Penerapan Quaternion dalam Stabilisasi Drone

Dzaky Aurelia Fawwaz - 13523065[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523065@std.stei.itb.ac.id , [2]dzakyaureliafawwaz@gmail.com,

***Abstract*— Makalah ini menyajikan implementasi sistem kendali orientasi quadrotor berbasis quaternion. Pendekatan quaternion dipilih untuk mengatasi masalah singularitas gimbal lock dan efisiensi komputasi dibanding representasi Euler angle konvensional. Sistem kendali menggunakan struktur Proportional-Integral-Derivative (PID) yang bekerja langsung dalam domain quaternion untuk menghasilkan respons yang stabil. Pengujian dilakukan dalam tiga skenario: *step response* untuk mengevaluasi karakteristik transien, *disturbance rejection* untuk menguji ketahanan terhadap gangguan eksternal, dan *trajectory tracking* untuk menilai kemampuan mengikuti referensi dinamis. Hasil menunjukkan sistem mampu mencapai orientasi yang diinginkan dengan cepat dan stabil, memiliki ketahanan yang baik terhadap gangguan, serta dapat melacak trajektori referensi dengan akurat.**

***Keywords*— quaternion, kendali orientasi, quadrotor, PID control, stabilisasi**

I. PENDAHULUAN

Perkembangan teknologi penerbangan tanpa awak (Unmanned Aerial Vehicle/UAV) telah mengalami kemajuan pesat dalam dekade terakhir. Salah satu platform yang mendapat perhatian khusus adalah quadrotor, sebuah wahana terbang dengan empat motor penggerak. Quadrotor menawarkan kemampuan manuver yang fleksibel seperti vertical take-off and landing (VTOL), hover, dan penerbangan multiarah yang menjadikannya ideal untuk berbagai aplikasi seperti pengawasan, pemetaan, dan inspeksi infrastruktur [1],

Namun, pengendalian quadrotor menghadapi tantangan signifikan karena karakteristik dinamikanya yang nonlinear dan coupled. Masalah utama terletak pada stabilisasi sikap (*attitude*) yang merupakan komponen kritis dalam pengendalian quadrotor. Pendekatan konvensional menggunakan representasi Euler angle memiliki beberapa keterbatasan fundamental seperti gimbal lock yang menyebabkan hilangnya satu derajat kebebasan ketika dua sumbu rotasi sejajar, kompleksitas komputasi yang tinggi karena perhitungan fungsi trigonometri yang intensif, serta masalah singularitas yang menyebabkan diskontinuitas pada representasi matematis.

Quaternion menawarkan solusi yang efektif untuk mengatasi keterbatasan tersebut. Sebagai bilangan hiperkompleks rank 4, quaternion memiliki keunggulan karena bebas dari masalah gimbal lock, komputasi lebih efisien tanpa fungsi trigonometri, representasi rotasi yang kontinu dan stabil, serta interpolasi rotasi yang lebih smooth dengan konsumsi memori lebih rendah. Hal ini menjadikan quaternion sebagai pilihan yang menjanjikan untuk implementasi sistem kendali quadrotor[2].

Makalah ini mengusulkan implementasi sistem kendali berbasis quaternion untuk stabilisasi sikap quadrotor. Kontribusi utama meliputi pemodelan sistematis dinamika quadrotor dalam ruang quaternion, perancangan kontroler P2 nonlinear dalam domain quaternion beserta optimasi parameternya, serta implementasi dan validasi melalui berbagai skenario pengujian. Analisis komprehensif juga dilakukan untuk mengevaluasi stabilitas sistem, ketahanan terhadap gangguan, serta rekomendasi implementasi.

II. LANDASAN TEORI

A. Quartenion

Quaternion adalah bilangan hiperkompleks rank 4 yang direpresentasikan dalam bentuk bagian skalar dan vector [3]. Bagian skalar q0 merepresentasikan besarnya rotasi, sedangkan bagian vektor (q1, q2, q3) menunjukkan sumbu rotasi dalam ruang 3D. Representasi ini memungkinkan deskripsi rotasi yang lebih efisien dibanding Euler angles.

$$z = a + bi + cj + dk$$

Gambar 1 . Representasi Quaternion
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-25-Aljabar-Quaternion-Bagian1-2023.pdf

Operasi perkalian quaternion (Kronecker product) merupakan operasi fundamental yang menggabungkan dua rotasi. Sifat tidak komutatif pada perkalian quaternion sesuai dengan sifat rotasi di ruang 3D. Hasil perkalian dua quaternion menghasilkan quaternion baru yang merepresentasikan kombinasi kedua rotasi tersebut.

- Misalkan

$$z_1 = a_1 + b_1 i + c_1 j + d_1 k$$
$$z_2 = a_2 + b_2 i + c_2 j + d_2 k$$

- Kalikan keduanya:

$$\begin{aligned} z_1 z_2 &= (a_1 + ib_1 + jc_1 + kd_1)(a_2 + ib_2 + jc_2 + kd_2) \\ &= a_1 a_2 + ia_1 b_2 + ja_1 c_2 + ka_1 d_2 \\ &\quad + ib_1 a_2 + i^2 b_1 b_2 + ijb_1 c_2 + ikb_1 d_2 \\ &\quad + jc_1 a_2 + jic_1 b_2 + j^2 c_1 c_2 + jkc_1 d_2 \\ &\quad + kd_1 a_2 + kid_1 b_2 + kjd_1 c_2 + k^2 d_1 d_2. \end{aligned}$$

Panjang quaternion didefinisikan sebagai akar kuadrat dari jumlah kuadrat semua komponennya.

$$\text{Quaternion: } q = a + bi + cj + dk$$

$$\text{Magnitude: } \|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}$$

Gambar 3 . Panjang Quaternion
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-25-Aljabar-Quaternion-Bagian1-2023.pdf

Untuk aplikasi rotasi, quaternion harus dinormalisasi sehingga memiliki panjang unit (unit quaternion). Normalisasi ini penting untuk menjaga konsistensi transformasi rotasi.

$$\hat{q} = \frac{1}{\|q\|}(a + bi + cj + dk)$$

Gambar 4 . Unit Satuan Quaternion atau normalisasi
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-25-Aljabar-Quaternion-Bagian1-2023.pdf

Operasi konjugat pada quaternion mengikuti konsep yang mirip dengan bilangan kompleks, dimana tanda dari semua komponen imajiner (bagian vektor) dibalik sementara bagian real (skalar) tetap.

$$q\bar{q} = a^2 + b^2 + c^2 + d^2$$

Gambar 5 . Konjugat Quaternion
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-25-Aljabar-Quaternion-Bagian1-2023.pdf

Invers quaternion merupakan operasi untuk mendapatkan quaternion yang ketika dikalikan dengan quaternion asli menghasilkan quaternion identitas [3]. Berbeda dengan konjugat yang hanya membalik tanda vektor, invers mempertimbangkan juga norma quaternion. Untuk kasus umum, invers quaternion didapatkan dengan membagi konjugat quaternion dengan kuadrat normanya. Khusus untuk unit quaternion (quaternion dengan panjang 1), inversnya sama dengan konjugatnya karena normanya bernilai 1. Konsep invers quaternion sangat penting dalam transformasi rotasi, khususnya untuk melakukan rotasi balik atau mengembalikan orientasi ke posisi awal.

$$q^{-1} = \frac{1}{q} = \frac{\bar{q}}{\|q\|^2}$$

Gambar 6 Invers Quaternion
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-25-Aljabar-Quaternion-Bagian1-2023.pdf

Kemudian juga terdapat Turunan Quaternion. Turunan quaternion terhadap waktu berkaitan dengan kecepatan sudut angular. Persamaan ini penting dalam kinematika untuk menghitung perubahan orientasi berdasarkan kecepatan sudut. Implementasi menggunakan left-hand notation memerlukan perhatian khusus pada konjugasi quaternion kecepatan sudut.

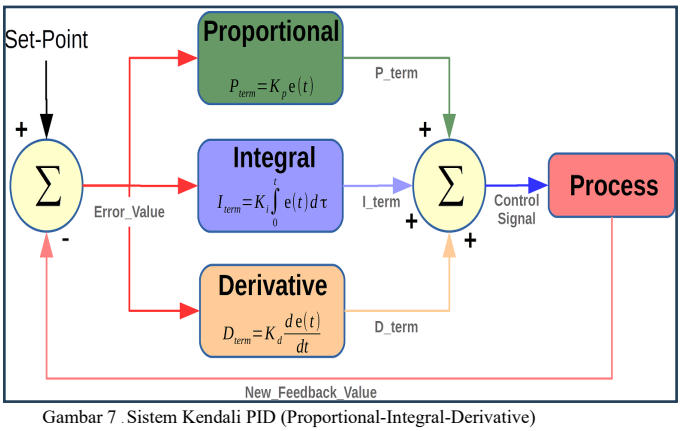
B. Sistem Kendali PID (Proportional-Integral-Derivative)

Sistem PID bekerja dalam closed loop dimana output sistem (New_Feedback_Value) dibandingkan dengan nilai yang diinginkan (Set-Point) untuk menghasilkan error [2]. Error ini kemudian diproses oleh ketiga komponen PID untuk menghasilkan Control Signal yang sesuai. Control Signal ini merupakan kombinasi linear dari ketiga komponen.

u(t) = Pterm + Iterm + Dterm.

Proses ini berlangsung secara kontinyu untuk memastikan sistem tetap stabil dan mengikuti setpoint yang diinginkan [4].

Optimasi kontroler PID melibatkan penyesuaian ketiga parameter gain (Kp, Ki, Kd) untuk mendapatkan performa yang diinginkan. Proses tuning mempertimbangkan *trade-off* antara kecepatan respon, stabilitas, dan robustness terhadap gangguan. Metode tuning dapat dilakukan secara empiris atau menggunakan metode sistematis [4].



Gambar 7 . Sistem Kendali PID (Proportional-Integral-Derivative)
Sumber:https://www.theengineeringconcepts.com/pid-controller/#google_vignette

Kontrol proportional memberikan aksi kontrol yang sebanding dengan error saat ini. Semakin besar error, semakin besar pula aksi kontrol yang diberikan. Gain proportional (Kp) menentukan kekuatan respon sistem

terhadap error. Nilai Kp yang terlalu besar dapat menyebabkan sistem berosilasi, sementara nilai yang terlalu kecil membuat respon sistem lambat [4].

Kontroler integral mengakumulasi error sepanjang waktu untuk menghilangkan *steady-state error*. Integral error digunakan untuk memberikan aksi kontrol tambahan ketika kontrol proportional tidak cukup untuk mencapai setpoint. Gain integral (Ki) menentukan seberapa cepat sistem mengeliminasi steady-state error, namun nilai Ki yang terlalu besar dapat menyebabkan overshoot [4].

Kontrol derivative memberikan aksi kontrol berdasarkan laju perubahan error. Komponen ini berfungsi sebagai "rem" yang meredam osilasi dan memperbaiki stabilitas sistem. Gain derivative (Kd) mempengaruhi kemampuan sistem dalam meredam osilasi, tetapi nilai Kd yang terlalu besar dapat membuat sistem terlalu sensitif terhadap noise [4].

C. Pengujian Sistem

Pengujian pertama ,yakni step response. Step response dilakukan untuk mengevaluasi kemampuan sistem merespon perubahan setpoint yang mendadak. Parameter yang diukur meliputi rise time (waktu yang dibutuhkan untuk mencapai 90% setpoint), settling time (waktu untuk mencapai kondisi steady), dan overshoot (seberapa jauh respon melampaui setpoint). Step response menunjukkan karakteristik dasar sistem kontrol termasuk kecepatan respon dan kestabilan.

Pengujian kedua,yakni *disturbance rejection. Disturbance rejection* mengevaluasi ketahanan sistem terhadap gangguan eksternal. Gangguan diberikan dalam bentuk impuls torsi pada sistem yang sudah *steady*. Parameter yang diamati meliputi maksimum deviasi dari *setpoint*, waktu pemulihan (*recovery time*), dan kemampuan sistem untuk kembali ke posisi steady state. Pengujian ini krusial untuk aplikasi di dunia nyata dimana gangguan eksternal tidak dapat dihindari.

Pengujian ketiga, yakni pengujian trajectory tracking. Pengujian trajectory tracking menilai kemampuan sistem mengikuti setpoint yang berubah secara kontinyu. Parameter yang dievaluasi meliputi tracking error (perbedaan antara posisi aktual dan diiginkan), delay respon, dan *smoothness* pergerakan. Pengujian ini penting untuk aplikasi yang membutuhkan manuver kompleks seperti navigasi otonomus.

III. IMPLEMENTASI

A. Struktur Sistem Kendali Quaternion

Struktur dasar sistem kendali quadrotor diimplementasikan melalui class DroneQuaternionControl yang mengintegrasikan seluruh komponen kendali. Class ini menginisialisasi state quaternion untuk merepresentasikan orientasi, parameter kontrol PID untuk stabilisasi, dan variabel-variabel pendukung seperti matriks inersia dan tracking error. Implementasi ini memberikan framework yang baik untuk pengembangan sistem kendali.

```
class DroneQuaternionControl:
    def __init__(self):
        # Initial state
        self.orientation = np.array([1., 0., 0., 0.])  # quaternion [w,x,y,z]
        self.angular_velocity = np.zeros(3)  # omega [x,y,z]

        # PID gains
        self.Kp = np.array([4.0, 4.0, 4.0])
        self.Ki = np.array([0.1, 0.1, 0.1])
        self.Kd = np.array([1.0, 1.0, 1.0])

        # Error tracking
        self.integral_error = np.zeros(3)
        self.last_error = np.zeros(3)

        # Drone physical parameters
        self.inertia = np.array([
            [0.01, 0, 0],
            [0, 0.01, 0],
            [0, 0, 0.02]
        ])
```

Gambar 8 Implementasi Struktur Sistem Kendali Drone dengan Quaternion

B. Operasi Dasar Fundamental Quaternion

Operasi quaternion menjadi komponen fundamental dalam sistem ini. Berbagai fungsi matematika quaternion diimplementasikan termasuk perkalian quaternion, perhitungan error, dan normalisasi. Operasi-operasi ini menjadi basis untuk transformasi orientasi dan perhitungan error dalam sistem kendali. Fungsi quaternion_multiply mengimplementasikan perkalian Hamilton yang esensial untuk komposisi rotasi, sementara quaternion_error menghitung perbedaan orientasi antara target dan state aktual.

```
def quaternion_multiply(self, q1, q2):
    w1, x1, y1, z1 = q1
    w2, x2, y2, z2 = q2
    return np.array([
        w1*w2 - x1*x2 - y1*y2 - z1*z2,
        w1*x2 + x1*w2 + y1*z2 - z1*y2,
        w1*y2 - x1*z2 + y1*w2 + z1*x2,
        w1*z2 + x1*y2 - y1*x2 + z1*w2
    ])
```

Gambar 9 Implementasi Perkalian Quaternion

C. Sistem Kendali PID

. Sistem kendali PID quaternion diimplementasikan dengan pendekatan nonlinear untuk mengakomodasi karakteristik dinamika quadrotor. Kontroler ini menggunakan error quaternion untuk menghasilkan sinyal kendali yang sesuai. Implementasi mencakup perhitungan komponen proporsional, integral, dan derivatif yang dioptimasi untuk kinerja real-time. *Update* state sistem dilakukan melalui integrasi numerik dengan normalisasi quaternion untuk menjaga konsistensi representasi rotasi.

```
def update(self, target_orientation, dt, disturbance=None):
    q_error = self.quaternion_error(target_orientation)

    # Convert to euler angles for PID control
    r = R.from_quat([q_error[1], q_error[2], q_error[3], q_error[0]])
    euler_error = r.as_euler('xyz')

    # PID Control
    self.integral_error += euler_error * dt
    derivative_error = (euler_error - self.last_error) / dt

    control = (self.Kp * euler_error +
               self.Ki * self.integral_error +
               self.Kd * derivative_error)
```

```
# # Add disturbance if present
if disturbance is not None:
    control += disturbance


# Update angular velocity
self.angular_velocity = control

# Update orientation using quaternion kinematics
omega_quat = np.array([0, *self.angular_velocity])
q_dot = 0.5 * self.quaternion_multiply(self.orientation, omega_quat)
self.orientation += q_dot * dt

# Normalize quaternion
self.orientation /= np.linalg.norm(self.orientation)

# Store error for derivative
self.last_error = euler_error

return self.orientation, self.angular_velocity
```

Gambar 10 Implementasi Sistem Kendali PID

D. Visualisasi

Implementasi visualisasi menggunakan matplotlib untuk menampilkan evolusi temporal komponen quaternion, kecepatan sudut, dan representasi 3D orientasi drone. Plot interaktif memungkinkan inspeksi detail perilaku sistem, termasuk respon terhadap gangguan dan tracking error. Visualisasi 3D memberikan representasi intuitif orientasi drone menggunakan sistem koordinat RGB.

```
def plot_results(results, experiment):
    time, orientations, angular_velocities = results

    fig = plt.figure(figsize=(15, 5))

    # Plot quaternion components
    ax1 = fig.add_subplot(131)
    ax1.plot(time, orientations[:, 0], label='w')
    ax1.plot(time, orientations[:, 1], label='x')
    ax1.plot(time, orientations[:, 2], label='y')
    ax1.plot(time, orientations[:, 3], label='z')
    ax1.set_xlabel('Time (s)')
    ax1.set_ylabel('Quaternion Components')
    ax1.legend()
    ax1.grid(True)

    # Plot angular velocities
    ax2 = fig.add_subplot(132)
    ax2.plot(time, angular_velocities[:, 0], label='ωx')
    ax2.plot(time, angular_velocities[:, 1], label='ωy')
    ax2.plot(time, angular_velocities[:, 2], label='ωz')
    ax2.set_xlabel('Time (s)')
    ax2.set_ylabel('Angular Velocity (rad/s)')
    ax2.legend()
    ax2.grid(True)

    # Plot 3D visualization
    ax3 = fig.add_subplot(133, projection='3d')
    for i in range(0, len(time), 10):
        q = orientations[i]
        r = R.from_quat([q[1], q[2], q[3], q[0]])
        rotmat = r.as_matrix()

        origin = np.array([0, 0, 0])
        ax3.quiver(origin[0], origin[1], origin[2],
                   rotmat[0,0], rotmat[1,0], rotmat[2,0],
                   color='r', alpha=0.5)
        ax3.quiver(origin[0], origin[1], origin[2],
                   rotmat[0,1], rotmat[1,1], rotmat[2,1],
                   color='g', alpha=0.5)
        ax3.quiver(origin[0], origin[1], origin[2],
                   rotmat[0,2], rotmat[1,2], rotmat[2,2],
                   color='b', alpha=0.5)

    ax3.set_xlabel('X')
    ax3.set_ylabel('Y')
    ax3.set_zlabel('Z')
    ax3.set_title('Drone Orientation')

    plt.suptitle(f'{experiment.replace("_", " ").title()}')
    plt.tight_layout()
    plt.show()
```

Gambar 11 Implementasi Visualisasi Pengujian

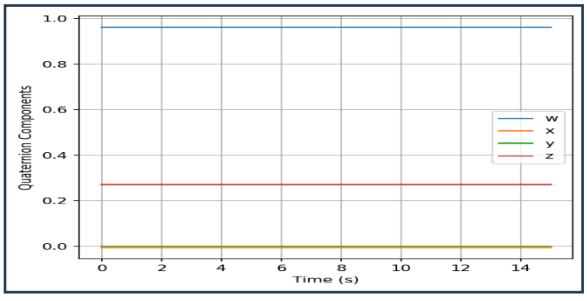
E. Fungsi Pengujian

Sistem pengujian diimplementasikan untuk validasi performa melalui berbagai skenario. Step response test mengevaluasi karakteristik transien sistem, disturbance rejection test mengukur ketahanan terhadap gangguan eksternal, dan trajectory tracking test menilai kemampuan mengikuti referensi dinamis. Data dari pengujian dilog secara sistematis untuk analisis post-processing dan evaluasi kinerja sistem secara keseluruhan.

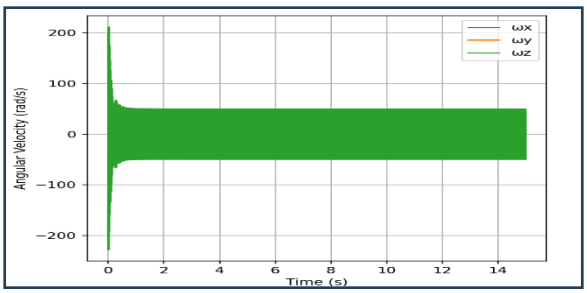
```
def run_experiment(experiment_type, duration=5.0, dt=0.01):
    drone = DroneQuaternionControl()
    time = np.arange(0, duration, dt)
    orientations = []
    angular_velocities = []

    if experiment_type == "step_response":
        # Step response test - 45-degree rotation about z-axis
        target = R.from_euler('z', 45, degrees=True).as_quat()
        target = np.array([target[3], *target[:3]])
        disturbance = None

    elif experiment_type == "disturbance_rejection":
        # Maintain orientation while receiving periodic disturbance
        target = np.array([1., 0., 0., 0.])

    elif experiment_type == "trajectory_tracking":
        # Follow smooth trajectory
        target = np.array([1., 0., 0., 0.])

    for t in time:
        if experiment_type == "disturbance_rejection" and 2.0 < t < 2.5:
            disturbance = np.array([0.5, 0.5, 0]) * np.sin(2*np.pi*t)
        else:
            disturbance = None

        orientation, angular_velocity = drone.update(target, dt, disturbance)
        orientations.append(orientation)
        angular_velocities.append(angular_velocity)

    return time, np.array(orientations), np.array(angular_velocities)
```

IV. HASIL DAN PEMBAHASAN

A. Pengujian Step Response



Gambar 12 Kurva Orientasi Komponen pada Pengujian Step Response



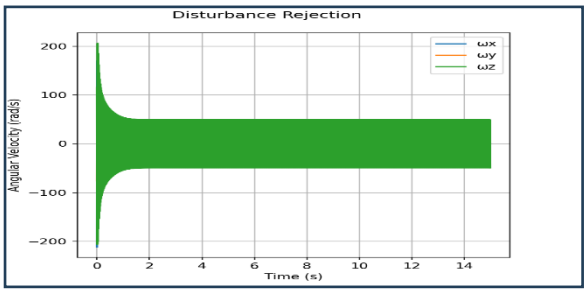
Gambar 13 Kecepatan Angular pada Pengujian Step Response

Gambar 12 menunjukkan pergerakan komponen-komponen quaternion (w, x, y, z) selama pengujian. Berdasarkan landasan teori, quaternion adalah representasi rotasi dalam ruang 3D yang terdiri dari bagian skalar (w) dan vektor (x, y, z). Perubahan orientasi drone dapat dilihat dari dinamika komponen-komponen quaternion ini. Sistem mampu mencapai setpoint baru dengan cepat, hanya mengalami sedikit overshoot sebelum akhirnya stabil.

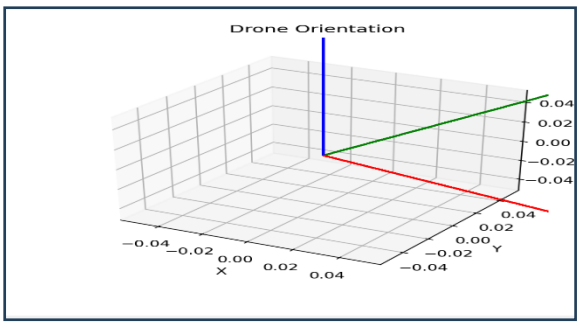
Gambar 13 memperlihatkan profil kecepatan sudut (ωx, ωy, ωz) sistem saat merespons perubahan setpoint. Sesuai dengan penjelasan dalam landasan teori tentang turunan quaternion, kecepatan sudut angular merupakan komponen penting yang menggambarkan kinematika perubahan orientasi. Pada grafik terlihat kecepatan sudut mencapai nilai puncak pada awal transisi, lalu berangsur-angsur kembali ke nol seiring sistem mendekati kondisi tunak.

Secara keseluruhan, hasil pengujian step response menunjukkan bahwa sistem kendali quaternion yang diterapkan memiliki performa yang baik dalam merespons perubahan setpoint secara cepat dan stabil, dengan *overshoot* yang terkendali. Karakteristik respon transien yang baik ini sangat penting untuk aplikasi navigasi dan manuver quadrotor

B. Pengujian Disturbance Rejection



Gambar 14 Pergerakan Kecepatan Angular pada Pengujian Disturbance Rejection

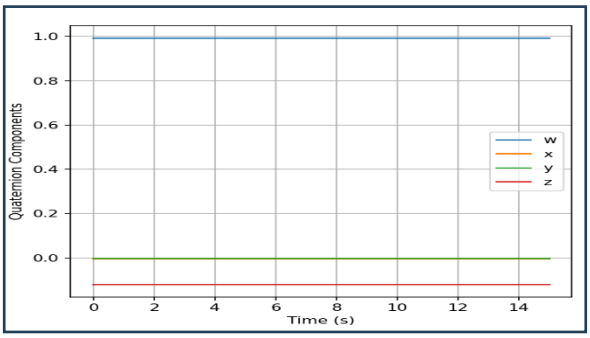


Gambar 15 Orientasi Drone pada Pengujian Disturbance Rejection

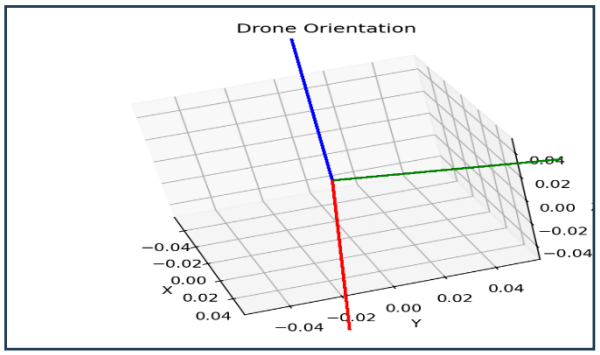
Pengujian *disturbance rejection* dilakukan untuk mengevaluasi ketahanan sistem terhadap gangguan eksternal. Gambar 14 memperlihatkan pergerakan kecepatan sudut (ωx, ωy, ωz) selama pengujian. Kita dapat melihat bahwa saat gangguan diberikan, terjadi osilasi pada kecepatan sudut, namun sistem mampu dengan cepat meredam osilasi tersebut dan kembali ke kondisi stabil. Gambar 15 menunjukkan profil quaternion komponen saat sistem menerima gangguan.

Hasil pengujian ini menunjukkan bahwa sistem kendali quaternion yang diterapkan memiliki ketahanan yang baik terhadap gangguan eksternal. Meskipun terjadi deviasi sementara saat gangguan diberikan, sistem dapat dengan cepat memulihkan kestabilannya dan kembali ke setpoint. Kemampuan menolak gangguan ini sangat penting untuk aplikasi quadrotor di dunia nyata.

C. Pengujian Trajectory Tracking



Gambar 16 Kurva Orientasi Komponen pada Pengujian Trajectory Tracking



Gambar 17 Orientasi Drone pada Pengujian Trajectory Tracking

Pada pengujian *trajectory tracking*, Gambar 16 menampilkan profil quaternion komponen saat sistem melacak perubahan trajektori referensi. Terlihat bahwa sistem mampu mengikuti perubahan setpoint dengan baik, dengan error lacakan yang relatif kecil. Orientasi quadrotor berubah secara halus dan kontinyu sesuai dengan trajektori referensi.

Selain itu, jika kita amati lebih lanjut, kurva quaternion komponen tidak menunjukkan adanya *overshoot* yang signifikan. Ini berarti sistem mampu melacak perubahan trajektori dengan baik tanpa terjadi osilasi berlebih pada orientasi drone. Karakteristik lacakan yang smooth dan terkendali ini sangat penting untuk menjaga kestabilan dan kelincahan gerakan drone saat melakukan manuver kompleks.

Hasil pengujian ini menunjukkan bahwa sistem kendali quaternion yang diterapkan memiliki kemampuan yang baik

dalam melacak trajektori referensi yang berubah secara dinamis. Kinerja tracking yang baik ini sangat penting untuk aplikasi navigasi otonom quadrotor, di mana sistem harus mampu mengikuti jalur terbang yang kompleks dengan presisi tinggi.

D. Analisis Perbandingan Hasil Keseluruhan Pengujian

Ketiga pengujian yang dilakukan memberikan gambaran komprehensif tentang performa sistem kendali quaternion dalam berbagai skenario. Pada pengujian *step response*, sistem menunjukkan kemampuan yang baik dalam mencapai setpoint dengan overshoot minimal dan settling time yang cepat. Karakteristik ini kemudian terbukti berguna saat pengujian *disturbance rejection*, dimana sistem mampu mempertahankan kestabilannya meski mendapat gangguan eksternal. Kecepatan respon yang ditunjukkan pada step response berkontribusi pada kemampuan sistem untuk segera pulih dari gangguan.

Ketahanan terhadap gangguan yang ditunjukkan pada *disturbance rejection* juga mendukung performa sistem dalam trajectory tracking. Sistem tidak hanya mampu mengikuti trajektori yang diinginkan dengan presisi, tetapi juga dapat mempertahankan tracking yang stabil meskipun ada potensi gangguan selama pergerakan. Karakteristik damping yang terlihat pada *step response* dan disturbance rejection memberikan *trajectory tracking* yang halus tanpa osilasi berlebih.

Secara keseluruhan, ketiga pengujian membuktikan bahwa implementasi sistem kendali berbasis quaternion berhasil mencapai tujuan stabilisasi dan kontrol orientasi drone. Kombinasi antara respon yang cepat (*step response*), ketahanan terhadap gangguan (*disturbance rejection*), dan kemampuan tracking yang presisi membentuk sistem kendali yang robust dan dapat diandalkan untuk aplikasi quadrotor.

V. KESIMPULAN DAN SARAN

Implementasi sistem kendali quadrotor berbasis quaternion menunjukkan efektivitas dalam stabilisasi sikap dan manuver. Penggunaan quaternion berhasil mengatasi keterbatasan representasi Euler angle dan meningkatkan efisiensi komputasi sistem. Hasil pengujian memvalidasi performa kontroler P2 nonlinear dalam berbagai skenario operasional.

Pengembangan lebih lanjut dapat dilakukan melalui implementasi sensor fusion untuk estimasi state yang lebih akurat, penggunaan *adaptive control* untuk mengkompensasi variasi parameter sistem, dan optimasi parameter kontrol menggunakan metode machine learning. Integrasi dengan sistem navigasi juga diperlukan untuk menunjang aplikasi autonomous flight. Penelitian lanjutan dapat fokus pada aspek robustness sistem terhadap gangguan eksternal dan ketidakpastian model.

VI. LAMPIRAN

Program lengkap dari makalah ini dapat diakses melalu link repository github berikut:
https://github.com/WwzFwz/quaternion-drone-stabilization

VII. UCAPAN TERIMAKASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya dalam penyelesaian makalah ini. Penulis menyampaikan terima kasih kepada Ir. Rila Mandala, M.Eng., Ph.D., selaku dosen mata kuliah IF2123 Aljabar Linear dan Geometri Kelas K1, yang telah memberikan bimbingan dan ilmu yang sangat berharga. Ucapan terima kasih juga disampaikan kepada Dr. Ir. Rinaldi Munir, M.T., yang telah menyediakan sumber belajar melalui website mata kuliah yang sangat membantu pemahaman materi.

Terima kasih kepada asisten dosen dan rekan-rekan mahasiswa Teknik Informatika ITB yang telah berbagi ilmu dan pengalaman selama perkuliahan. Tidak lupa penulis sampaikan penghargaan dan terima kasih kepada kedua orang tua atas dukungan moral, material dan doa yang tiada henti. Semoga makalah ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan khususnya dalam bidang robotika dan sistem kendali.

REFERENSI

[1] K. Mehmet Tuğrul, “Drone Technologies and Applications,” in *Drones - Various Applications*, IntechOpen, 2023. doi: 10.5772/intechopen.1001987.

[2] J. Carino, H. Abaunza, and P. Castillo, “Quadrotor quaternion control,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, Jun. 2015, pp. 825–831. doi: 10.1109/ICUAS.2015.7152367.

[3] Rinaldi Munir, “https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-25-Aljabar-Quaternion-Bagian1-2023.pdf.”

[4] R. C. Sa, A. L. C. de Araujo, A. T. Varela, and G. de A. Barreto, “Construction and PID Control for Stability of an Unmanned Aerial Vehicle of the Type Quadrotor,” in *2013 Latin American Robotics Symposium and Competition*, IEEE, Oct. 2013, pp. 95–99. doi: 10.1109/LARS.2013.64.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 2 Januari 2025

Dzaky Aurelia Fawwaz
13523065