

PERENCANAAN SISTEM KENDALI MODEL REFERENCE

AI Al ^{1)*}, Gusri Rahmawati ²⁾, Aswir Premadi ³⁾, Asnal Effendi ⁴⁾

^{1), 2), 3), 4)} Program Studi Teknologi Rekayasa Instalasi Listrik

^{1), 2), 3), 4)} Fakultas Vokasi

^{1), 2), 3), 4)} Institut Teknologi Padang

*Corresponding Author E-mail: al.mtdrs@gmail.com

Abstract

The commercial use of drones has increased rapidly in this sector in the last few decades, one of which is the Hexacopter Type Unmanned Aerial Vehicle (UAV). Hexacopter is a flying robot without a pilot that can be controlled manually using a remote control and controlled automatically or on autopilot. The advantage of the hexacopter is its ability to maneuver flexibly, it can move in 8 cardinal directions without needing to turn first. However, despite their advantages, certain hexacopters must continue to fly stably under a variety of changing conditions, such as changing weather, different loads, or changing environmental conditions. For this reason, a control system is needed that is able to adapt to these changes. This condition makes it interesting to carry out experimental and simulation studies, namely designing a hexacopter with Model Referene Adaptive Control (MRAC) control. By using a Model Reference Adaptive Control (MRAC) control system with the main control using PID control, the response given is good and optimal, the system parameters reach the values: rise time (ts) 1.0709 s, delay time (td) 0.99 s, peak time (tp) 2.6600 s, settling time (ts) 3.9351 s, overshoot (mp) 4.0847 and steady state error 0.0001%. The mathematical analysis of the MRAC control system needs to be studied in more depth so that the MRAC control calculations become more accurate.

Keywords: Hexacopter, MRAC, PID

Abstrak

Penggunaan drone secara komersial meningkat pesat di sektor ini dalam beberapa dekade terakhir, salah satunya adalah Unmanned Aerial Vehicle (UAV) Tipe Hexacopter. Hexacopter merupakan robot terbang tanpa pilot yang dapat dikendalikan secara manual dengan menggunakan remote control dan dikendalikan secara otomatis atau autopilot. Kelebihan dari hexacopter adalah kemampuannya dalam bermanuver dengan fleksibel, dapat bergerak ke 8 arah mata angin tanpa perlu berbelok terlebih dahulu. Namun, terlepas dari kelebihannya, hexacopter tertentu harus terus terbang dengan stabil dalam berbagai kondisi yang berubah, seperti perubahan cuaca, beban yang berbeda, atau perubahan kondisi lingkungan. Untuk itu diperlukan suatu sistem pengendalian yang mampu beradaptasi terhadap perubahan tersebut. Kondisi ini menjadi menarik untuk melakukan studi eksperimen dan simulasi yaitu merancang hexacopter dengan kendali Model Referene Adaptive Control (MRAC). Dengan menggunakan sistem kendali Model Reference Adaptive Control (MRAC) dengan kendali utama menggunakan kendali PID, respon yang diberikan baik dan optimal, parameter sistem mencapai nilai: rise time (ts) 1.0709 s, delay time (td) 0.99 s, peak time (tp) 2,6600 s, settling time (ts) 3,9351 s, overshoot (mp) 4,0847 dan steady state error 0,0001%. Analisis matematis sistem kendali MRAC perlu dipelajari lebih mendalam agar perhitungan kendali MRAC menjadi lebih akurat.

Kata Kunci: Hexacopter, MRAC, PID

1. PENDAHULUAN

Penggunaan drone secara komersial semakin pesat di sektor ini dalam beberapa dekade terakhir [1]. Salah satunya adalah Unmanned Aerial Vehicle (UAV) Jenis Hexacopter. Hexacopter memiliki karakteristik yaitu dengan memiliki 6 buah motor baling-baling sebagai penggeraknya dimana 3 buah motor berputar searah jarum jam atau CW (Clock Wise) dan 3 motor lainnya bergerak berlawanan dengan arah jarum jam atau CCW (Conter Clock Wise) [2]. Pesawat ini memiliki konfigurasi dengan enam motor yang mengendalikan pergerakan dan stabilitas. Kompleksitas struktur ini mengharuskan pengembangan sistem kendali yang canggih untuk memastikan stabilitas, penyeimbangan, dan navigasi yang tepat [3]. Kelebihan hexacopter adalah kemampuannya yang dapat bermanuver dengan fleksibel, dapat bergerak ke 8 arah mata angin tanpa perlu memutar terlebih dahulu, dapat terbang

secara vertical sehingga tidak memerlukan landasan pacu dan kemampuan yang mendasar hexacopter adalah dapat berhenti (hover) di suatu titik lokasi di udara. Dengan kelebihan drone tersebut dimanfaatkan manusia untuk berbagai keperluan seperti pemetaan, penyemprotan area perkebunan, pengiriman barang dan lain-lain [4].

MRAC dapat digunakan untuk mengoptimalkan respons kendali Hexacopter terhadap perubahan lingkungan atau kondisi operasional yang berbeda. MRAC merupakan salah satu skema kendali adaptif sebagaimana performansi keluaran system, dapat mengikuti performansi keluaran model referensinya. Dengan mengintegrasikan kendali MRAC dalam perancangan hexacopter, diharapkan dapat meningkatkan kemampuan yang adaptif dan efisiensi operasional dari hexacopter dalam berbagai situasi operasional dengan fungsi [5]. Dalam meningkatkan kemampuan adaptif dan efisiensi operasional Hexacopter dalam berbagai situasi operasional perlu memanfaatkan model referensi dan simulasi untuk menghasilkan sinyal kontrol yang disesuaikan secara adaptif untuk mengoptimalkan respons kendali hexacopter terhadap perubahan lingkungan atau kondisi operasional [6]. Untuk memecahkan masalah yang terjadi tersebut perlu pemahaman mendalam tentang konsep dan prinsip dasar MRAC dan bagaimana konsep ini didesain. Selain itu perancangan kontrol adaptif perlu dilakukan untuk membuat tujuan kendali yang di inginkan yang sesuai dengan perubahan yang terjadi [7].

Mencermati penelitian terdahulu tentang “Perancangan sistem Kendali MRAC untuk hexacopter”. Salah satunya penelitian tentang “Peningkatan performance kestabilan hexacopter dengan sistem MRAC”. Yang menggabungkan pengontrol PID tradisional dengan Model Reference Adaptive Control (MRAC), dan “Implementasi Dan Perancangan Sistem Kendali Posisi Ketinggian (Altitude) Quadcopter Berbasis Arduino Uno R3” yang mendapatkan hasil Quadcopter bisa menahan posisi ketinggian sesuai setpoint yang diberikan. Sedangkan untuk referensi pengambilan parameter mengambil data dari penelitian “Model Referensi Kontrol Adaptif Ketinggian Udara Untuk Kendaraan Tak Berawak Dengan Kontrol PID” dan untuk simulasi di matlab mengambil referensi dari penelitian “Model Reference Adaptive Control and Neural Network Based Control of Altitude of Unmanned Aerial Vehicles”. Hidayatullah 2022, dalam penelitian yang berjudul “Peningkatan performance kestabilan hexacopter dengan sistem MRAC” Penerapan sistem PID adaptif untuk meningkatkan kinerja stabilitas pesawat hexacopter dengan menggabungkan pengontrol PID tradisional dengan Model Reference Adaptive Control (MRAC), sistem ini mampu beradaptasi dengan perubahan lingkungan dan karakteristik komponen, dengan konstanta $K_p=1,455$, $K_i= 1,3045$ dan $K_d= 0,34$ [8]. Perubahan lingkungan dan karakteristik komponen dengan nilai performansi sistem rise time = 0,006 mS, Settling time = 0,0012,5 mS, overshoot = 5%, error steady state = 0%, Delay time= 0,0038 mS, dan Peak time= 0,0068 mS, sehingga meningkatkan stabilitas selama manuver penerbangan. Al et al. 2023, tujuan penelitiannya adalah untuk meningkatkan kinerja manuver roll menggunakan kontrol PID adaptif dan pengontrol logika fuzzy [9]. Setelah pengujian real-plant, kinerja sistem diperoleh sebagai berikut: settling time = 24mS; waktu puncak 21mS; waktu naik =9mS; waktu tunda = 4,4mS; persen overshoot dan kesalahan kondisi tunak =1%. Sedangkan [10], melakukan penelitian tentang “Implementasi Dan Perancangan Sistem Kendali Posisi Ketinggian (Altitude) Quadcopter Berbasis Arduino Uno R3” kemudian mendapatkan hasil Quadcopter bisa menahan posisi ketinggian sesuai setpoint yang diberikan mampu stabil di ketinggian tersebut dengan menetapkan nilai $P = 3.0$, $I = 0.02$, $D = 0$, untuk menahan posisi di ketinggian 10 cm rata-rata error-nya 0.42 cm, pada ketinggian 20 cm rata-rata error-nya 0.29 cm dan pada ketinggian 30 cm rata-rata error-nya 0.32 cm [11].

Adiyansyah et al. 2023, dalam penelitiannya yang berjudul “Model Referensi Kontrol Adaptif Ketinggian Udara Untuk Kendaraan Tak Berawak Dengan Kontrol PID” menjelaskan Model Referensi Adaptive Control (MRAC) merupakan kontrol adaptif tingkat tinggi yang menggunakan parameter tetap dan memiliki turunan Proportional-Integral-Derivative untuk mengendalikan sikap Pitch pesawat [3]. MRAC adalah kontrol adaptif di mana kinerja sistem akan mengikuti kinerja model referensinya. Dan mendapatkan hasil pengujian yang menunjukkan bahwa ketika terjadinya perubahan parameter pada sistem, controller mampu memperbaiki respons agar tetap mengikuti model referensi dan metode adaptasi MRAC memiliki kemampuan yang baik dengan

waktu yang relatif singkat. Dimana pesawat mendapatkan kestabilan pada ketinggian 2.0 membutuhkan waktu selama 600s atau 10menit. [12], melakukan penelitian dengan judul “Desain Dan Simulasi Sistem Kontrol PID Pada Pengendalian Kecepatan Motor Penggerak Quadcopter” dengan melakukan simulasi pada Simulink MATLAB [12]. Hasil simulasi menunjukkan bahwa dengan kontrol PID pada ketinggian 10 m sampai 90 m didapatkan respon waktu naik lebih cepat 0.01 detik, overshoot 0%, dan waktu steady state 0.06 detik. Sedangkan pengaturan PI pada ketinggian 10 m sampai 90 m menghasilkan respon waktu naik lebih lama 0.013 detik, overshoot 0% dan waktu steady state 0.1 detik. Matthews and Yi 2019, Melakukan penelitian dengan judul “Model Reference Adaptive Control and Neural Network Based Control of Altitude of Unmanned Aerial Vehicles” dengan pendekatan kontrol adaptif tingkat tinggi digunakan dengan parameter tetap berupa proporsional-integral-derivative menggunakan sistem identifikasi matlab mendapatkan hasil penelitian menunjukkan bahwa Model Reference Adaptive Controller adalah kontrol adaptif tingkat tinggi yang mampu memberikan kinerja pelacakan yang tinggi dalam menghadapi ketidakpastian. Penggunaan Model Reference Adaptive Controller dalam menghadapi gangguan pada sistem telah diuji dan dievaluasi untuk mengamati kinerja respon [13-14].

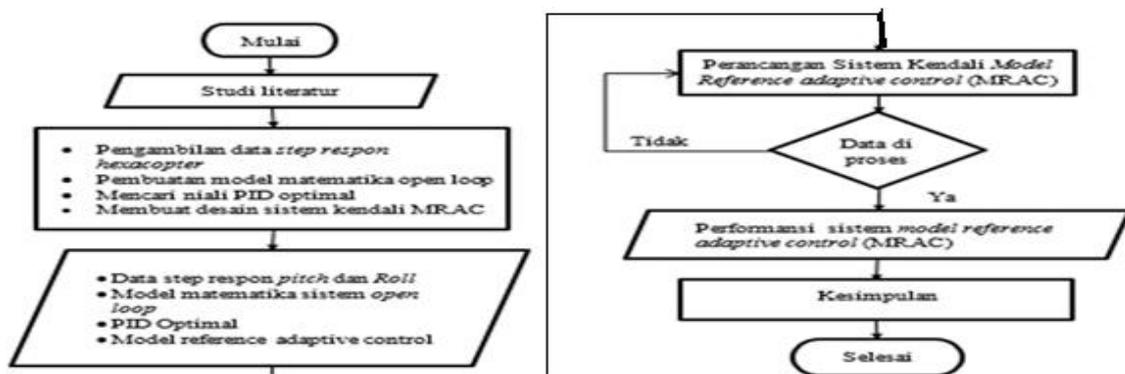
2. METODOLOGI

Penelitian ini di lakukan di Laboratorium Teknik Elekrto Institut Teknologi Padang, selama 4 bulan yakni pada bulan September 2023 sampai Bulan Maret 2024. Adapun stan Uji alat ini diperlihatkan seperti Gambar 1.



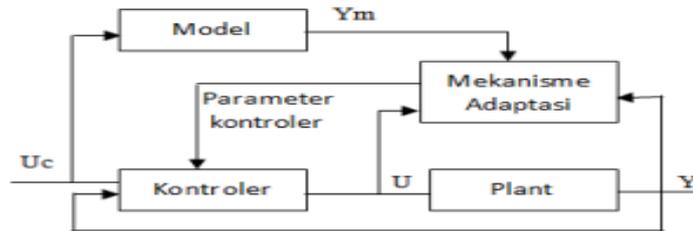
Gambar 1. Stand Uji Penelitian

Penelitian ini dimulai dengan studi literatur, setelah memahami data apa saja yang dibutuhkan dilakukanlah pengambilan data *step respon* pada *hexacopter*, kemudian melakukan pembuatan model matematika, mencari nilai PID optimal serta membuat desain simulasi dari sistem kendali MRAC, setelah semua data di dapatkan maka perancangan sistem kendali sudah bisa di kerjakan, jika hasil sistem kendali belum sesuai maka dilakukan perancangan ulang hingga mendapatkan hasil yang baik, Jika sudah sesuai kebutuhan dilanjutkan dengan mengambil data performansi sistem kendali dan kesimpulan. Diagram alir penelitian dapat dilihat pada Gambar 2.



Gambar 2. Diagram Alir Penelitian

Model Reference Adaptive Control (MRAC) adalah salah satu skema kontrol adaptif di mana kinerja keluaran sistem (proses) mengikuti kinerja keluaran model referensi. Parameter pengontrol disesuaikan melalui mekanisme penyetelan berbasis *error* di mana adalah perbedaan antara keluaran plan dan keluaran model referensi seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Diagram blok skema MRAC

Dari Gambar 3 dapat dilihat bahwa skema sistem MRAC terdapat dua *loop*, *Loop* pertama adalah *loop* umpan balik normal antara *output* proses dengan kontroler sedangkan *loop* kedua adalah *loop* yang digunakan untuk melakukan mekanisme pengaturan parameter kontroler. Pada *loop* kedua ini dilakukan proses untuk *update* parameter – parameter kontroler maupun parameter *plant* sesuai dengan skema adaptif yg digunakan. Pengaturan dilakukan dengan meminimalkan sinyal *error*, sehingga keluaran sistem (*y*) sesuai dengan keluaran model referensinya (*ym*).

$$y_m(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1)$$

Keterangan :

$y_m(s)$ = fungsi transfer dari sistem yang merespons input $s(t)$

S = variabel kompleks dalam domain Laplace

ω_n = frekuensi natural sistem (*natural frequency*)

ζ = adalah rasio redaman (*damping ratio*).

Persamaan di atas merupakan fungsi transfer dari sistem orde 2 dalam domain *laplace* yang digunakan dalam teori kontrol dan sistem. Yang merupakan bentuk umum dari respons dinamis dari sistem orde dua terhadap input $s(t)$ yang dinyatakan sebagai fungsi transfer $Y_m(s)$ dalam domain Laplace. *Plan* diasumsikan memiliki struktur yang diketahui, meskipun parameternya tidak diketahui. Untuk *plant* linier, ini berarti bahwa jumlah kutub dan jumlah nol diasumsikan diketahui, tetapi lokasi kutub dan nol ini tidak diketahui. Untuk pembangkit *nonlinier*, ini berarti bahwa struktur persamaan dinamik diketahui, tetapi beberapa parameter tidak diketahui. Model referensi digunakan untuk menentukan respons ideal sistem kontrol adaptif terhadap perintah eksternal. Secara intuitif, model ini memberikan respons *plant* yang ideal yang harus dicari oleh mekanisme adaptasi dalam menyesuaikan parameter. Pemilihan model referensi adalah bagian dari desain sistem kontrol adaptif. Pilihan ini harus memenuhi dua persyaratan. Di satu sisi, itu harus mencerminkan spesifikasi *performance* dalam tugas kontrol, seperti waktu naik, waktu penyelesaian, *overshoot* atau karakteristik domain frekuensi. Disisilain, perilaku ideal ini harus dapat dicapai untuk sistem kontrol adaptif, yaitu, ada beberapa kendala yang melekat pada struktur model referensi (misalnya, urutan dan derajat relatifnya) mengingat struktur model pabrik yang diasumsikan.

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, simulasi dibuat menjadi beberapa bagian simulasi. Diantaranya mencari nilai PID yang optimal, mencari nilai transfer function sebagai input dalam blok diagram MRAC. Kecepatan motor pada sebuah hexacopter dapat bervariasi secara signifikan tergantung pada desain dan komponen yang digunakan. Secara umum motor dapat dinyatakan dalam Revolutions Per Minute (Rpm). Kecepatan motor diukur menggunakan alat ukur tachometer. Data kecepatan masing masing motor hexacopter dapat dilihat pada tabel 1.

Tabel 1. Kecepatan Propeler Motor

No	Motor	Kecepatan
1	Motor 1	1716 Rpm
2	Motor 2	1096 Rpm
3	Motor 3	1871 Rpm
4	Motor 4	1551 Rpm
5	Motor 5	2196 Rpm
6	Motor 6	2634 Rpm

Sistem *Open loop* merupakan sistem yang tidak memiliki umpan balik atau *feedback* untuk mengontrol ataupun mengatur kinerjanya. Pada sistem *open loop* nilai PID yang digunakan dapat dilihat pada Tabel 2. Nilai tersebut merupakan nilai *default* yang sudah ada di dalam *flight controller drone hexacopter* yang digunakan dalam penelitian ini.

Tabel 2. Nilai PID Default

No	Gerakan	Nilai P	Nilai I	Nilai D
1	<i>Picth</i>	1,615	0,99344	0,00300
2	<i>Roll</i>	1,615	0,99344	0,00300
3	<i>Yaw</i>	0,200	0,020	0,0000

Pada sistem *open loop* ini dilakukan dengan melihat langsung gerakan *drone hexacopter* yang telah terhubung ke *software mission planner* menggunakan kabel USB, dengan menggunakan percobaan respon *plant* dari input *step* pada rangkaian *open-loop* sehingga dapat menghasilkan grafik seperti Gambar 4 dan Gambar 5.

**Gambar 4.** Grafik Respon Gerakan Pitch

Dari Gambar 4 dapat dilihat grafik berwarna kuning merupakan grafik dari *set point* atau sudut yang ingin dicapai dari gerakan *pitch*, grafik ini diperoleh secara *open loop* dari gerakan *throttle remote control drone* dari sudut 0° hingga sudut maximal *throttle*, sehingga sudut yang terbaca pada sumbu y di *software Mission Planner* adalah angka 35° . Sedangkan grafik berwarna kuning merupakan grafik pergerakan *drone* atau respon sistem *open loop* Ketika perintah diberikan yaitu menggerakkan *throttle* pada sudut *maximum*, sehingga respon sistem dapat dilihat pada sumbu x (waktu), di angka 686,2 *drone* baru merespon dan respon yang diberikan tidaklah tetap, pada angka 686,8 respon sudah bisa mencapai target namun mengalami *overshoot* sampai 15%, pada angka 687,4 respon sudah mulai menyesuaikan dengan target atau *set point* namun masih ada *osilasi* dan baru bisa menyesuaikan Kembali dengan target dan stabil di angka 689,2. Atau waktu yang dibutuhkan *drone* bisa stabil 5,84s. Dari gerakan *pitch* ini mempengaruhi gerakan *roll hexacopter* hingga dapat dilihat pada grafik warna biru dan hijau mengalami *osilasi* dan tidak stabil. Seharusnya dalam melakukan gerakan *pitch* gerakan lainnya yaitu *roll* dalam kondisi stabil atau tetap disudut 0° . Performa sistem gerakan *Pitch* secara *open loop* dapat dilihat pada tabel 3.

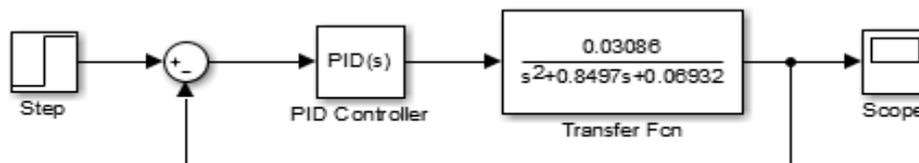
Tabel 3. Performance system open loop Pitch

No	Step respon	Nilai
1	Rise Time (T_r)	1,99 s
2	Delay time (T_d)	2,89 s
3	Peak time (T_p)	4,34 s
4	Setling Time (T_s)	9,56 s
5	Persen Overshoot (%Mp)	15,5%,

Dari tabel 3. waktu naik yang memotong sumbu yang pertama dari hasil grafik sudah cukup bagus, namun waktu yang di butuhkan untuk sampai ke titik optimal cukup lama yaitu 9,56 s, dan mengalami *overshoot* yang tinggi yaitu 15,5% untuk itu perlu memperbaiki respon sistem dengan memperbaiki sistem kendali. Untuk memperoleh model matematis perlu dilakukan sebagaimana nantinya akan digunakan untuk nilai *plant* pada sistem kendali yang akan dirancang, yaitu sistem kendali MRAC. Langkah pertama yang dilakukan untuk membuat model matematis adalah mengeksport grafik *open loop* yang telah dilakukan menggunakan situs online grapher atau plot digitizer sehingga menjadi data. Setelah data tersebut didapatkan kemudian di salin ke *Microsoft excel*, untuk memudahkan proses export ke *software* matlab. Untuk menjadikan data ke model matematika dengan memanfaatkan tools yang ada di matlab yaitu *System Identification Toolbox* (SIT) matlab. Untuk mendapatkan model matematis gerakan *pitch* perlu memasukan data yang sudah diperoleh melalui plot digitizer ke matlab. Untuk mendapatkan model matematis pada menu import data pilih *Time-Domain signals* dan imputkan beberapa data yang diperlukan, *input* dan *output* yang di masukan sesuai dengan nama data yang telah di upload ke workspace, data X dan Y merupakan data gerakan *Pitch*. Dan beberapa data lainnya, perlu diperhatikan angka *starting time*, haruslah dimulai dari angka 0, Jika data sudah di *import* maka kolom *import* model akan terisi. Selanjutnya untuk memproses data hingga menjadi *transfer function*, pada kolom *estimate* pilih *transfer function*, isi data yang diperlukan. Perlu di perhatikan untuk membuat *transfer function* pastikan pada kolom *operation working data* adalah data yang akan dibuatkan model matematikanya. Setelah proses *estimate* dilakukan akan muncul model matematika nya pada kolom *import models*. Model matematika gerakan *pitch* dapat dilihat pada persamaan 2.

$$\frac{0.03086}{s^2+0.8497s+0.6932} \quad (2)$$

PID optimal merujuk pada pencarian nilai parameter PID yang memberikan kinerja kontrol yang terbaik sesuai dengan kriteria respon sistem. Parameter PID bisa dianggap optimal ketika *output* dapat memberikan respon persyaratan sistem dengan cara yang terbaik seperti seberapa cepat sistem dapat mencapai *setpoint* atau stabil, seberapa jauh *overshoot* respon sistem melebihi *setpoint* sebelum kembali ke *setpoint*. Seberapa baik sistem dapat mengatasi sistem mengatasi gangguan atau kesalahan yang tak terduga tanpa mengalami *osilasi* yang tidak diinginkan. Ada beberapa cara yang dilakukan untuk mendapatkan nilai parameter PID optimal, namun cara yang digunakan untuk mendapatkan PID optimal menggunakan fitur aplikasi Matlab, yaitu PID Tuner, seperti diagram blok Gambar 5.

**Gambar 5. Diagram Blok Pengujian PID Optimal**

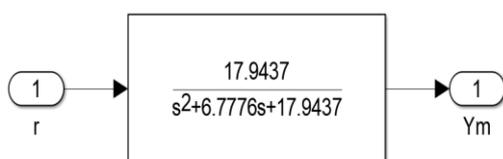
Input step digunakan untuk melihat sinyal keluaran sistem berupa step respon, keluaran step menjadi inputan PID controller dimana nilai PID yang digunakan terlebih dahulu adalah nilai PID Optimal terdahulu. Selanjutnya transfer function yang digunakan merupakan model matematika dari gerakan *open loop pitch*, keluaran transfer function menjadi *feedback* bagi PID Controller, dimana sistem kendali PID pada prinsipnya menghasilkan sinyal kontrol yang mengatur *output* sistem berdasarkan

kesalahan. dan scope digunakan untuk melihat bentuk sinyal keluaran dari sistem, setelah semua block diagram dibuat, pada block PID controller bisa melakukan *self-tuning* mencari nilai PID yang lebih optimal dari PID sebelumnya dengan memanfaatkan PID Tuner. Sehingga di dapatkan nilai $P=14,4525$, $I=8,1981$, $D=6,3696$.

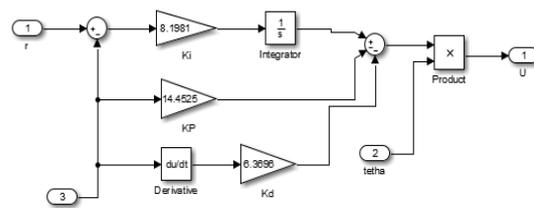
Tabel 4. Performansi sistem PID optimal

No	Step respon	Nilai
1	Rise Time (T_r)	3,9 s
2	Delay time (T_d)	2,8 s
3	Peak time (T_p)	6,50 s
4	Settling Time (T_s)	16 s
5	Persen Overshoot ($\%Mp$)	7%

Berdasarkan Tabel 4 waktu yang dibutuhkan respon untuk mencapai titik stabil cukup lama maka diperlukan solusi yaitu dengan menambahkan sistem kendali Model reference adaptive control (MRAC). MRAC adalah sistem kontrol yang dirancang untuk mengontrol sebuah sistem dinamis dengan mengadaptasi parameter kontrolnya secara otomatis berdasarkan perubahan dalam dinamika sistem yang sebenarnya. Tujuannya adalah untuk mengontrol sistem tersebut sehingga mengikuti perilaku model referensi yang telah ditentukan. Dengan menambahkan MRAC ke dalam sistem kendali PID untuk meng-ejust konstanta PID diharapkan mengantisipasi error dan delta error sesaat. Selain pengaruh eksternal juga dapat menyesuaikan dengan keadaan perubahan *set point*. Perubahan *set point* ini diidentik dengan merubah-ubah perintah melalui *remote control* ataupun *throttle* dan perubahan parameter eksternal lainnya. Secara umum, sistem *kontrol adaptif* model-referensi secara skematis dapat diwakili oleh Gambar 6, Sistem ini terdiri dari empat bagian: plant yang berisi parameter gerakan *open loop pitch*, model *reference* untuk menentukan secara ringkas output yang diinginkan dari sistem kontrol, *hukum kontrol* umpan balik yang berisi parameter yang dapat disesuaikan, dan *mekanisme adaptasi* untuk memperbarui parameter yang dapat disesuaikan. Sistem MRAC terdapat dua *loop*, *Loop* pertama adalah *loop* umpan balik normal antara *output* proses dengan kontroler sedangkan *loop* kedua adalah *loop* yang digunakan untuk melakukan mekanisme pengaturan parameter kontroler. Pada Gambar 6 Model plan yang digunakan merupakan sistem yang diketahui dimana nilai yang dipakai adalah sistem yang ingin dikendalikan yaitu nilai *transfer function* atau model matematika yang didapat pada gerakan *open loop pitch* yang telah dilakukansebelumnya. Model referensi adalah sebuah model matematis atau fungsi yang digunakan sebagai panduan untuk mengatur respon sistem yang sebenarnya. Model referensi ini menggambarkan perilaku yang diinginkan atau target yang ingin dikendalikan. pada Gambar 6. Model referensi memiliki 1 masukan (r) yang dihubungkan dengan sinyal input sedangkan keluaran dari model referensi menjadi yang berupa error dengan plan menjadi masukan ke block diagram mekanisme adaptif. Nilai Model referensi yang digunakan dapat di lihat pada Gambar 6 dimana model referensi yang digunakan tersebut menggunakan sistem *black box* yang mengacu pada penelitian terdahulu yang dilakukan oleh [11] dimana frekuensi dan zeta digunakan untuk menentukan fungsi transfer. Model referensi ditampilkan sebagai respon transien yang merupakan sistem orde 2 yang dituliskan pada model matematika.



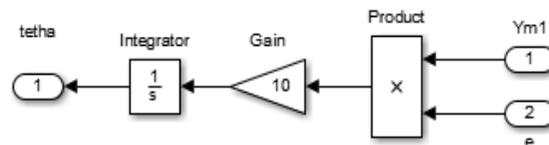
Gambar 6. Diagram block model referensi



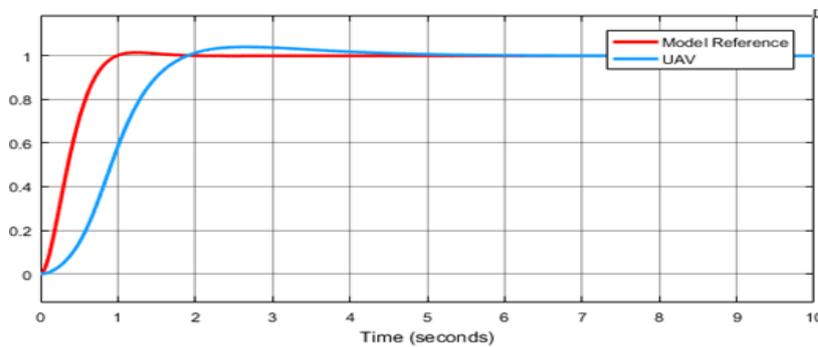
Gambar 7. Diagram Block PID

Pada diagram blog pada Gambar 7 pengendali atau sistem kontrol utama yang digunakan adalah nilai PID optimal yang telah *hexacopter* dimana nilai $P=14,4525$, $I=8,1981$, $D=6,3696$. Sebagaimana di lihat pada gambar 7 blog memiliki 2 input dimana inputan r dihubungkan dengan inputan model referensi atau sinyal input. Kemudian inputan y dihubungkan dengan error dan delta error keluaran model referensi dengan plan. Keluaran dari diagram PID tersebut adalah U yang beroperasi dengan keluaran mekanisme adaptif (theta). Dimana $U=U_{pid} * tetha$. Menurut aturan MIT, (e) menyajikan kesalahan antara keluaran model referensi, sedangkan tetha adalah parameter yang dapat disesuaikan.

Mekanisme adaptif merupakan bagian penting dari kontrol MRAC dimana mekanisme adaptif berfungsi untuk menyesuaikan parameter kontrol secara otomatis berdasarkan perbedaan antara respon aktual sistem dengan model referensi yang diinginkan. Mekanisme adaptif memungkinkan sistem untuk beradaptasi dengan dinamika yang berubah sehingga tetap dapat mengikuti perilaku yang diinginkan. Pada Gambar 8 menggambarkan blok mekanisme adaptif. Nilai gain hadir seperti yang diterapkan untuk melihat dari efek sistem, Dimana nilai gain yang digunakan adalah 100. *Output* dari tetha(θ) yang didasarkan pada kesalahan antara *output plant* (y) dan *output* model referensi (y_m).



Gambar 8. Diagram blok adaptive mekanisme



Gambar 9. Step respon Kendali MRAC

Gambar 9 merupakan grafik keluaran dari sistem kendali *model reference adaptive control* yang telah dirancang. Grafik berwarna kuning merupakan bentuk grafik model referensi sedangkan grafik berwarna biru merupakan grafik keluaran dari sistem MRAC, pada grafik keluaran Model referensi waktu dimulai dari 0 hingga mencapai 50% adalah 0,45 detik sedangkan keluaran MRAC mencapai waktu 50% 0,88 terdapat perselisihan waktu tapi tidak terlalu besar dan sudah cukup baik. Namun pada keluaran MRAC terjadi *overshoot* dari *setpoint* sebesar 4.0847% di waktu 2.6600 detik dan bisa kembali optimal di waktu 3.9351 detik. Sehingga Performa sistem dari keluaran MRAC pada Tabel 5.

Tabel 5. Performansi Sitem MRAC

No	Step respon	Nilai
1	Rise Time (T_r)	1,0709s
2	Delay time (T_d)	0,99s
3	Peak time (T_p)	2.6600 s
4	Settling Time (T_s)	3.9351 s
5	Persen Overshoot ($\%M_p$)	4.0847%

Berdasarkan Tabel 5 dapat di lihat dengan menambahkan sistem kendali *Model Reference Adaptive control* (MRAC) ke dalam control PID dapat memperbaiki *step respon* menjadi lebih baik, dan

memperbaiki respon yang lebih optimal dari percobaan yang dilakukan sebelumnya. Perbandingan performa sistem kendali MRAC dan *open loop* dapat dilihat pada tabel 6. Sistem kendali MRAC juga mampu beradaptasi dengan dengan perubahan yang terjadi dan mampu menyesuaikan dengan model referensi.

Tabel 6. Perbandingan Performa Sistem MRAC Dengan *Open Loop*

No	Step Respon	Respon MRAC	<i>Open Loop</i>
1	Rise Time (T_r)	1.0709 s	1,99 s
2	Delay time (T_d)	0,99 s	2,13 s
3	Peak time (T_p)	2.6600 s	4,34 s
4	Settling Time (T_s)	3.9351 s	9,56 s
5	Persen Overshoot ($\%M_p$)	4.0847 %	15,5%,
6	Steady state error	0.001 %	0,0445 %

4. KESIMPULAN

Dengan menggunakan sistem kendali PID respon yang diberikan sudah bagus namun masih ada overshoot yang cukup tinggi, parameter sitem mencapai nilai-nilai: rise time(t_s)= 1,99 s, delay time(t_d)= 2,13 s, peak time(t_p)=4,34 s, settling time(t_s) = 9,56 s, persen overshoot ($\%M_p$) =15,5% dan steady error state = 0,0445%. Untuk itu perlu mengoptimalkan performansi sistem dengan kendali MRAC. Dengan menggunakan sistem kendali Model reference Adaptive Control (MRAC) dengan kendali utama yang digunakan kendali PID respon yang diberikan sudah bagus dan sudah optimal, parameter sitem mencapai nilai-nilai: rise time(t_s)= 1,0709 s, delay time(t_d)= 0,99 s, peak time(t_p)=2,6600 s, settling time(t_s) = 3,9351 s, persen overshoot ($\%M_p$) =4,0847 dan steady error state = 0,0001%.

Unknownledman

Penelitian ini disponsori oleh Program Matching Found 2023, DIRJEN Vokasi Kemendikbud RI.

DAFTAR PUSTAKA

- [1] Rosadi. 2023. "Mengenal Sejarah Singkat Dan Jenis-Jenis Drone." *PRCF Indonesia*. Retrieved October 30, 2023 (<https://prcfindonesia.org/mengenal-sejarah-singkat-dan-jenis-jenis-drone/>).
- [2] Amin, Zulkifli, and Derry Meldi. 2018. "Pengidentifikasian Dan Pencarian Manusia Berbasis Citra Menggunakan Unmanned Aerial Vehicle." *METAL: Jurnal Sistem Mekanik Dan Termal* 2(2):50. doi: 10.25077/metal.2.2.50-60.2018.
- [3] Chiel, Hillel J., and Peter J. Thomas. 2011. *Special Issue on Applied Neurodynamics: From Neural Dynamics to Neural Engineering*. Vol. 8.
- [4] Adiyansyah, Januar, Ath Thuur, Noval Alam, Hadi Salam, Rikardus Patun, Muhammad Iqbal Zakaria, Faizal Fajri, David Julianto, and Wahyu Setyo Pambudi. 2023. "MODEL REFERENSI KONTROL ADAPTIF KETINGGIAN UDARA UNTUK." (2022):1968–75.
- [5] My, Andril. 2023. "ITP Gandeng PT Frogs Indonesia, Kembangkan Hexacopter Dalam Pengambilan Fotogrametri." *Infosumbar.Net*. Retrieved November 1, 2023 (<https://infosumbar.net/berita/itp-gandeng-pt-frogs-indonesia-kembangkan-hexacopter-dalam-pengambilan-fotogrametri/>).
- [6] Kusuma, A. P., A. Wajiansyah, and ... 2017. "Optimasi Pergerakan Robot Cerdas Berkaki Dengan Menerapkan Model Reference Adaptive Control." *Prosiding Snitt ...*
- [7] Nguyen, Ngoc Phi, Nguyen Xuan Mung, Le Nhu Ngoc Thanh Ha, and Sung Kyung Hong. 2022. "Fault-Tolerant Control for Hexacopter UAV Using Adaptive Algorithm with Severe Faults." *Aerospace* 9(6). doi: 10.3390/aerospace9060304.
- [8] Teddy Sudewo, Eka Iskandar, dan Katjuk Astrowulan. 2012. "Disain Dan Implementasi Kontrol PID Model Reference Adaptive Control Untuk Automatic Safe Landing Pada Pesawat UAV Quadcopter." *Jurnal Teknik Its* 1(1):78–83.

- [9] Hidayatullah, Ikhwan. 2022. "Peningkatan Performansi Sistem Kendali Hexacopter Dengan Sistem Pid Adaptif Tugas Akhir." (September).
- [10] Al, Al, Aswir Premadi, Asnal Effendi, and Andi Syofian. 2023. "Peningkatan Performansi Manuver Gerakan Roll Hexacopter Dengan Metoda Kendali Pid Adaptif." 1(2):75–82.
- [11] Suksmadana, Made Budi, Zaenudin Ramli, and Giri Wahyu Wiriasto. 2020. "Implementasi Dan Perancangan Sistem Kendali Posisi Ketinggian (Altitude) Quadcopter Berbasis Arduino Uno R3." *Dielektrika* 7(1):38. doi: 10.29303/dielektrika.v7i1.233.
- [12] Rothe, Julian, Jasper Zevering, Michael Strohmeier, and Sergio Montenegro. 2020. "A Modified Model Reference Adaptive Controller (M-MRAC) Using an Updated MIT-Rule for the Altitude of a UAV." *Electronics (Switzerland)* 9(7):1–15. doi: 10.3390/electronics9071104.
- [13] Matthews, MacKenzie T., and Sun Yi. 2019. "Model Reference Adaptive Control and Neural Network Based Control of Altitude of Unmanned Aerial Vehicles." *Conference Proceedings - IEEE SOUTHEASTCON 2019-April(1)*. doi: 10.1109/SoutheastCon42311.2019.9020447.
- [14] Mulyadi, Adi, Wijono, and Bambang Siswojo. 2020. "Kecepatan Motor Penggerak Quadcopter." *Jurnal Transmisi* 22(4):107–16.