan *pitch* $[\psi_R \ \psi_P]$. Hal tersebut dikarenakan sudut *roll* dan *pitch* $[\psi_R \ \psi_P]$ ter-coupling terhadap posisi $[x \ y]$. Selisih dari posisi $[x \ y]$ dan sudut *roll* dan *pitch* $[\psi_R \ \psi_P]$ aktual menjadi keluaran dari *outer loop* agar dapat menghasilkan gaya untuk mendapatkan posisi $[x \ y]$ sesuai dengan yang diinginkan.

Melalui *outer loop* ini rancangan sistem kendali ini mempertimbangkan posisi dari quadrotor. Kesalahan (*error*) posisi sumbu-x dan sumbu-y dihitung dalam *frame* global dan dirotasikan berdasarkan sudut *yaw* pada *frame* $\{A\}$. *Frame* $\{A\}$ adalah *frame* yang diberikan pada quadrotor dengan nilai koordinat orisinal yang sama dengan *body frame*, namun sumbu-x dan sumbu-y paralel dengan bumi. Pengendalian posisi dilakukan dengan menggunakan kecepatan pada *frame* $\{A\}$. Berdasarkan selisih kesalahan antara data referensi posisi dan posisi aktual, maka kecepatan yang diinginkan dalam *frame* $\{A\}$ adalah

$${}^A\mathbf{v} = {}^A\mathbf{R}_0(\psi_Y)\mathbf{v} = {}^0\mathbf{R}_A^T(\psi_Y)\mathbf{v} \quad (39)$$

Sehingga,

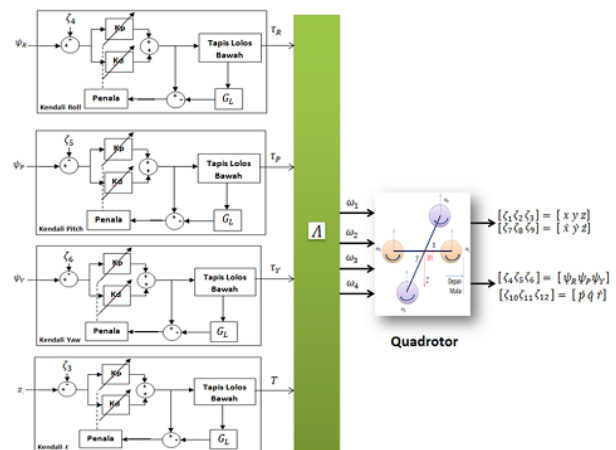
$$\begin{bmatrix} {}^A v_x \\ {}^A v_y \end{bmatrix} = {}^0\mathbf{R}_A^T(\psi_Y) \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \quad (40)$$

Dengan ${}^A\mathbf{R}_0$ adalah matrik rotasi dari kerangka global ke *frame* $\{A\}$, ${}^0\mathbf{R}_A^T$ adalah matrik transformasi rotasi, ${}^A v_x$ adalah kecepatan yang diinginkan di sumbu-x, dan ${}^A v_y$ adalah kecepatan yang diinginkan di sumbu-y.

Jika quadrotor akan bergerak pada arah v_x , untuk mendapatkan posisi yang diinginkan dapat mengkomputasi nilai kecepatan yang tepat. Kemudian sudut *pitch* (ψ_P) akan menghasilkan gaya yang

menggerakkan quadrotor tersebut. Perlu diperhatikan, pergerakan dari v_x membutuhkan rotasi yang bernilai negatif dari sumbu-y (ψ_P), sedangkan v_y membutuhkan rotasi positif dari sumbu-x (ψ_R) yang disesuaikan dengan aturan koordinat berdasarkan konfigurasi quadrotor pada Gambar 2 [3].

Pada Gambar 3, menunjukkan pengendali PD *auto-tuning loop shaping* diimplementasikan pada *inner loop* sistem. Masukan dari *inner loop* adalah $[z \ \psi_R \ \psi_P \ \psi_Y]$ dan keluaran dari pengendali adalah $[\tau_R \ \tau_P \ \tau_Y \ T]$. Ketinggian z adalah peubah yang tidak dipengaruhi oleh peubah lain, sehingga z menjadi masukan dari *inner loop* tanpa melewati *outer loop*, sebagaimana $[x \ y]$. Hal ini dikarenakan posisi z hanya dipengaruhi oleh kecepatan rotor untuk menggerakkan quadrotor keatas atau kebawah terhadap sumbu-z sebagaimana dalam persamaan (18). Berbeda dengan sudut *roll* dan sudut *pitch* $[\psi_R \ \psi_P]$ adalah keluaran dari *outer loop* yang berhubungan dengan posisi $[x \ y]$ dan pergerakannya dipengaruhi oleh sudut *yaw* sebagaimana direpresentasikan pada persamaan (39) dan (40). Diagram blok dari *inner loop* ditunjukkan pada Gambar 4.



Gambar 4. *Inner loop* rancangan sistem kendali

Dari Gambar 4, *inner loop* menggunakan pengendali PD yang ditala secara otomatis seperti pada [9]. Isyarat kendali $o_i(t)$, merupakan keluaran dari pengendali PD yang diberikan oleh persamaan,

$$o_i(t) = Kp_i(t)e_i(t) + Kd_i(t)\frac{de_i(t)}{dt} \quad (41)$$

Dengan $i \in \{\tau_R, \tau_P, \tau_Y, T\}$ dan $e_i(t)$ adalah selisih nilai antara nilai referensi dan keluaran sistem. Masukan terhadap penala PD berupa isyarat kendali $\varphi_i = [\varepsilon_i \ (\frac{d\varepsilon_i}{dt})^T]$ dan keluaran dari penala PD adalah berupa konstanta proporsional dan derivatif $q_i = [Kp_i \ Kd_i]^T$.

Blok tapis pada Gambar 4 digunakan untuk memberikan kompensasi dari pengaruh yang disebabkan oleh integrator orde dua. Sehingga dalam penelitian ini digunakan sebuah penyeimbang dalam bentuk tapis lolos bawah. Fungsi alih, $N(s)$, dari tapis lolos bawah adalah,

$$N(s) = \frac{s}{\rho(s^2+2s+1)} \quad (42)$$

Dengan $\rho = 0,25$.

Pada *loop shaping* pengendali dirancang sedemikian rupa sehingga karakteristik kalang terbuka mengikuti karakteristik tanggapan frekuensi yang diinginkan $G_L(s)$ [8]. Tujuan dari teknik *loop shaping* ini adalah untuk mendapatkan perbedaan yang sekecil mungkin antara isyarat keluaran yang sebenarnya dan isyarat keluaran yang dari kalang yang diinginkan. Fungsi objektif dasar pada *loop shaping* ini adalah meminimalkan fungsi nonkonveks pada persamaan berikut,

$$\min_{\{K_p, K_d\}} = \left\| \frac{G_L u - G_p G_K u}{1 + G_p G_K} \right\|_2 \quad (43)$$

G_L adalah target *loop shaping*, G_p adalah *plant*, dan G_K pengendali PD.

B. Algoritme Estimasi K_p dan K_d

Metode estimasi pertama, *Recursive Least Squares* (RLS) adalah salah satu algoritme yang digunakan untuk mengestimasi nilai K_p dan K_d dari penala pengendali PD [9]. Algoritme tersebut dapat didefinisikan,

$$\dot{\theta} = P \epsilon \varphi \quad (44)$$

$$\dot{P} = \beta P - P \varphi \varphi^T P. \quad (45)$$

Dengan P adalah matrik kovarian yang menentukan validasi nilai gain. Fungsi objektif (J_1) dari sistem ini ditunjukkan sebagai berikut [11]:

$$J_1 = \min_{\{K_p, K_d\}} \int_0^t (L(\tau) - K_p \epsilon(\tau) K_d \frac{d\epsilon(\tau)}{dt})^2 d\tau. \quad (46)$$

Metode estimasi kedua, *Gradient Method* (GM) memiliki persamaan sebagai berikut,

$$\dot{\theta} = -\Gamma \nabla J_2(\theta) \quad (47)$$

$$= \Gamma \epsilon \varphi.$$

Dengan $\Gamma = \Gamma^T > 0$ adalah matrik *scaling* yang berhubungan dengan *gain* adaptif dan Γ adalah matrik diagonal.

Metode ini memiliki fungsi objektif,

$$J_2 = \min_{\{K_p, K_d\}} \int_0^t \epsilon^2 d\tau. \quad (48)$$

Dengan kesalahan (ϵ) pada penala PD dari kedua metode tersebut, dinormalisasikan oleh persamaan,

$$\epsilon = q - \theta^T \varphi. \quad (49)$$

Pendekatan nilai q dapat dinotasikan sebagai,

$$\hat{q} = \theta_{(t)}^T \varphi \quad (50)$$

Pada sistem kendali *multi loop*, maka terdapat $\theta_1 = [P_z \ D_z]$ pada *altitude* (z), $\theta_2 = [P_{\psi_R} \ D_{\psi_R}]$ pada sudut *roll*, $\theta_3 = [P_{\psi_P} \ D_{\psi_P}]$ pada sudut *pitch*, $\theta_4 = [P_{\psi_Y} \ D_{\psi_Y}]$ pada sudut *yaw*, sehingga

$$\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]^T \quad (51)$$

dan

$$\varphi = [\epsilon_z \ \dot{\epsilon}_z \ \epsilon_{\psi_R} \ \dot{\epsilon}_{\psi_R} \ \epsilon_{\psi_P} \ \dot{\epsilon}_{\psi_P} \ \epsilon_{\psi_Y} \ \dot{\epsilon}_{\psi_Y}]^T \quad (52)$$

C. Integral of Time Absolute Error (ITAE).

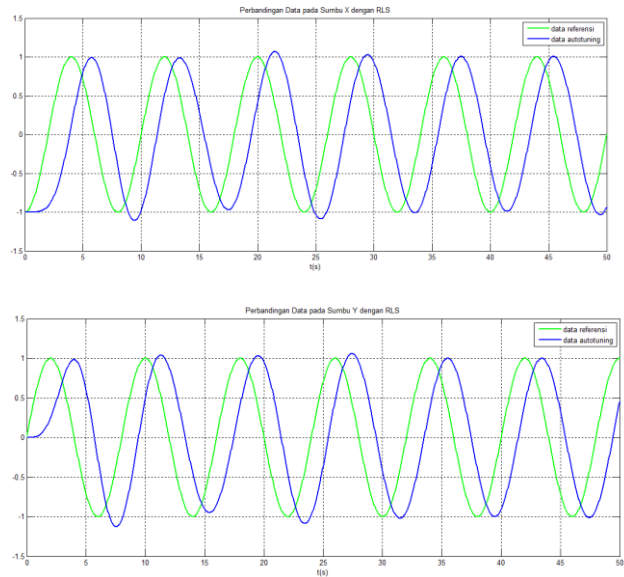
Integral of Time Absolute Error merupakan salah satu indeks parameter yang menghitung nilai kesalahan (*error*). ITAE memiliki persamaan sebagai berikut

$$ITAE = \int_0^{\infty} t |\epsilon(t)| dt \quad (53)$$

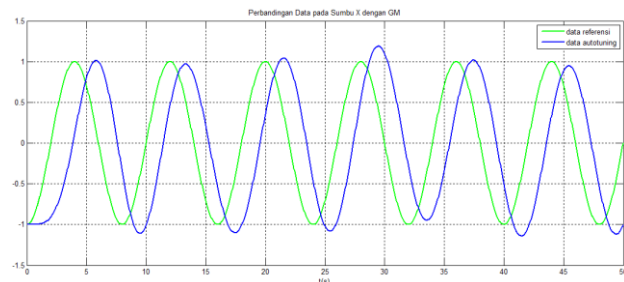
dengan t adalah waktu, sedangkan $\epsilon(t)$ adalah perbedaan antara data referensi dengan data keluaran dari pengendali [12]. Dalam penelitian ini, menggunakan ITAE untuk menganalisis kinerja dari pengendali PD *auto-tuning* terhadap parameter yang dikendalikan yaitu ketinggian dan *attitude* quadrotor, $[z \ \psi_R \ \psi_P \ \psi_Y]$. Analisis ITAE dilakukan karena masukan dan tanggapan dari *attitude* quadrotor adalah isyarat sinusoidal dan *coupling* dinamika system quadrotor dari posisi $[x \ y]$ dengan sudut *roll* dan *pitch* $[\psi_R \ \psi_P]$ yang rumit.

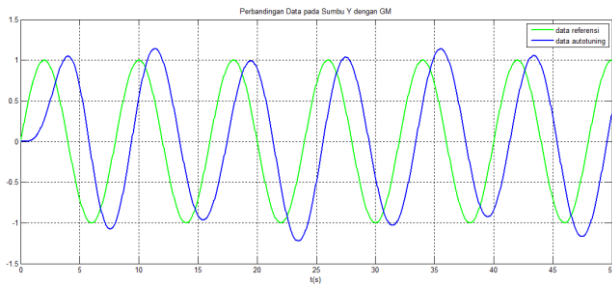
IV. SIMULASI DAN ANALISIS

Dari simulasi perancangan pengendali PD *auto-tuning loop shaping* dianalisis melalui tanggapan-tanggapan sistem.



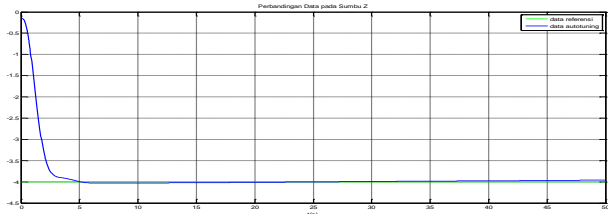
Gambar 5. Tanggapan posisi x dan y dengan algoritme RLS



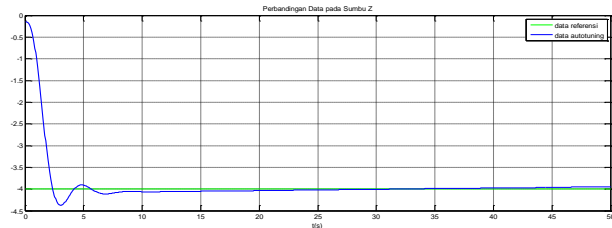


Gambar 6. Tanggapan posisi x dan y dengan algoritme GM

Gambar 5 menunjukkan tanggapan posisi [x y] dengan menggunakan algoritme RLS, Gambar 6 memperlihatkan tanggapan posisi [x y] dengan menggunakan algoritme GM. Berdasarkan Gambar 5 dan Gambar 6 memperlihatkan tanggapan posisi [x y] dari masing-masing algoritme dan terlihat bahwa dengan algoritme RLS memiliki tanggapan posisi yang lebih baik dibandingkan dengan menggunakan GM.

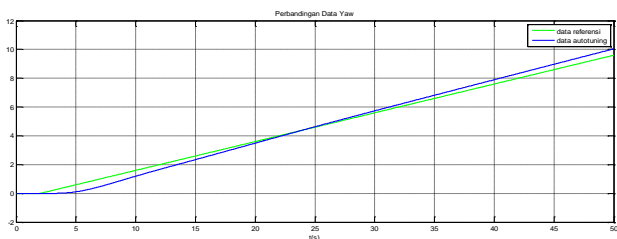


Gambar 7. Tanggapan Ketinggian Quadrotor dengan Algoritme RLS

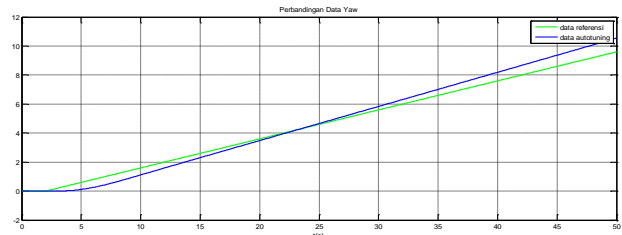


Gambar 8. Tanggapan Ketinggian Quadrotor dengan Algoritme GM

Pada Gambar 7 dan Gambar 8, memperlihatkan tanggapan ketinggian dengan menggunakan algoritme RLS dan GM. Tanggapan ketinggian dengan menggunakan algoritme RLS pada Gambar 7, memiliki tanggapan yang lebih baik dibandingkan dengan menggunakan algoritme GM pada Gambar 8. Terlihat bahwa dengan menggunakan algoritme RLS tanggapan ketinggian quadrotor tidak memiliki overshoot dan memiliki waktu tunak yang lebih cepat dibandingkan dengan menggunakan GM.

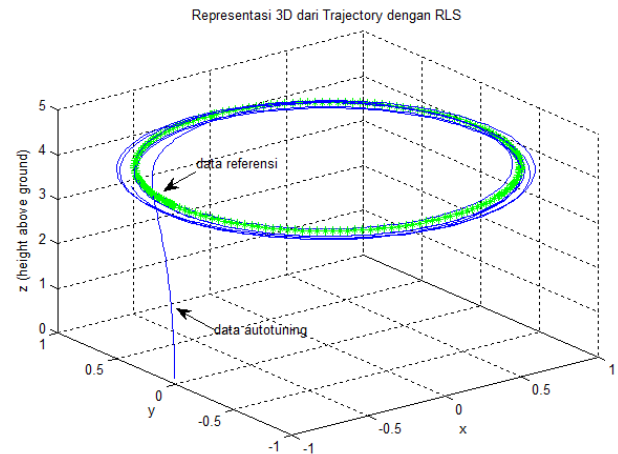


Gambar 9. Tanggapan yaw quadrotor dengan algoritme RLS

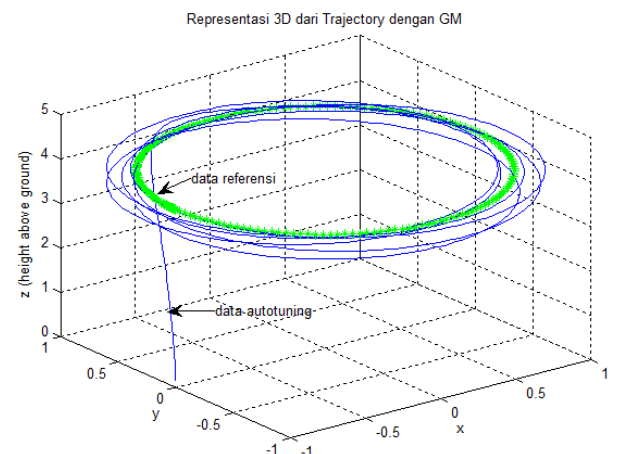


Gambar 10. Tanggapan yaw quadrotor dengan algoritme GM

Dari tanggapan yaw quadrotor dengan menggunakan algoritme RLS pada Gambar 9 dan tanggapan yaw quadrotor menggunakan GM terlihat bahwa, dengan algoritme RLS memberikan tanggapan yaw quadrotor yang lebih baik. Selanjutnya, pada Gambar 11 dan Gambar 12 memperlihatkan representasi posisi x,y,z dengan menggunakan algoritme RLS dan GM. Berdasarkan Gambar 11, algoritme RLS memberikan tanggapan posisi yang lebih baik dibandingkan GM pada Gambar 12.



Gambar 11. Posisi x,y,z Quadrotor dalam 3D dengan RLS



Gambar 12. Posisi x,y,z Quadrotor dalam 3D dengan GM

Hal tersebut bersesuaian dengan tanggapan masing-masing posisi x,y dan z terhadap algoritme RLS dan GM.

Sudut roll, pitch, yaw dan ketinggian dari sistem dianalisis dengan menggunakan ITAE. Tabel 1 memperlihatkan perbandingan ITAE dari RLS dan GM.

TABEL 1. ANALISIS ITAE

| ITAE | | | | |
|------------|---------|----------|----------|----------|
| Algoritme | Z | Yaw | Pitch | Roll |
| | ms^2 | $rads^2$ | $rads^2$ | $rads^2$ |
| RLS | 2602.1 | 6457.4 | 11249.5 | 11172.1 |
| GM | 11842.7 | 4048.4 | 11566.2 | 11969.9 |

Berdasarkan Tabel 1, nilai ITAE pada ketinggian quadrotor GM memiliki nilai yang lebih rendah dibandingkan RLS. Tanggapan sudut *pitch* dan *roll* juga memiliki nilai yang lebih rendah dibandingkan dengan menggunakan GM dibandingkan RLS. Pada nilai ITAE *yaw* dengan algoritme RLS lebih rendah. Perbedaan nilai ITAE dengan algoritme RLS dan GM dikarenakan komputasi yang berbeda dari kedua algoritme tersebut terhadap estimasi K_p dan K_d dari setiap *loop*. Algoritme RLS memiliki komputasi yang lebih akurat dibandingkan dengan GM sehingga pada waktu $t = 50s$, ITAE pada tanggapan *yaw* RLS lebih tinggi.

V. KESIMPULAN

Dua algoritme RLS dan GM telah diimplementasikan dalam pengendali PD *auto-tuning loop shaping* pada quadrotor. Pengendali PD *auto-tuning loop shaping* ini digunakan untuk mengendalikan *attitude* dan ketinggian dari quadrotor. Berdasarkan hasil simulasi, rancangan sistem kendali PD *auto-tuning loop shaping* dapat mengendalikan *attitude* dan *altitude* sistem quadrotor. Hal tersebut terlihat pada tanggapan *attitude* dan *altitude* sistem quadrotor. Selain itu, tanggapan posisi $[x\ y\ z]$ quadrotor berhasil dikendalikan mengikuti skenario lintasan yang diinginkan. Dari hasil analisis, diperoleh nilai ITAE untuk algoritme RLS pada *altitude* 2602,1 ms^2 , dan *attitude* 6457,4 $rads^2$, 11249,5 $rads^2$, 11172,1 $rads^2$ sedangkan untuk algoritme GM pada *altitude* 11842,7 ms^2 , dan *attitude* 4048,4 $rads^2$, 11566,2 $rads^2$, 11969.9 $rads^2$. Berdasarkan nilai ITAE, algoritme RLS, memiliki nilai kesalahan relatif terhadap waktu yang lebih rendah dibandingkan dengan algoritme GM kecuali pada bagian *attitude* di sudut *yaw*. Perbedaan nilai ITAE dengan algoritme RLS dan GM dikarenakan komputasi yang berbeda dari kedua algoritme tersebut terhadap estimasi K_p dan K_d dari masing-masing *loop*.

DAFTAR PUSTAKA

- [1] P. Ferrell, S. Member, B. Smith, B. Stark, Y. Chen, and S. Member, "Dynamic Flight Modeling of a Multi-Mode Flying Wing Quadrotor Aircraft," *Int. Conf. Unmanned Aircr. Syst.*, pp. 398–404, 2013.
- [2] H. Bolandi, M. Rezaei, and R. Mohsenipour, "Attitude Control of a Quadrotor with Optimized PID Controller," *Intell. Control Autom.*, vol. 2013, no. August, pp. 335–342, 2013.
- [3] P. Corke, *Robotics Vision and Control, Fundamental Algorithms in MATLAB*®, vol. 73. Springer Tracts in Advance Robotics, 2011.
- [4] S. Bouabdallah and R. Siegwart, "Design and Control of an Indoor Micro Quadrotor," *Proceeding 2004 IEEE Int. Conf. Robot. Autom.*, no. April, pp. 4393–4398, 2004.
- [5] A. Modirrousta and M. Khodabandeh, "Adaptive Robust Sliding Mode Controller Design for Disturbances," *Proceeding 2nd RSI/ISM Int. Conf. Robot. Mechatronics*, pp. 870–877, 2014.
- [6] M. G. Earl and R. D'Andrea, "Real-time attitude estimation techniques applied to a four rotor helicopter," *2004 43rd IEEE Conf. Decis. Control (IEEE Cat. No.04CH37601)*, vol. 4, pp. 3956–3961, 2004.
- [7] A. Das, K. Subbarao, and F. Lewis, "Dynamic inversion of quadrotor with zero-dynamics stabilization," *Proc. IEEE Int. Conf. Control Appl.*, pp. 1189–1194, 2008.
- [8] S. Gaikwad, S. Dash, and G. Stein, "Loop-Shaping Ideas," *Int. Conf. Control Appl. Proc. 1999 IEEE*, pp. 589–593, 1999.
- [9] H. Tnunay, M. Q. Abdurrohman, Y. Nugroho, R. Inovan, A. I. Cahyadi, and Y. Yamamoto, "Auto-Tuning Quadcopter Using Loop Shaping," *Int. Conf. Comput. Control. Informatics Its Appl.*, pp. 1–4, 2013.
- [10] A. Ataka, H. Tnunay, R. Inovan, M. Q. Abdurrohman, H. Prestianto, A. Cahyadi, and Y. Yamamoto, "Controllability and Observability Analysis of the Gain Scheduling Based Linearization for UAV," *Conf. Int. Syst. Intell. Comput.*, no. November, pp. 25–27, 2013.
- [11] P. A. Ioannou and J. Sun, "Robust Adaptive Control," *Cour. Dover Publ.*, 2013.
- [12] F. G. Martins, "Tuning PID Controllers using the ITAE Criterion *," vol. 21, no. 5, 2005.