

Evaluatie van PID- en Adaptieve Backstepping-Regelaars bij de Trajectvolgeregeling van een Quadrotorsysteem

Yi Yang*

December 2024

Abstract

Dit technische rapport presenteert de ontwikkeling en analyse van een wiskundig model voor een autonome quadrotor, met de nadruk op geavanceerde niet-lineaire regelstrategieën om de stabiliteit en prestaties te verbeteren. Twee verschillende regeltechnieken worden ontworpen en geïmplementeerd: een klassieke Proportioneel-Integrerend-Differentieel (PID) regelaar en een geavanceerdere niet-lineaire backstepping-benadering. De PID-regelaar, die veel wordt gebruikt vanwege zijn eenvoud en robuustheid, biedt een basiscontroleprestatie, terwijl de backstepping-techniek een verbeterde trajectvolgning en verstoringsonderdrukking mogelijk maakt via een systematisch Lyapunov-gebaseerd ontwerp. Om de effectiviteit van deze regeltechnieken te beoordelen, worden uitgebreide simulaties uitgevoerd onder verschillende operationele omstandigheden, waaronder externe verstoringen en dynamische vliegscenario's. De vergelijkende analyse biedt inzicht in de voordelen en beperkingen van elke regelstrategie en toont het potentieel van de backstepping-methode voor verbeterde autonomie van de quadrotor.

1 Inleiding

1.1 Achtergrond

Vliegende objecten hebben de mensheid al lange tijd gefascineerd en vormen de inspiratie voor voortdurende onderzoeks- en technologische vooruitgang. Recente ontwikkelingen op het gebied van sensoren, actuatoren, processors en energieopslagsystemen hebben de mogelijkheden van onbemande luchtvaartuigen (UAV's) aanzienlijk uitgebreid. Onder de verschillende UAV-platforms zijn quadrotors bijzonder aantrekkelijk vanwege hun unieke vliegeigenschappen, zoals de mogelijkheid om te zweven, verticaal op te stijgen en te landen, en met hoge wendbaarheid te manoeuvreren [3]. Deze kenmerken maken quadrotors zeer geschikt voor een breed scala aan toepassingen, waaronder gebouwbewaking, zoek- en reddingsmissies, en interventies in gevaarlijke of beperkte ruimtes. Nu de technologische drempels voor het bouwen en inzetten van quadrotors zijn verlaagd, hebben zowel de academische als de industriële gemeenschap zich intensief gericht op de ontwikkeling van

robuuste modellen, besturingsstrategieën en trajectplanningsmethoden [1, 7].

Ondanks hun voordelen zijn quadrotors van nature ondergereageerd en vertonen zij sterk gekoppelde, niet-lineaire dynamica. Dit brengt uitdagingen met zich mee bij nauwkeurige trajectvolgning en stabilisatie, vooral wanneer men te maken krijgt met windstoten, modelonzekerheden of problemen met de actuatoren/invoervertraging [7, 2]. Bovendien blijven geavanceerde taken zoals real-time trajectplanning in dynamisch veranderende omgevingen complex [4]. Eerder onderzoek heeft diverse benaderingen voor besturing aangetoond, variërend van klassieke proportionele-integraal-afgeleide (PID) regelingen tot meer verfijnde adaptieve, robuuste of intelligente regelkaders [20, 6, 8]. Hoewel PID-regelaars vaak geprezen worden vanwege hun eenvoud, kunnen zij moeite hebben bij ernstige modelonzekerheden of beperkingen zoals actuatorsverzadiging of grote externe storingen [10, 5].

Om deze beperkingen te overwinnen, verkent de literatuur diverse technieken. Zo richten robu-

uste backstepping-regelaars [10] zich op de niet-lineariteit door stapsgewijs regelwetten te construeren voor elke deel-dynamische vergelijking van de toestand van de quadrotor. Adaptieve methoden, zoals adaptieve fuzzy backstepping, zijn erop gericht onbekende parameters of externe storingen te compenseren [6], terwijl op neurale netwerken gebaseerde benaderingen proberen onzekerheden online te leren of te benaderen, met behoud van stabiliteitsgaranties [9]. De combinatie van backstepping met concepten zoals sliding-mode control of super-twisting sliding surfaces heeft ook aangetoond dat deze chattering kan verminderen en modelfouten kan opvangen [5, 7]. Ondertussen nemen output-feedback ontwerpen observatoren op om niet-gemeten toestanden te schatten, wat cruciaal is voor quadrotors met een beperkt aantal sensoren [26].

Een andere actieve onderzoekstak betreft visie-gebaseerde navigatie en besturing. Sommige quadrotors maken gebruik van monoculaire camera's en inertiële sensoren voor gelijktijdige lokalisatie en mapping. Zo wordt image-based visual servoing (IBVS) ingezet om referentieposities te volgen of objecten te detecteren in binnenomgevingen zonder GPS, zoals besproken door Zheng et al. [27]. Elders zijn data-gedreven detectiesystemen (bijv. Faster R-CNN) gecombineerd met volgalgoritmen zoals KCF (Kernelized Correlation Filters) gebruikt om meerdere doelen in dynamische scènes robuust te volgen [1]. Deze benaderingen benadrukken hoe perceptie en besturing steeds meer geïntegreerd worden, waardoor meer autonome vluchten mogelijk worden met minder afhankelijkheid van externe infrastructuur.

Naast integraal backstepping en intelligente observatoren zijn fractioneel-orderegelaars naar voren gekomen als een veelbelovend gebied, omdat zij in staat zijn complex systeemgedrag nauwkeuriger te modelleren dan modellen met gehele orde [14, 16, 15]. Fractionele calculus is ingezet bij het ontwerpen van regelaars met verbeterde robuustheid en afstemmingsflexibiliteit voor systemen die variëren van mechatronica tot het regelen van stationair toerental van verbrandingsmotoren [21, 18]. Specifiek voor quadrotors vergeleken Timis et al. [8] klassieke PID-regelaars met fractioneel-orderegelaars in een adaptieve structuur, waarmee werd aangetoond dat fractionele ontwerpen een betere gesloten-lus prestatie kunnen opleveren. Bovendien kan fractioneel backstepping-besturing onzekere quadrotor-

dynamica en onbekende tijdvertragingen opvangen, terwijl een nauwkeurige trajectvolgning wordt bereikt [17].

Andere studies passen fractioneel-PID-regelaars toe in combinatie met optimalisatie-algoritmen, zoals particle swarm optimization (PSO) of genetische algoritmen, om de afstemmingsparameters automatisch te optimaliseren [19, 12]. In sommige gevallen leiden geavanceerde of gecombineerde optimalisatieprocessen, zoals multi-objective multi-verse optimization, tot trajectplanningsoplossingen voor UAV's die botsingen met bewegende obstakels vermijden [4]. Evenzo kan robuuste besturing worden gerealiseerd via nieuwe schattingsmethoden, zoals onzekerheids- en storingsschatters, die externe windstoten of ongemodelleerde dynamica kunnen opvangen [2]. De synergie van deze methodologieën—fractionele besturing, robuuste backstepping, adaptieve fuzzy of neurale netwerkaanpakken—verlegt voortdurend de grenzen van prestaties, betrouwbaarheid en autonomie in UAV-toepassingen.

Ondertussen is er ook aandacht voor de modellering. Onderzoekers zoals Yang en Zhang [13] hebben een reeks fractioneel-ordemodeller- en besturingstechnieken bestudeerd die toepasbaar zijn in de techniek, wat heeft geresulteerd in bredere behandelingen van fractionele calculus [14] en de geavanceerde toepassingen ervan in mechatronica en robotica [19]. Inderdaad, fractionele benaderingen zijn uitgebreid naar meer complexe taken, zoals het regelen van stationair toerental in auto's [21], of gecombineerd met geavanceerde identificatieprocedures voor roterende omgekeerde slingers en andere benchmarksystemen [12]. Deze verkenningen dienen als opstapjes voor verbeterde UAV-modellering, met name in onzekere of tijdvarierende contexten. Daarnaast beginnen multi-layer spiking neurale netwerken een rol te spelen in UAV-besturingskaders, waarbij nieuwe trainingsparadigma's zoals fractioneel spike-timing-dependent gradient descent worden geïntroduceerd [17], wat een opkomende kruising weergeeft tussen neuromorfe computing en UAV-begeleiding.

Ten slotte vereisen praktische beperkingen vaak dat aan de eisen op het gebied van grootte, gewicht en vermogen (SWaP) wordt voldaan. Zoals beschreven in diverse technische rapporten en proefschriften [22, 23, 25, 24, 11], blijven onderzoekers zowel de integratie van hardware als software voor quadrotors verfijnen, met als doel de over-

gang van simulatie- of laboratoriumdemonstraties naar veldtoepassingen. De combinatie van robuuste of adaptieve backstepping, fractionele besturing en geavanceerde toestandsobservatoren vormt een cruciaal pad naar de bouw van de volgende generatie luchtrobots die met minimale menselijke interventie door krappe binnenruimtes of uitdagende buitenomgevingen kunnen navigeren. Samenvattend blijft de zoektocht naar nauwkeurige, stabiele en robuuste besturing van quadrotors een actief en vruchtbaar onderzoeksgebied, dat put uit innovatieve regeltheorieën en computationele intelligentie om te voldoen aan de eisen van moderne luchtvaarttoepassingen.

1.2 Aerodynamische Modelleren van Quadrotor-luchtvaartuigen

Het model zal 12 staten hebben: verplaatsing in drie richtingen (x, y, z), snelheid in drie richtingen ($\dot{x}, \dot{y}, \dot{z}$), rolhoek (ϕ), pitchhoek (θ), yawhoek (ψ) en drie hoeksnelheden ($\dot{\phi}, \dot{\theta}, \dot{\psi}$) met betrekking tot deze drie hoeken. Laat $[\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, z, \dot{z}, x, \dot{x}, y, \dot{y}] = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]$. Het oorspronkelijke niet-lineaire dynamische model voor dit systeem wordt als volgt gegeven:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{(I_y - I_z)}{I_x} x_4 x_6 + \frac{U_2 l}{I_x} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{(I_z - I_x)}{I_y} x_2 x_4 + \frac{U_3 l}{I_y} \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{(I_x - I_y)}{I_z} x_2 x_4 + \frac{U_4}{I_z} \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \cos(x_1) \cos(x_3) \frac{U_1}{m} - g \\ \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= (\sin(x_3) \cos(x_1) \cos(x_5) + \sin(x_1) \sin(x_5)) \frac{U_1}{m} \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= (\sin(x_3) \cos(x_1) \sin(x_5) - \sin(x_1) \cos(x_5)) \frac{U_1}{m} \end{aligned}$$

waarbij:

$$\begin{aligned} U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= b(\Omega_4^2 - \Omega_2^2) \\ U_3 &= b(\Omega_3^2 + \Omega_1^2) \\ U_4 &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{aligned}$$

We kiezen de systeemmodelparameters in Table 1.

2 Systeemontwerp op basis van PID-methode

2.1 Ontwerp de PID-controller

Uit de aerodynamische vergelijkingen van quadrotors weten we dat de houdingshoeken en hoeksnelheden onafhankelijk zijn van translatiesnelheid. Echter, de translatiebewegingen zijn afhankelijk van hoekbeweging. Daarom kunnen we het hele systeem verdelen in twee subsystemen: lineair bewegingssubstelsysteem en hoekbewegingssubstelsysteem, ze hebben semi-koppeling relaties zoals geïllustreerd door de volgende figuur:

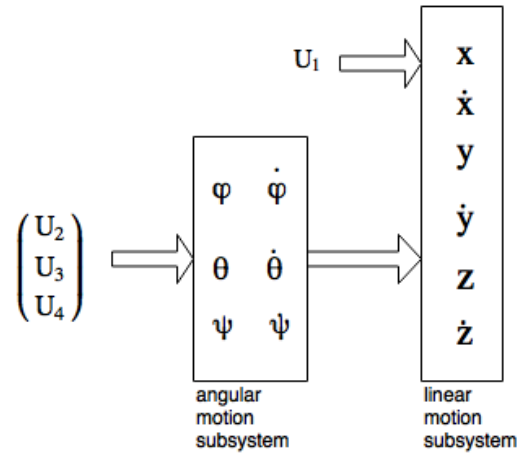


Figure 1: Systeem met controller

Volgens deze semi-gekoppelde relatie kunnen we eerst de binnenlus (houdingscontrole) ontwerpen en vervolgens de buitenlus (positiecontrole), zoals getoond in Figuur2b. De binnenlus is om de houdingshoeken te regelen en de stabiliteit te garanderen wanneer de vlucht stabiel vliegt of op een vaste positie blijft. De buitenlus is om de vlucht te control-

g	m	L	I_x	I_y	I_z	d	b
9.81	0.53	0.232	$6.228e-3$	$6.228e-3$	$1.121e-2$	$7.5e-7$	$3.13e-5$

Table 1: Geometrische parameters van het quadrotorsysteem.

eren om in onze doeltraject te vliegen en uiteindelijk ervoor te zorgen dat de quadrotor ons doelbestemming bereikt. (x_d, y_d, z_d) is de referentiedoelpositie, $(\phi_d, \theta_d, \psi_d)$ is de verwachte waarde van ons binnenlus-subsysteem.

Eerst vereenvoudigen we de dynamische vergelijking

$$\begin{aligned} \dot{x}_{10} &= U_x \frac{U_1}{m} \\ \dot{x}_{12} &= U_y \frac{U_1}{m} \end{aligned}$$

waarbij:

$$\begin{aligned} U_x &= \sin(x_3)\cos(x_1)\sin(x_5) - \sin(x_1)\cos(x_5) \\ U_y &= \sin(x_3)\cos(x_1)\cos(x_5) + \sin(x_1)\sin(x_5) \end{aligned}$$

De XYZ-controller gebruikt de informatie van feedback en gewenste x, y, z om de gewenste U_{xd} en U_{yd} te bepalen, vervolgens wordt U_{xd} en U_{yd} gebruikt om de gewenste $x_{1d}(\phi)$ en $x_{3d}(\theta)$ te berekenen, waarbij:

$$\begin{aligned} x_{1d} &= \arcsin(u_x \sin(x_5) - u_y \cos(x_5)) \\ x_{3d} &= \arcsin\left[\frac{u_x - \sin(x_1)\sin(x_5)}{\cos(x_1)\cos(x_5)}\right] \end{aligned}$$

Gebruik gewenste x_{1d} en x_{3d} als referentie voor de hoekcontroller om de gewenste U_{1d} U_{2d} U_{3d} en U_{4d} te berekenen. Bereken ten slotte de benodigde spanning, wat de echte input is.

De PID-parameter die we hebben gekozen wordt getoond in Tabel 2. Deze parameters worden gebruikt in het continue geval en in het geval van 0,05s bemonstering. Echter in het geval van 0,1s bemonstering worden de parameters van de hoek-PID-controller gewijzigd om stabiel te blijven.

Gebruik het lineaire systeem dat we de openlus Bode-plot plotten voor de phi, zie Figuur 3 De versterkingsmarge is 24,1 db, de fasemarge is 46,4 graden, wat bevredigend is.

	K_p	K_i	K_d
ϕ	0.5	0.1	0.15
θ	0.5	0.1	0.15
ψ	0.1	0.05	0.1
x	2	0.25	3
y	2	0.35	3
z	1	0.2	1

Table 2: Parameters voor PID-controller

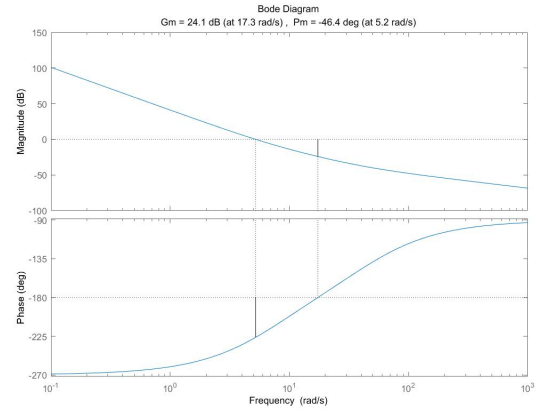
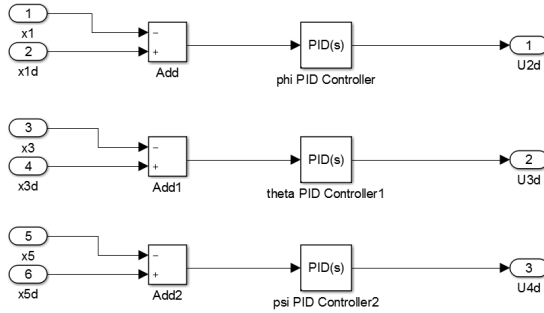


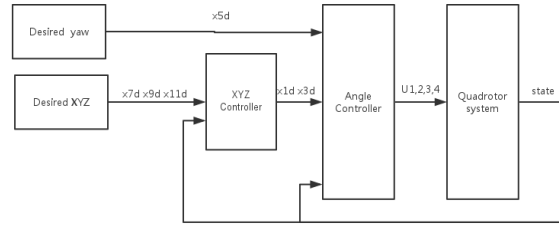
Figure 3: Bode-plot van het lineaire systeem

2.2 Simulatieresultaten en analyse

We testen eerst de controller in het continue systeem. De beginvoorwaarde is ingesteld op $\phi = 0.4$, $\theta = 0.3$, $\psi = 0.2$, $x = 0$, $y = 0$, $z = 0$. De bestemming is $x = 1$, $y = 1.5$, $z = 2$. Het resultaat wordt getoond in Figuur 4. In feite kan een meer optimale keuze van de PID-parameters worden gekozen voor het continue systeem, maar voor consistentie hebben we de optimale parameters gekozen die het discrete systeem van 0,05s stabiel kunnen maken. Figuur 5 toont het resultaat van bemonsteringstijd 0,05s. Het kan worden waargenomen dat de hoeken meer oscilleren en een hogere amplitude hebben. Wat betreft posities wordt oscillatie geïntroduceerd tijdens het stijgen. Het voordeel van deze discrete controller is dat de stijgtijd en de overshoot van x y worden verminderd. Figuur 6 toont het resultaat van bemon-



(a) Voorbeeld van PID-controle



(b) Een vereenvoudigde kaart van hoe het controlesysteem werkt

Figure 2: PID-regelsysteem

steringstijd 0,1. Het stabiele bereik van parameters kromp naarmate de bemonsteringstijd toenam, dus we kozen een nieuwe set parameters voor dit geval, de prestaties zijn beperkt in vergelijking met eerdere gevallen omdat elke agressieve keuze van parameters onstabiel zou veroorzaken.

We voeren meer tests uit in het geval van bemonsteringstijd 0,05s. Eerst voegen we verstoringen toe aan het systeem. Een tegenwind van 2 m/s treedt op om 15 s en eindigt om 20 s, een extra gewicht van 0,5 kg wordt toegevoegd aan de quadrotor om 15 s en wordt niet verwijderd tot het einde. Het resultaat wordt getoond in Figuur 7. Er kan worden gezien dat de PID-controller deze twee gevallen met succes afhandelt. Echter, de controller faalt wanneer de windsnelheid 18 m/s bereikt. Er is geen beperking voor het extra gewicht in onze simulatie. Echter, de overshoot zal erg groot zijn als het extra gewicht te hoog is. Het is ook bekend dat in het echte leven het extra gewicht beperkt is door de capaciteit van de motoren.

Vervolgens voegen we ruis toe aan onze feedbacks, de gaussische ruis is voornamelijk beperkt tot 0,03 m en 0,03 rad. Figuur 8 toont het resultaat. Uit de plot kunnen we observeren dat z niet significant wordt beïnvloed door de ruis, omdat de ruisinval in vier rotors wordt geneutraliseerd. Het convergentieproces van x en y blijft in principe hetzelfde, behalve dat ze de ruis behouden na zich te vestigen. In het geval van phi en theta wordt de ruis versterkt in de steady-state respons.

2.3 Samenvatting

Concluderend, de PID-controller presteert goed in zowel het continue als het discrete geval. Het toont ook robuustheid onder verstoringen en ruis.

	K_p	K_i	K_d
ϕ	0.5	0.1	0.15
θ	0.5	0.1	0.15
ψ	0.1	0.05	0.1
x	2	0.25	3
y	2	0.35	3
z	1	0.2	1

Table 3: Parameters voor PID-controller

3 Systeemontwerp op basis van terugkoppelmethode

Lineaire feedback is de meest voorkomende controllerontwerpmethode voor MIMO-systeem. Het abstraheert lineaire ingrediënten uit het niet-lineaire model door linearisatie. Om het systeem met hoge precisie te besturen, vereist het dat het systeem sterk lineaire kenmerken heeft. Integendeel, terugkoppelmethode hoeft het niet-lineaire deel niet te elimineren, wat het efficiënt maakt bij het besturen van niet-lineaire systemen. In dit hoofdstuk zullen we een methode genaamd terugkoppeling tonen en een uitdrukking van de regelvariabelen afleiden. Vervolgens proberen het quadrotor-systeem te besturen met de terugkoppelmethode.

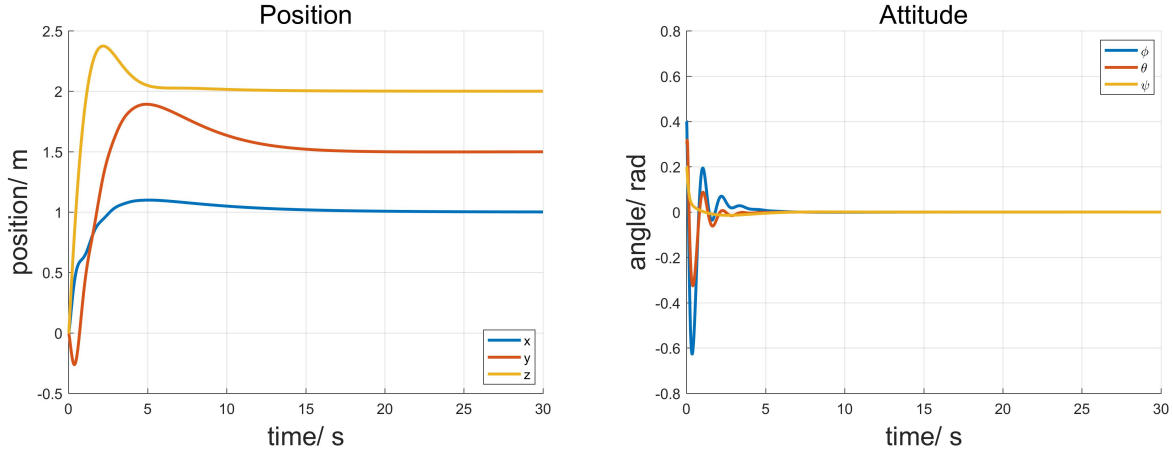


Figure 4: PID-control in continue tijd

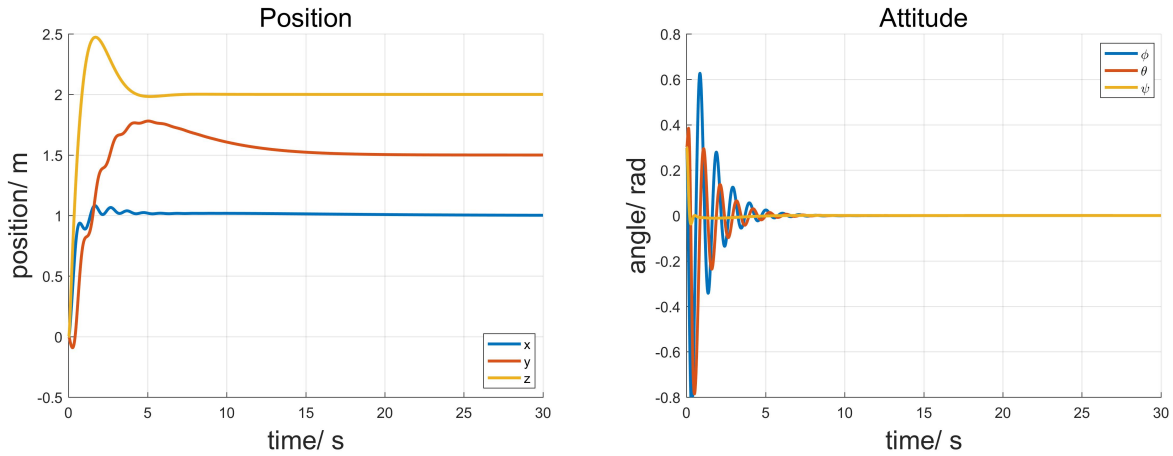


Figure 5: PID-control bij bemonsteringstijd=0.05s

3.1 Ontwerp van terugkoppelingscontroller

3.1.1 Simulatie van terugkoppelingscontrole met vereenvoudigde inputs

Volgens Bijlage A.1 en A.1.1, konden we een terugkoppelingscontroller in Matlab Simulink bouwen zoals getoond in Figuur 9.

In de volgende simulatie testen we de terugkoppeling, een niet-lineaire regelmethode. We simuleren het systeem in continue tijd, proberen de quadrotor te besturen van punt (0,0,0) naar punt (1,1.5,2). Totale simulatietijd is 30 seconden met de beginconditie $X = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}]^T = [0.4, 0, 0.3, 0, 0.2, 0, 0, 0, 0, 0, 0, 0]^T$. Bovendien hebben parameters c_1 — c_{12} ook invloed op de simu-

latieresultaten. Als c_1 — c_{12} groot wordt, wordt de regelcapaciteit sterker, evenals de regelfout. Aan de andere kant, als c_1 — c_{12} kleiner wordt, gebeurt oscillatie en overshoot vaker. In de resultaten getoond in Figuur 10, kiezen we $c_1 = 10; c_2 = 3; c_3 = 5; c_4 = 8; c_5 = 3; c_6 = 2; c_7 = 2; c_8 = 2; c_9 = 2; c_{10} = 2; c_{11} = 2; c_{12} = 2$; (Lezers kunnen eerst **run_backstepping.m** bestand draaien, dan **backstepping.slx** bestand)

Simulatieresultaat in Figuur 10 lijkt vrij bevredigend. Het hele systeem wordt stabiel in ongeveer 3 seconden zonder overshoot langs de x,y,z assen.

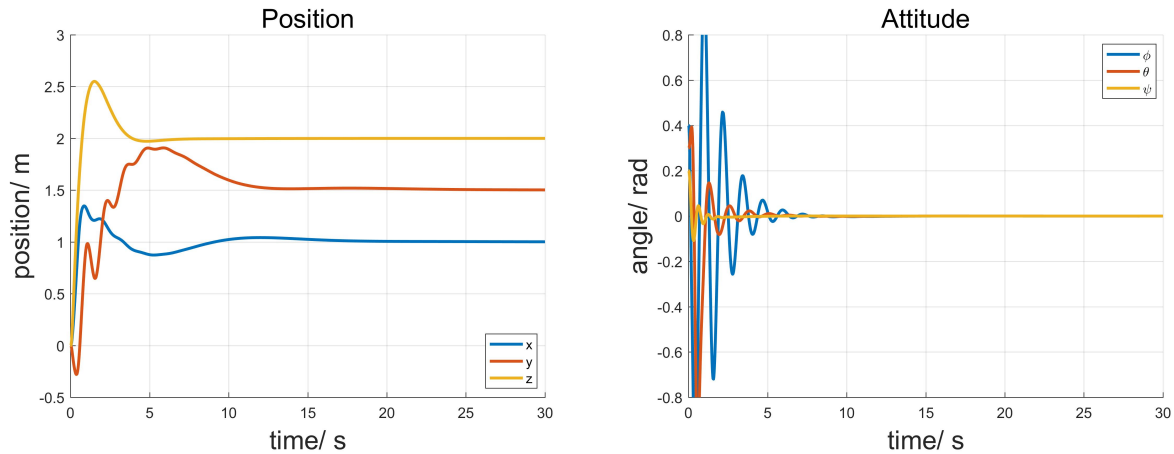


Figure 6: PID-controle bij bemonsteringstijd=0.1s

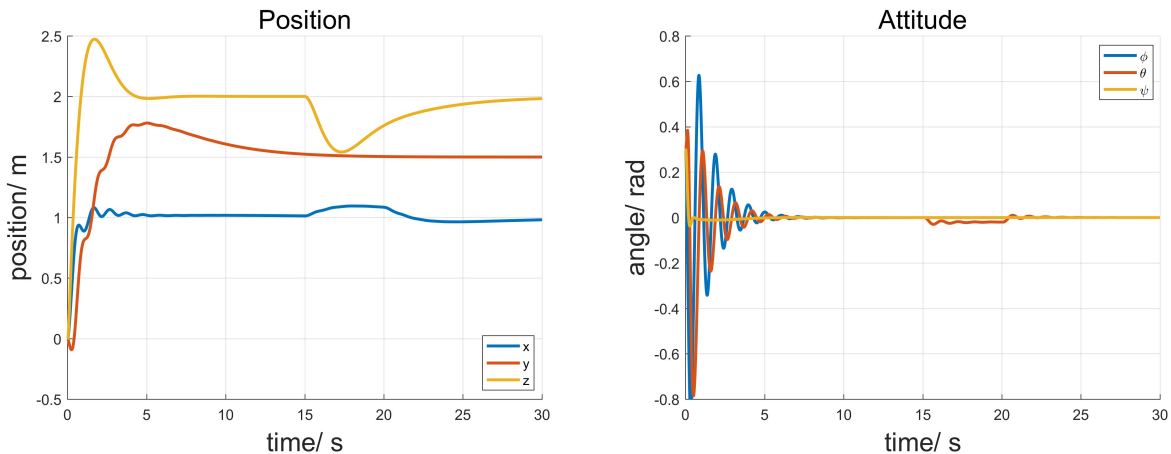


Figure 7: PID-controle in discrete tijd onder verstoring

3.1.2 Simulatie van terugkoppelingscontrole met echte inputs

In praktische gevallen zijn de enige inputs die we direct kunnen bereiken de spanningen van vier rotors. In de volgende simulatie hebben we de gewenste inputs van rotorspanning opgelost met een "U trans V" subblok en hebben we de overdrachtsfunctie van de elektrische rotor beschouwd als $\frac{800}{0.01s+1}$. De rotatiesnelheid van rotors groeit met 800r/s als de inputspanning met 1V toeneemt. Beginvoorwaarde en parameters blijven ongewijzigd. (Lezers kunnen eerst `run_backstepping.m` bestand draaien, dan `backstepping_real.slx` bestand)

Van Figuur 11 merkten we een klein verschil op van de echte situatie naar de vereenvoudigde conditie. Het duurde ongeveer 2 seconden voor de

quadrotor stabiel bleef na tijd 0. Gelukkig, hoewel de rotor een kleine vertraging van de rotatiesnelheid had, werd de responstijd van het systeem nog steeds gehouden op ongeveer 3 seconden.

3.1.3 Simulatie van terugkoppelingscontrole in discrete tijd

Zoals we weten, werken controllers in praktische onderzoeken onder discrete tijd. Figuur 12 en Figuur 13 testten de regelresultaten bij verschillende bemonsteringstijden, 0.05s en 0.1s. (Lezers kunnen eerst `run_backstepping_discrete.m` bestand draaien en de parameter `samplingtime` wijzigen, dan `backstepping_real_discrete.slx` bestand)

Zoals te zien in die foto's, treedt trilling op wan-

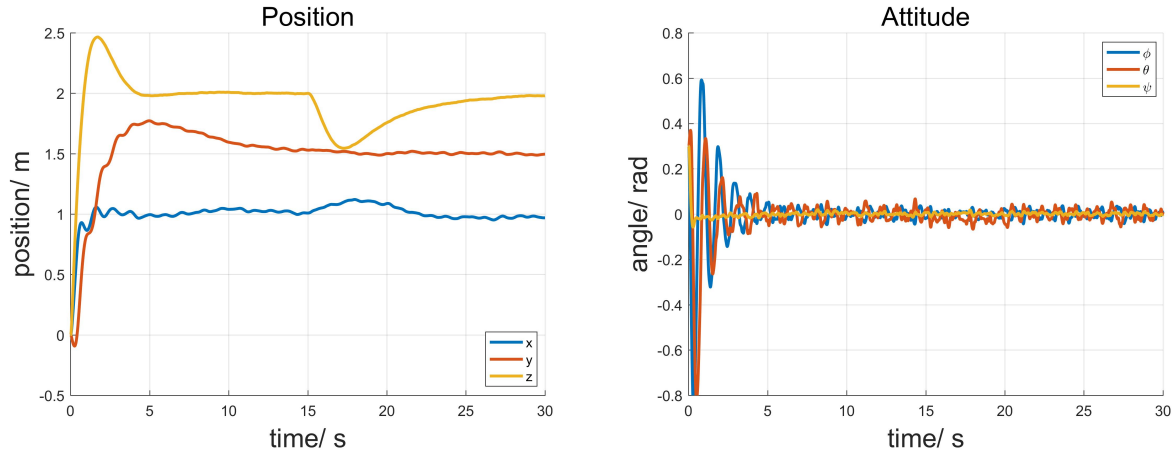


Figure 8: PID-control met ruis

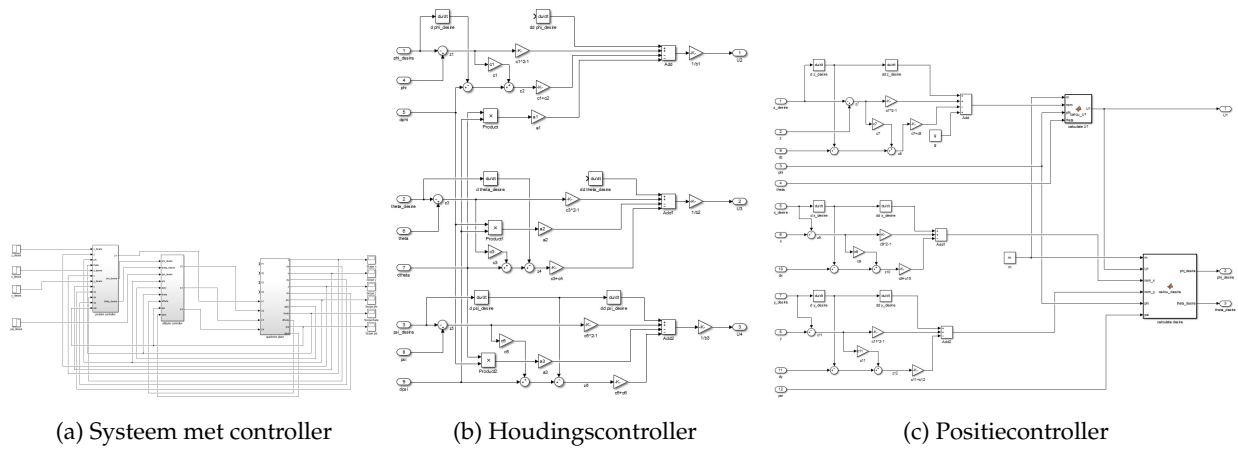


Figure 9: Terugkoppingsregelsysteem

neer de bemonsteringstijd wordt gewijzigd. De amplitude van trilling is ongeveer $0.01\text{rad} \approx 0.573\text{deg}$ maximaal, wat acceptabel is na alles. Om hetzelfde tempo van PID-control te houden, kiezen we bemonsteringstijd=0.05s in de volgende secties.

3.1.4 Terugkoppelingscontrole in discrete tijd onder verstoring

Om de robuustheid van terugkoppelingscontrole te testen, pasten we verschillende verstoringen toe op de quadrotor. Een horizontale wind van 2 m/s op de x-as begint om $t=15\text{s}$ en eindigt om $t=20\text{s}$. Een extra gewicht $m=0.5\text{kg}$ zal op de quadrotor worden toegevoegd bij $t=15$ en gehandhaafd tot $t=30\text{s}$. Simulatieresultaten worden getoond in Figuur 14. (Lezers kunnen

eerst `run_backstepping_discrete_disturb.m` bestand draaien en vervolgens `backstepping_real_discrete_disturb.slx` bestand)

De simulatietijd in Figuur 14 is 30 seconden. We merken op, van de plot van z , dat het systeem stabiel zal zijn nadat de verstoring ongeveer 3 seconden had plaatsgevonden en op het nieuwe stabiele punt blijft. Vanwege de plot van positie x , zal de quadrotor niet terugvliegen naar punt (1,1.5,2) zolang de verstoring bestaat. Om het systeem aan te passen aan de verstoringen, zal een Adaptieve Terugkoppelingsbenadering worden afgeleid in de volgende subsectie.

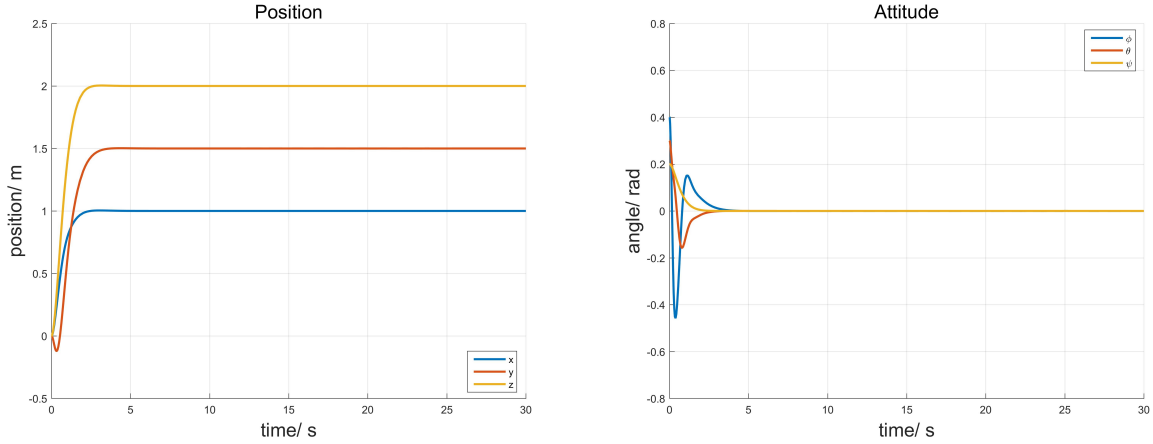


Figure 10: Terugkoppelingscontrole in continue tijd

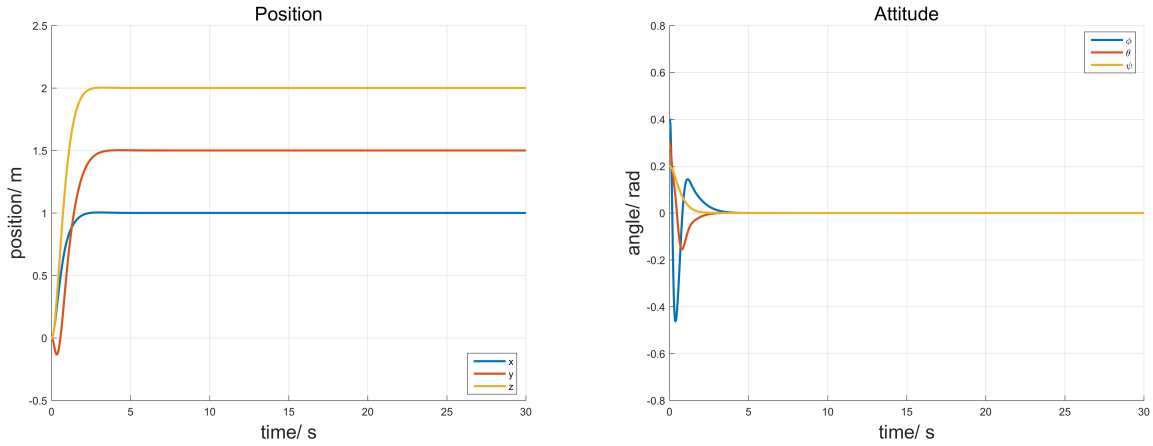


Figure 11: Terugkoppelingscontrole met inputs $V1, V2, V3, V4$ in continue tijd

3.1.5 Simulatie van Adaptieve Terugkoppelingscontrole in discrete tijd onder verstoringen

We kunnen het probleem van onbekende verstoringen oplossen nadat we een Adaptieve Terugkoppelingscontroller hebben gebouwd zoals beschreven in Bijlage A.2.

Figuur 15 toont het simulatieresultaat van Adaptieve Terugkoppelingscontrole wanneer verstoringen worden toegepast, d.w.z. een horizontale wind van 2 m/s op de x-as begint om $t=15s$ en eindigt om $t=20s$, een onbekend extra gewicht $m=0.5kg$ toegevoegd aan de quadrotor om $t=15$ en gehandhaafd tot $t=30s$. Uit de plot

van houding kunnen we concluderen dat adaptieve terugkoppeling sneller convergeert dan de niet-adaptieve. Bovendien schat adaptieve controle met succes de onbekende verstoring en vliegt terug naar de gewenste positie. (Lezers kunnen eerst **run_backstepping_discrete_disturb.m** bestand draaien en vervolgens **backstepping_real_discrete_disturb_adp.slx** bestand)

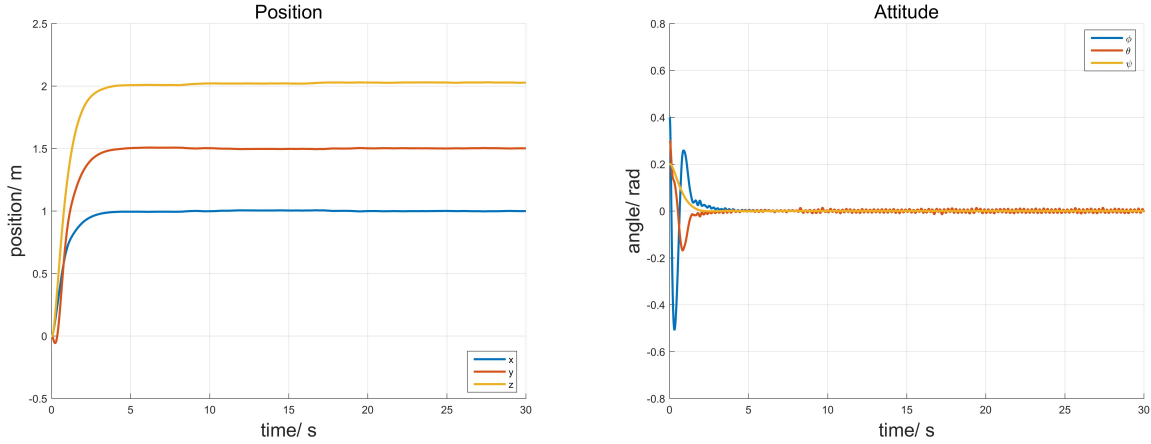


Figure 12: Terugkoppelingscontrole bij bemonsteringstijd=0.05s

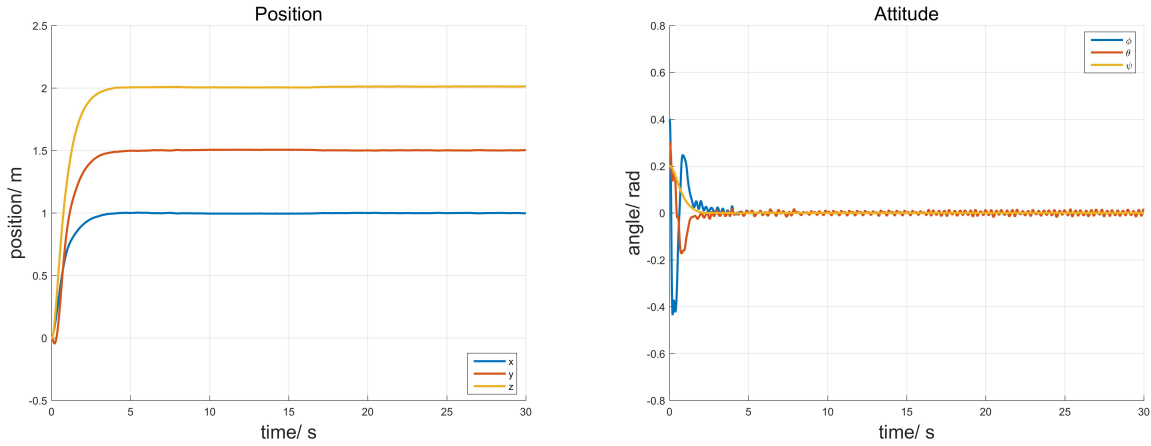


Figure 13: Terugkoppelingscontrole bij bemonsteringstijd=0.1s

3.1.6 Simulatie van Adaptieve Terugkoppelingscontrole met ruis

In de volgende simulatie worden verstoringen en andere parameters gehouden zoals voorheen. Bovendien wordt ruis met amplitude ≈ 0.6 toegevoegd in $\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\phi}, \ddot{\theta}, \ddot{\psi}$. Merk op dat, in dit geval, ruis ook terug zal worden toegevoegd aan het systeem zelf, wat kan worden beschouwd als een combinatie van bewegingsfout en sensorfout. We kunnen ontdekken van het simulatieresultaat (Figuur 16) dat trilling zal optreden vanwege ruis. De amplitude van trilling is ongeveer 0.005m langs x en y as en $0.01\text{rad} \approx 0.05\text{deg}$ op rollhoek ϕ en pitchhoek θ . (Lezers kunnen

eerst `run_backstepping_discrete_disturb.m` bestand draaien en vervolgens `backstepping_real_discrete_disturb_adp_noise.slx` bestand)

4 Vergelijking

Van de vergelijking kunnen we vinden dat de Terugkoppeling een meer acute reactie heeft dan PID. Wat meer is, de Terugkoppeling heeft minder overshoot, wat betekent dat het in werkelijkheid minder kans heeft om obstakels te raken die niet in het verwachte traject zitten. De oscillatiefrequentie en amplitude van hoeken in Terugkoppeling zijn

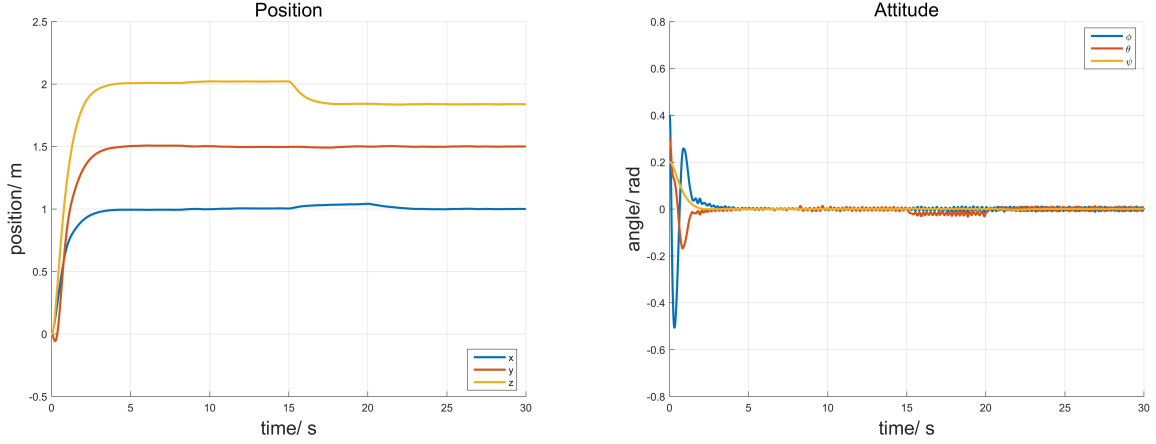


Figure 14: Terugkoppelingscontrole in discrete tijd onder verstoring

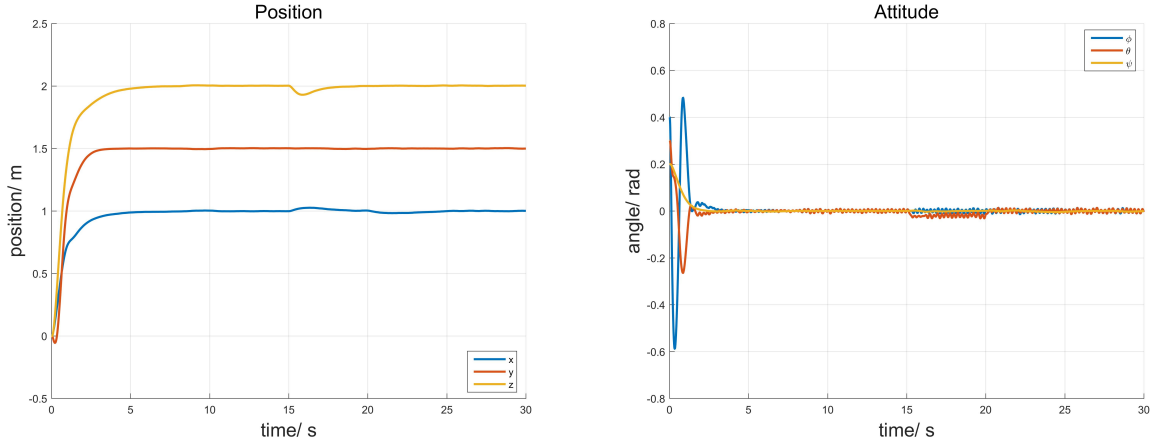


Figure 15: Adaptieve Terugkoppelingscontrole in discrete tijd onder verstoring

ook minder dan die in PID.

To further advance the field, future research in adaptieve backstepping-regeling kan worden uitgebreid naar fractionele-orde systemen, waarbij gebruik wordt gemaakt van recente ontwikkelingen en toepassingen van fractionele calculus. Voor een overzicht van relevante literatuur wordt verwezen naar.

A Terugkoppelingscontrole

A.1 Terugkoppelingshoudingscontrole

Zoals getoond in hoofdstuk twee, zal de houdingscontroller de positiecontroller niet beïnvloeden. Daarom konden we drie Terugkoppelingscontrollers ontwerpen om de drie hoeken ϕ, θ, ψ respectievelijk te besturen. Neem de rolhoek ϕ als voorbeeld, we konden de uitdrukking van U_2 afleiden door de Terugkoppelingsmethode:

$$\begin{cases} \dot{x}_1 &= \dot{\phi} &= x_2 \\ \dot{x}_2 &= a_1 \dot{\theta} \dot{\psi} + b_1 U_2 &= a_1 x_4 x_6 + b_1 U_2 \end{cases}$$

waarbij $a_1 = (I_y - I_z)/I_x$, $b_1 = L/I_x$.

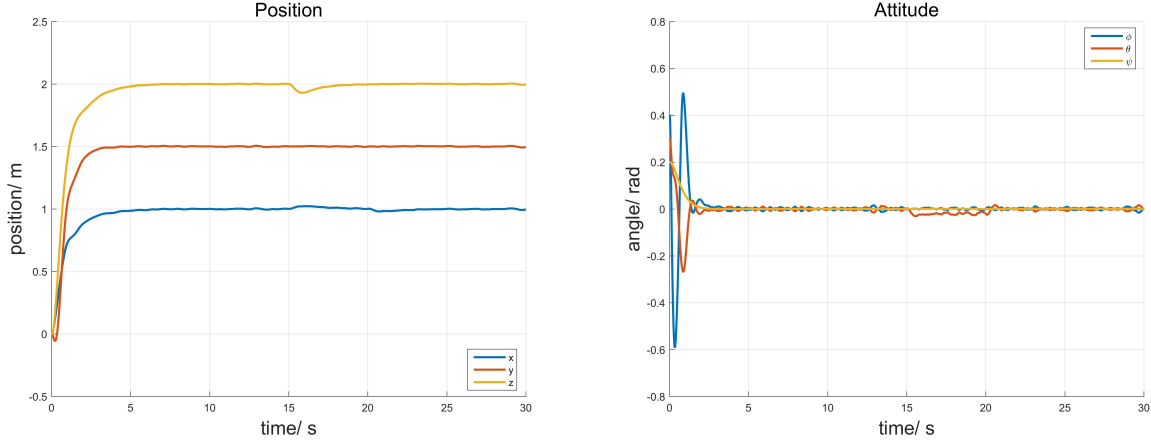


Figure 16: Adaptieve Terugkoppelingscontrole met sensorruis

Reactie-eigenschappen	PID	Terugkoppeling
tijd tot de verwachte hoogte	5s	4s
tijd tot de doelhorizontale positie	15s	4s
tijd om te herstellen van verstoring	10s	5s

Table 4: Prestatie van verschillende methoden

Stap 1

Noem de gewenste rolhoek als $\phi_d = x_{1d}$, en de foutvariabele $z_1 = x_1 - x_{1d}$. Neem vervolgens de afgeleide van z_1 , we hebben:

$$\dot{z}_1 = \dot{x}_1 - \dot{x}_{1d} = x_2 - \dot{x}_{1d}$$

Dan definiëren we een virtuele regelvariabele α_1 , zodat $z_2 = x_2 - \alpha_1$, en kiezen de Lyapunov-functie $V_1 = \frac{1}{2}z_1^2$. Dus:

$$\dot{V}_1 = z_1 = z_1(x_2 - \dot{x}_{1d}) = z_1(z_2 + \alpha_1 - \dot{x}_{1d})$$

Volgens het Lyapunov-stabiliteitscriterium moeten we $\alpha_1 = \dot{x}_{1d} - c_1 z_1$ kiezen zodat $\dot{V}_1 < 0$, waarbij $c_1 > 0$. Dan kunnen we hebben:

$$\begin{aligned}\dot{V}_1 &= -c_1 z_1^2 + z_1 z_2 \\ z_2 &= x_2 - \dot{x}_{1d} + c_1 z_1\end{aligned}$$

We kunnen zien dat als $z_2 = 0$, dan is z_1 asymptotisch stabiel. Aangezien, z_2 is niet altijd nul, we hebben een andere regelvariabele nodig voor $z_1 z_2$.

Stap 2

Definieer Lyapunov-functie $V_2 = \frac{1}{2}z_2^2 + V_1$,

aangezien $\dot{z}_2 = \dot{x}_2 - \ddot{x}_{1d} + c_1 \dot{z}_1$, we hebben:

$$\begin{aligned}\dot{V}_2 &= z_2 \dot{z}_2 + V_1 = z_2(\dot{x}_2 - \ddot{x}_{1d} + c_1 \dot{z}_1) - c_1 z_1^2 + z_1 z_2 \\ &= z_2(b_1 U_2 + a_1 x_4 x_6 - \ddot{x}_{1d} + c_1 \dot{z}_1) - c_1 z_1^2 + z_1 z_2 \\ &= z_2(b_1 U_2 + a_1 x_4 x_6 - \ddot{x}_{1d} + c_1(z_2 - c_1 z_1)) - c_1 z_1^2 + z_1 z_2\end{aligned}\quad (1)$$

Dan zou de regelininput moeten zijn:

$$\begin{aligned}U_2 &= \frac{1}{b_1}(\ddot{x}_{1d} - z_1 - a_1 x_4 x_6 - c_1 z_2 + c_1^2 z_1 - c_1 z_2) \\ &= \frac{1}{b_1}(\ddot{x}_{1d} + (c_1^2 - 1)z_1 - a_1 x_4 x_6 - (c_1 + c_2)z_2)\end{aligned}\quad (2)$$

waarbij $c_2 > 0$, $\ddot{x}_{1d} = 0$, $z_1 = x_1 - x_{1d}$, $z_2 = x_2 - \dot{x}_{1d} + c_1 z_1$. Zet (2) in (1), we hebben:

$$\dot{V}_2 = -c_1 z_1^2 - c_2 z_2^2 < 0$$

Volgens het Lyapunov-stabiliteitscriterium, kan U_2 garanderen dat z_1 en z_2 asymptotisch stabiel zijn.

Zelfde redenering, we konden de regelvariabele voor pitchhoek θ en yawhoek ψ afleiden als:

$$\begin{cases} U_3 = \frac{1}{b_2}[\ddot{x}_{3d} + (c_3^2 - 1)z_3 - a_2 x_2 x_6 - (c_3 + c_4)z_4] \\ U_4 = \frac{1}{b_3}[\ddot{x}_{5d} + (c_5^2 - 1)z_5 - a_3 x_2 x_4 - (c_5 + c_6)z_6] \end{cases}$$

waarbij:

$$\begin{aligned} z_3 &= \theta - \theta_d = x_3 - x_{3d} \\ z_4 &= x_4 - \dot{x}_{3d} + c_3 z_3 \\ z_5 &= \psi - \psi_d = x_5 - x_{5d} \\ z_6 &= x_6 - \dot{x}_{5d} + c_5 z_5 \end{aligned}$$

A.1.1 Terugkoppelingspositiecontrole

Positiecontrole is een combinatie van hoogtecontrole en horizontale controle. De hoogte z wordt bestuurd door input U_1 , en horizontale positie x, y worden niet direct bestuurd door inputs. Aangezien x, y direct gecorreleerd zijn met ϕ en θ , konden we enkele virtuele input U_x, U_y , definiëren, uitgedrukt door ϕ en θ , parallel aan U_1 . Neem de hoogtecontroller als voorbeeld, we konden een Terugkoppelingscontroller ontwerpen.

De dynamische vergelijkingen van z en $\dot{z}$ zijn:

$$\begin{cases} \dot{x}_7 &= x_8 \\ \dot{x}_8 &= -g + (\cos(x_1)\cos(x_3)) \frac{U_1}{m} \end{cases} \quad (3)$$

Stap 1

Voor een gegeven gewenste hoogte $z_d = x_{7d}$, definieer de foutvariabele $z_7 = x_7 - x_{7d}$ en bereken de afleiding ervan:

$$\dot{z}_7 = \dot{x}_7 - \dot{x}_{7d} = x_8 - \dot{x}_{7d}$$

Dan, definieer α_7 , waar $z_8 = x_8 - \alpha_7$, en kies de Lyapunov-functie $V_7 = \frac{1}{2}z_7^2$, dan:

$$\dot{V}_7 = z_7 \dot{z}_7 = z_7(x_8 - \dot{x}_{7d}) + z_7 z_8$$

Om te zorgen dat z_7 asymptotisch stabiel is bij de oorsprong, volgens het Lyapunov-stabiliteitscriterium, moeten we $\dot{V}_7 < 0$ hebben, d.w.z. $\alpha_7 = \dot{x}_{7d} - c_7 z_7$ ($c_7 > 0$). Dus:

$$\begin{aligned} \dot{V}_7 &= -c_7 z_7^2 + z_7 z_8 \\ z_8 &= x_8 - \dot{x}_{7d} + c_7 z_7 \end{aligned}$$

Zoals we deden in houdingscontrole, moeten we een andere Lyapunov-functie V_8 definiëren zodat $z_8 \rightarrow 0$, om onze regelvariabelen uit te drukken.

Stap 2

Definieer Lyapunov-functie $V_8 = \frac{1}{2}z_8^2 + V_7$, aangezien $\dot{z}_8 = \dot{x}_8 - \dot{x}_{7d} + c_7 \dot{z}_7$, dan:

$$\begin{aligned} \dot{V}_8 &= z_8 \dot{z}_8 + \dot{V}_7 = z_8(\dot{x}_8 - \dot{x}_{7d} + c_7 \dot{z}_7) - c_7 z_7^2 + z_7 z_8 \\ &= z_8[\dot{x}_8 - \dot{x}_{7d} + c_7(z_8 - c_7 z_7)] - c_7 z_7^2 + z_7 z_8 \\ &= -c_7 z_7^2 + z_8[z_7 - \dot{x}_{7d} + c_7(z_8 - c_7 z_7) + \dot{x}_8] \end{aligned} \quad (4)$$

Om te garanderen dat $\dot{V}_8 < 0$, kies:

$$z_7 - \dot{x}_{7d} + c_7(z_8 - c_7 z_7) + \dot{x}_8 = -c_8 z_8 \quad (c_8 > 0) \quad (5)$$

Zet (5) in (4), we hebben:

$$\begin{aligned} \dot{V}_8 &= -c_7 z_7^2 + z_8[z_7 - \dot{x}_{7d} + c_7 z_7 + \dot{x}_8] \\ &= -c_7 z_7^2 - c_8 z_8^2 < 0 \end{aligned}$$

Bovendien, kan (5) worden geschreven als:

$$\dot{x}_8 = \dot{x}_{7d} + (c_7^2 - 1)z_7 - (c_7 c_8)z_8 \quad (c_7 > 0, c_8 > 0) \quad (6)$$

Zet (6) in (3):

$$U_1 = \frac{m}{\cos(x_1)\cos(x_3)} [\dot{x}_{7d} + (c_7^2 - 1)z_7 + g - (c_7 + c_8)z_8]$$

Nu hebben we U_1 zodat z_7 en z_8 asymptotisch stabiel zijn.

Zelfde redenering, we konden de uitdrukking voor virtuele regelvariabelen afleiden:

$$\begin{cases} u_x &= \frac{m}{U_1} [\ddot{x}_{9d} + (c_9^2 - 1)z_9 - (c_9 + c_{10})z_{10}] \\ u_y &= \frac{m}{U_1} [\ddot{x}_{11d} + (c_{11}^2 - 1)z_{11} - (c_{11} + c_{12})z_{12}] \end{cases} \quad (7)$$

waarbij

$$\begin{aligned} z_9 &= x - x_d = x_9 - x_{9d} \\ z_{10} &= x_{10} - \dot{x}_{9d} + c_9 z_9 \\ z_{11} &= y - y_d = x_{11} - x_{11d} \\ z_{12} &= x_{12} - \dot{x}_{11d} + c_{11} z_{11} \end{aligned}$$

Vergelijkbaar met wat we deden in PID-controle, hebben we de outputs van rollhoek en pitchhoek van de positiecontroller nodig en nemen ze als de gewenste inputs van de houdingscontroller. Dus, volgens het systeemmodel, verwerven we de niet-lineaire relatie tussen ϕ_d, θ_d en u_x, u_y :

$$\phi_d = \arcsin(u_x \sin(\psi) - u_y \cos(\psi)) \quad (8)$$

$$= \arcsin(u_x \sin(x_5) - u_y \cos(x_5)) \quad (9)$$

$$\theta_d = \arcsin\left[\frac{u_x - \sin(\phi)\sin(\psi)}{\cos(\phi)\cos(\psi)}\right] \quad (10)$$

$$= \arcsin\left[\frac{u_x - \sin(x_1)\sin(x_5)}{\cos(x_1)\cos(x_5)}\right] \quad (11)$$

Zet (9) en (11) in (7), we konden de uiteindelijke uitdrukking van inputs afleiden:

$$\begin{cases} U_1 &= \frac{m}{\cos(x_1)\cos(x_3)} [\dot{x}_{7d} + (c_7^2 - 1)z_7 + g - (c_7 + c_8)z_8] \\ U_2 &= \frac{1}{b_1} [\dot{x}_{1d} + (c_1^2 - 1)z_1 - a_1 x_4 x_6 - (c_1 + c_2)z_2] \\ U_3 &= \frac{1}{b_2} [\dot{x}_{3d} + (c_3^2 - 1)z_3 - a_2 x_2 x_6 - (c_3 + c_4)z_4] \\ U_4 &= \frac{1}{b_3} [\dot{x}_{5d} + (c_5^2 - 1)z_5 - a_3 x_2 x_4 - (c_5 + c_6)z_6] \end{cases}$$

A.2 Adaptieve Terugkoppelingscontrole in discrete tijd onder verstoring

In de echte omgeving zal een quadrotor altijd worden beïnvloed door onbekende parameters en onverwachte verstoringen aangeduid als D . Om de verstoring D online te schatten en adaptieve feedback te berekenen door de Terugkoppelingsmethode, konden we de dynamische vergelijking van het positiesubstelsysteem als volgt uitdrukken:

$$\begin{cases} m\ddot{x} = & u_x U_1 + D_x \\ m\ddot{y} = & u_y U_1 + D_y \\ m\ddot{z} = & \cos(\phi)\cos(\theta)U_1 - mg + D_z \end{cases}$$

waarbij $u_x = \sin(\theta)\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\psi)$, $u_y = \sin(\theta)\cos(\phi)\sin(\psi) - \sin(\phi)\cos(\psi)$ en $D_i (i = x, y, z)$ wordt aangeduid als verstoring op x, y, z assen.

Neem de hoogtestaat z als voorbeeld, het is de staatruijnte representatie is:

$$\begin{cases} \dot{x}_7 = & x_8 \\ \dot{x}_8 = & \frac{1}{m}(\cos(x_3)\cos(x_1)U_1 - mg + D_z) \end{cases}$$

waarbij $x_7 = z, x_8 = \dot{z}, x_1 = \phi, x_3 = \theta$ en D_z is de verstoring toegevoegd op de z as. Aangezien D_z onbekend is, moeten we het online schatten en de geschatte D_z definiëren als $\hat{D}_z$, de schattingsfout $\tilde{D}_z = D_z - \hat{D}_z$.

Voor een gegeven gewenste hoogte $z_d = x_{7d}$, definieer de hoogtefoutvariabele als $z_7 = x_7 - x_{7d}$, snelheidsfoutvariabele $z_8 = x_8 - \alpha_7$, waarbij α_7 een virtuele regelvariabele is.

Stap 1: ontwerp voor z_7 substelsysteem.

$$\dot{z}_7 = \dot{x}_7 - \dot{x}_{7d} = x_8 - \dot{x}_{7d}$$

Neem x_8 als de regelvariabele, als $x_8 = \dot{x}_{7d} - c_7 z_7$, dan is het z_7 substelsysteem asymptotisch stabiel. Aangezien x_8 een tijdelijke regelvariabele is, kunnen we x_8 alleen uitdrukken door α_7 . Dus, ontwerp:

$$\alpha_7 = \dot{x}_{7d} - c_7 z_7 (c_7 > 0)$$

Kies de Lyapunov-functie van het eerste substelsysteem als $V_7 = \frac{1}{2}z_7^2$, dan:

$$\begin{aligned} \dot{V}_7 &= z_7 \dot{z}_7 = z_7(x_8 - \dot{x}_{7d}) \\ &= z_7(z_8 + \alpha_7 - \dot{x}_{7d}) \\ &= z_7 z_8 - c_7 z_7^2 \end{aligned}$$

Als $\dot{V}_7 < 0$, zal $z_7 z_8$ worden geannuleerd in de volgende stap en zou het eerste substelsysteem stabiel zijn.

Stap 2: bereken voor de daadwerkelijke regelwet en de adaptieve regelwet.

$$\dot{z}_8 = \dot{x}_8 - \ddot{x}_{7d} + c_7 \dot{z}_7$$

Definieer Lyapunov-functie als $V_8 = \frac{1}{2}z_8^2 + V_7$, dan:

$$\begin{aligned} \dot{V}_8 &= z_8 \dot{z}_8 + \dot{V}_7 = z_8(\dot{x}_8 - \ddot{x}_{7d} + c_7 \dot{z}_7) - c_7 \dot{z}_7^2 + z_7 z_8 \\ &= z_8[\dot{x}_8 - \ddot{x}_{7d} + c_7(z_8 - c_7 z_7)] - c_7 \dot{z}_7^2 + z_7 z_8 \\ &= z_8[\frac{1}{m}(\cos(x_3)\cos(x_1)U_1 - mg + D_z) - \ddot{x}_{7d} + c_7(z_8 - c_7 z_7)] - c_7 \dot{z}_7^2 + z_7 z_8 \end{aligned}$$

Nu, de systeeminput:

$$U_1 = \frac{m}{\cos(x_3)\cos(x_1)}(x_{7d} - z_7 - c_8 z_8 - c_7 z_8 + c_7^2 z_7 + g) - \frac{\ddot{D}_z}{m}$$

Aangezien D_z onbekend is, konden we D_z vervangen door $\hat{D}_z$, dan:

$$U_1 = \frac{m}{\cos(x_3)\cos(x_1)}(x_{7d} - z_7 - c_8 z_8 - c_7 z_8 + c_7^2 z_7 + g) - \frac{\hat{D}_z}{m}$$

Om de adaptieve wet van $\hat{D}_z$ te verklaren, stel de Lyapunov-functie als $V_8 = V_7 + \frac{1}{2}z_8^2 + \frac{\tilde{D}_z^2}{2m\lambda}$, ($\lambda > 0$), dan konden we de afgeleide ervan hebben als:

$$\begin{aligned} \dot{V}_8 &= z_8 \dot{z}_8 + \dot{V}_7 - \frac{1}{m\lambda} \tilde{D}_z \dot{\tilde{D}}_z \\ &= -c_7 \dot{z}_7^2 - c_8 z_8^2 + \frac{1}{m} z_8 (D_z - \hat{D}_z) - \frac{1}{m\lambda} \tilde{D}_z \dot{\tilde{D}}_z \\ &= -c_7 \dot{z}_7^2 - c_8 z_8^2 + \frac{\hat{D}_z}{m} (z_8 - \frac{1}{\lambda} \dot{\tilde{D}}_z) \end{aligned}$$

waarbij $\dot{\tilde{D}}_z = \lambda_1 z_8$, ($\lambda > 0$)

Dus, we konden de Lyapunov-functie $\dot{V}_8 = -c_7 \dot{z}_7^2 - c_8 z_8^2 < 0$, ($c_7, c_8 > 0$) houden en garanderen dat $z_7, z_8, \hat{D}_z$ asymptotisch stabiel zijn.

Zelfde redenering, we konden adaptieve terugkoppelingsmethode toepassen op D_x en D_y :

$$\begin{cases} u_x = \frac{m}{U_1} [\ddot{x}_{9d} + (c_9^2 - 1)z_9 - (c_9 + c_{10})z_{10} - \frac{\hat{D}_x}{m}] \\ u_y = \frac{m}{U_1} [\ddot{x}_{11d} + (c_{11}^2 - 1)z_{11} - (c_{11} + c_{12})z_{12} - \frac{\hat{D}_y}{m}] \end{cases} \quad (12)$$

waarbij

$$\begin{cases} \dot{D}_x = \lambda_2 z_{10} & (\lambda_2 > 0) \\ \dot{D}_z = \lambda_3 z_{12} & (\lambda_3 > 0) \end{cases} \quad (13)$$

$$z_9 = x - x_d = x_9 - x_{9d} \quad z_{11} = y - y_d = x_{11} - x_{11d} \quad (14)$$

$$z_{10} = x_{10} - \dot{x}_{9d} + c_9 z_9 \quad z_{12} = x_{12} - \dot{x}_{11d} + c_{11} z_{11}$$

Dan, zet (12),(13),(14)in (9) en (11), we konden de gewenste rollhoek en pitchhoek oplossen adaptief aan onbekende verstoringen.

References

- [1] F. Bu, Y. Cai, and Y. Yang. Multiple object tracking based on faster-rcnn detector and kcf tracker. Technical Report [Online], 2016. Available at: <https://pdfs.semanticscholar.org>.
- [2] Dinesh D. Dhadekar, Prithvi D. Sanghani, K. K. Mangrulkar, and S. E. Talole. Robust Control of Quadrotor using Uncertainty and Disturbance Estimation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 101(3):1–21, mar 2021.
- [3] Shweta Gupte, Paul Infant Teenu Mohandas, and James M. Conrad. A survey of quadrotor unmanned aerial vehicles. In *2012 Proceedings of IEEE Southeastcon*, Orlando, FL, USA, mar 2012. IEEE.
- [4] Raja Jarray, Mujahed Al-Dhaifallah, Hegazy Rezk, and Soufiene Bouallègue. Path Planning of Quadrotors in a Dynamic Environment Using a Multicriteria Multi-Verse Optimizer. *Computers, Materials & Continua*, 69(2):2159–2180, jul 2021.
- [5] Zhenyue Jia, Jianqiao Yu, Yuesong Mei, Yongbo Chen, Yuanchuan Shen, and Xiaolin Ai. Integral backstepping sliding mode control for quadrotor helicopter under external uncertain disturbances. *Aerospace Science and Technology*, 68:299–307, sep 2017.
- [6] Hongyi Li, Lijie Wang, Haiping Du, and Abdesselem Boulkroune. Adaptive fuzzy backstepping tracking control for strict-feedback systems with input delay. *IEEE Transactions on Fuzzy Systems*, 25(3):642–652, jun 2017.
- [7] Omid Mofid, Saleh Mobayen, Chunwei Zhang, and Balasubramanian Esakki. Desired tracking of delayed quadrotor UAV under model uncertainty and wind disturbance using adaptive super-twisting terminal sliding mode control. *ISA Transactions*, 123:455–471, apr 2022.
- [8] Daniel D. Timis, Cristina I. Muresan, and Eva H. Dulf. Design and Experimental Results of an Adaptive Fractional-Order Controller for a Quadrotor. *Fractal and Fractional*, 6(4):204, apr 2022.
- [9] Yuanyuan Xu and Bing Chen. Adaptive neural network control for nonlinear non-strict feedback time-delay systems. *Systems Science & Control Engineering*, 9(S2):81–92, 2021.
- [10] Nguyen Xuan-Mung and Sung Kyung Hong. Robust Backstepping Trajectory Tracking Control of a Quadrotor with Input Saturation via Extended State Observer. *Applied Sciences*, 9(23):5184, nov 2019.
- [11] Y. Yang. *Electromechanical Characterization of Organic Field-Effect Transistors with Generalized Solid-State and Fractional Drift-Diffusion Models*. PhD thesis, Purdue University, 2021.
- [12] Y. Yang and H. H. Zhang. Stability study of lqr and pole-placement genetic algorithm synthesized input-output feedback linearization controllers for a rotary inverted pendulum system. *International Journal of Engineering Innovations and Research*, 7(1):62–68, 2018.
- [13] Y. Yang and H. H. Zhang. Control applications in engineering and technology. In *Fractional Calculus with its Applications in Engineering and Technology*, pages 67–89. Springer International Publishing, Cham, 2019.
- [14] Y. Yang and H. H. Zhang. *Fractional calculus with its applications in engineering and technology*. Morgan & Claypool Publishers, 2019.
- [15] Y. Yang and H. H. Zhang. Fractional-order controller design. In *Fractional Calculus with its Applications in Engineering and Technology*, pages 43–65. Springer International Publishing, Cham, 2019.
- [16] Y. Yang and H. H. Zhang. Preliminary tools of fractional calculus. In *Fractional Calculus with its Applications in Engineering and Technology*, pages

- 3–42. Springer International Publishing, Cham, 2019.
- [17] Y. Yang and H. H. Zhang. Neural network-based adaptive fractional-order backstepping control of uncertain quadrotors with unknown input delays. *Fractal and Fractional*, 7(3):232, 2023.
 - [18] Y. Yang and H. H. Zhang. Optimal model reference adaptive fractional-order proportional integral derivative control of idle speed system under varying disturbances. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, page 09596518241266670, 2024.
 - [19] Y. Yang, H. H. Zhang, and R. M. Voyles. Rotary inverted pendulum system tracking and stability control based on input-output feedback linearization and pso-optimized fractional order pid controller. In *Automatic Control, Mechatronics and Industrial Engineering*, pages 79–84. CRC Press, 2019.
 - [20] Y. Yang, H. H. Zhang, and R. M. Voyles. Optimal fractional-order proportional-integral-derivative control enabling full actuation of decomposed rotary inverted pendulum system. *Transactions of the Institute of Measurement and Control*, 45(10):1986–1998, 2023.
 - [21] Y. Yang, H. H. Zhang, W. Yu, and L. Tan. Optimal design of discrete-time fractional-order pid controller for idle speed control of an ic engine. *International Journal of Powertrains*, 9(1-2):79–97, 2020.
 - [22] Yi Yang. Prediction of hotel price with regression models. Technical report, School of Public Health, University of Michigan, USA, 2023. Presented at a Machine Learning and Biostatistics Online Course, April 2023. Available at: https://yeeyoung.github.io/assets/regression_english.pdf.
 - [23] Yi Yang. Voorspelling van hotelprijzen met regressiemodellen. Technisch rapport, School of Public Health, Universiteit van Michigan, VS, 2023. Gepresenteerd tijdens een online cursus Machine Learning en Biostatistiek, april 2023. Beschikbaar op: https://yeeyoung.github.io/assets/regression_dutch.pdf.
 - [24] Yi Yang. Identificatie van fractionele-orde systeemdynamica voor een roterende omgekeerde slinger. Technisch verslag, Afdeling Werktuigbouwkunde, Universiteit van Michigan, VS, 2024. Gepresenteerd tijdens een online cursus Mechatronica en Robotica, december 2024. Beschikbaar op: https://yeeyoung.github.io/assets/Identification_dutch.pdf.
 - [25] Yi Yang. Identification of fractional-order system dynamics for a rotary inverted pendulum. Technical report, Department of Mechanical Engineering, University of Michigan, USA, 2024. Presented at a Mechatronics and Robotics Online Course, December 2024. Available at: https://yeeyoung.github.io/assets/Identification_english.pdf.
 - [26] Yongliang Zhan, Xiaomei Li, and Shaocheng Tong. Observer-Based Decentralized Control for Non-Strict-Feedback Fractional-Order Nonlinear Large-Scale Systems With Unknown Dead Zones. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
 - [27] Dongliang Zheng, Hesheng Wang, Jingchuan Wang, Siheng Chen, Weidong Chen, and Xinwu Liang. Image-based visual servoing of a quadrotor using virtual camera approach. *IEEE/ASME Transactions on Mechatronics*, 22(2):972–982, apr 2017.