



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH**

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA  
I TECHNOLOGIE KOSMICZNE

## **AUTOREFERAT ROZPRAWY DOKTORSKIEJ**

Szybkie sterowanie niecałkowitego rzędu obiektami  
nieliniowymi

Autor: mgr inż. Jakub Żegleń-Włodarczyk

Promotor rozprawy: prof. dr hab. inż. Krzysztof Oprzędkiewicz  
Promotor pomocniczy: dr inż. Maciej Rosół

Praca wykonana: Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie,  
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Kraków, 2023



# Rozdział 1

## Wstęp

Pojęcie operatora różniczkowania  $D = \frac{d}{dx}$  jest bardzo dobrze znane w matematyce. Dla odpowiednich funkcji  $f(x)$  można nawet wyznaczyć  $n$ -tą pochodną, którą oznacza się jako  $D^n f(x) = \frac{d^n}{dx^n} f(x)$ , przy czym  $n$  musi być dodatnią liczbą całkowitą. W tym miejscu należałoby zapytać o sytuację, gdy  $n$  miałoby wartość ułamkową. W 1695 roku L'Hopital zadał właśnie takie pytanie Leibnizowi [1]. Od tego momentu zaczęto interesować się rachunkiem ułamkowym. Nad tematem pracowało wielu znanych i cenionych matematyków, między innymi Euler, Laplace, Fourier, Liouville, Laurent, Riemann czy Abel. Przez wiele lat rozszerzano teorię w tej dziedzinie [14]. Warto wspomnieć między innymi Lagrange'a, który w 1772 roku przygotował prawo wykładników dla operatorów różnicowych. Z kolei Laplace w 1812 roku opisał pochodną ułamkową za pomocą całki. Jedno z pierwszych zastosowań praktycznych datuje się na 1823 rok, kiedy to Abel zastosował operacje ułamkowe do rozwiązania problemu tautochronu. Dopiero w 1884 roku ta dziedzina osiągnęła wystarczający poziom, aby stać się punktem wyjścia dla matematyków. Oprócz wspomnianych wyżej naukowców w tym samym czasie rozszerzono wiedzę o operatory  $D^m$ , gdzie  $m$  może już być dowolną liczbą dodatnią, ujemną, rzeczywistą, złożoną, wymierną lub niewymierną. W związku z powyższym należy zauważyć, że termin *rachunek ułamkowy* może być trochę mylący. Niemniej postanowiono go zostawić, aby uszanować historię. Dzisiaj stosuje się go naprzemiennie ze sformulowaniem *rachunek niecałkowitego rzędu*.

Idea całek i pochodnych ułamkowych jest trudna do prostego wytłumaczenia. Nie da się jej przedstawić w łatwy sposób, w przeciwieństwie do klasycznych operatorów, gdzie na przykład przyspieszenie to pierwsza pochodna prędkości po czasie. Mimo wszystko wiele systemów i zjawisk fizycznych można opisać w o wiele dokładniejszy sposób, korzystając z rachunku ułamkowego. W związku z tym możliwe jest lepsze zamodelowanie świata, co implikuje dokładniejsze symulacje przeprowadzane na komputerach.

### 1.1 Cele i tezy pracy

Celem dysertacji jest zbadanie właściwości regulatorów ułamkowych podczas pracy w systemach sterowania, gdzie istotne są dwa czynniki: jakość sterowania w stosunku do wybranych funkcji kosztu oraz czas wykonania obliczeń. To niezmiernie istotne dla urządzeń, których dynamika jest stosunkowo szybka. Można do nich zaliczyć suwnicę 3D, dwurotorowy system aerodynamiczny oraz wahadło odwrócone. Sterowa-

nie dla tego typu obiektów musi być wyznaczone w twardym czasie rzeczywistym. Zagadnienia związane z szybkim sterowaniem cyfrowym niecałkowitego rzędu nie były do tej pory zbyt szeroko przedstawiane w literaturze. Z tego względu tezy pracy zostały sformułowane w następujący sposób:

**Teza I:** Regulator PID niecałkowitego rzędu (FOPID) może być zastosowany do sterowania szybkimi nieliniowymi systemami dynamicznymi.

**Teza II:** Zastosowanie regulatora FOPID w większości sytuacji zapewnia lepszą jakość regulacji, w sensie wybranych funkcji kosztu, niż w przypadku klasycznych metod sterowania (regulator PID).

**Teza III:** Algorytm regulatora FOPID może być zaimplementowany na platformie cyfrowej ze spełnieniem wymagań czasu rzeczywistego.

W celu udowodnienia powyższych tez należało rozwiązać następujące zadania naukowe:

**Zadanie I:** Implementacja algorytmu regulacyjnego FOPID na platformie Matlab/Simulink z wykorzystaniem aproksymacji Oustaloup Recursive Approximation (ORA).

**Zadanie II:** Opracowanie i zastosowanie metod dostrajania regulatora FOPID do konkretnego obiektu.

**Zadanie III:** Wykonanie testów symulacyjnych działania algorytmu FOPID w środowisku Matlab/Simulink na wybranych nieliniowych modelach obiektów.

**Zadanie IV:** Weryfikacja doświadczalna zaimplementowanego algorytmu na stanowiskach laboratoryjnych. Testy muszą objąć zarówno jakość regulacji w sensie wybranych funkcji kosztu (specyficznych dla konkretnego obiektu), jak również spełnienie wymagań czasu rzeczywistego, co związane jest z miarą szybkości wykonywania obliczeń podczas pracy algorytmu.

## Rozdział 2

# Elementy rachunku niecałkowitego rzędu

### 2.1 Elementarny operator niecałkowitego rzędu

Systemy ułamkowe (inaczej systemy niecałkowitego rzędu) można uznać za uogólnienie systemów całkowego rzędu [4, 8, 12].

Rachunek ułamkowy jest generalizacją rachunku różniczkowego i całkowego na podstawowy operator  ${}_a D_t^r$  niecałkowitego rzędu, gdzie  $a$  oraz  $t$  są granicami operacji, natomiast  $r \in \mathfrak{R}$ . Ciągły operator całkowo-różniczkowy definiowany jest jako [1]:

$${}_a D_t^r = \begin{cases} \frac{d^r}{dt^r} & \text{jeśli } r > 0, \\ 1 & \text{jeśli } r = 0, \\ \int_a^t (d\tau)^{-r} & \text{jeśli } r < 0. \end{cases} \quad (2.1)$$

Następnie należy sformułować funkcję gamma, która jest również nazywana gammą Eulera [5]. Jest ona zdefiniowana dla dodatnich liczb rzeczywistych. Wzór funkcji gamma wygląda następująco:

$$\Gamma(m) = \int_0^{\infty} e^{-u} u^{m-1} dt. \quad (2.2)$$

Ponadto, jeśli  $m$  jest liczbą całkowitą, to gammę Eulera można zapisać jako:

$$\Gamma(m + 1) = m! \quad (2.3)$$

Oznacza to, że funkcję gamma należy traktować jako uogólnienie silni na liczby rzeczywiste.

Operator niecałkowitego rzędu (2.1) można opisać różnymi definicjami. W podrozdziałach 2.2 oraz 2.3 przedstawiono najbardziej popularne. Zostały one wykorzystane w dalszych rozważaniach. Należy jednak nadmienić, iż istnieje wiele innych, między innymi można do nich zaliczyć definicje Grünwalda-Letnikova (GL) [15], Atangana-Baleanu (AB) [18] oraz Hadamarda (H) [7].

## 2.2 Definicja Riemanna-Liouville'a (RL)

Definicja Riemanna-Liouville'a jest uogólnieniem klasycznego wzoru na całkę  $n$ -tego rzędu [11]. Dla każdego  $x$  należącego do przedziału  $[a, b]$  oraz  $n \in \mathbf{N}$  wzór na całkę Riemanna dla funkcji  $f(t)$  w przedziale  $[a, b]$  przedstawia się następująco:

$${}_a I_t^n f(t) = \frac{1}{(n-1)!} \int_a^t (t-\tau)^{n-1} f(\tau) d\tau. \quad (2.4)$$

Ponadto, jeśli funkcja  $f(t)$  jest różniczkowalna  $n$  razy w przedziale  $[a, b]$ , podczas gdy  $m, n \in \mathbf{N}$  oraz  $m > n$ , to można wyprowadzić następujący wzór:

$$\frac{d^n}{dt^n} f(t) = \frac{d^m}{dt^m} I_t^{m-n} f(t). \quad (2.5)$$

Korzystając z gammy Eulera (2.2), formułę (2.4) można uogólnić na niecałkowite rzędy, uzyskując następujące równanie:

$${}_a^{RL} I_t^r f(t) = \frac{1}{\Gamma(r)} \int_a^t (x-\tau)^{r-1} f(\tau) d\tau, \quad (2.6)$$

co stanowi całkę Riemanna-Liouville'a rzędu  $r$  w przedziale  $[a, b]$ , gdzie  $r \in \mathfrak{R}$ .

Pochodną ułamkową w sensie Riemanna-Liouville'a rzędu  $r$  w przedziale  $[a, b]$ , gdzie  $r \in \mathfrak{R}$  oraz  $a \leq t \leq b$ , wykorzystując wzór (2.6) i uogólniając (2.5), definiuje się następująco:

$${}_a^{RL} D_t^r f(t) = \frac{1}{\Gamma(n-r)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{r-n+1}} d\tau, \quad (2.7)$$

gdzie  $n-1 < r < n$ .

## 2.3 Definicja Caputo (C)

Reguła została wyprowadzona przez Michele'a Caputo w 1967 roku [2]. Ta definicja również wykorzystuje całkę w sensie Riemanna-Liouville'a.

Jeśli funkcję  $f(t)$  można różniczkować  $n$  razy w przedziale  $[a, b]$ , podczas gdy  $m, n \in \mathbf{N}$  oraz  $m > n$ , to ogólny wzór na tę operację można zapisać jako:

$$\frac{d^n}{dt^n} f(t) = I_t^{m-n} \frac{d^m}{dt^m} f(t). \quad (2.8)$$

Ostatecznie pochodną ułamkową w sensie Caputo rzędu  $r$  w przedziale  $[a, b]$ , gdzie  $r \in \mathfrak{R}$  oraz  $a \leq t \leq b$ , przedstawia poniższa formuła:

$${}_a^C D_t^r f(t) = \frac{1}{\Gamma(n-r)} \int_a^t \frac{\frac{d^n}{dt^n} f(\tau)}{(t-\tau)^{r-n+1}} d\tau, \quad (2.9)$$

gdzie  $n - 1 < r < n$ .

Warunki początkowe dla równań różniczkowych ułamkowego rzędu z pochodnymi Caputo są w takiej samej postaci jak dla ich odpowiedników całkowego rzędu.

## 2.4 Transformata Laplace'a niecałkowitego rzędu

Standardową definicję transformaty Laplace'a przedstawia się wzorem:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt. \quad (2.10)$$

Z kolei odwrotna transformata jest wyrażana poprzez:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} e^{st} F(s) ds, \quad (2.11)$$

gdzie  $c$  to stała większa od części rzeczywistych każdego z punktów funkcji na płaszczyźnie  $s$ , gdzie funkcja  $F(s)$  nie istnieje.

Aby wyliczyć transformatę dla pochodnej  $n$ -tego rzędu, należy skorzystać ze wzoru:

$$\mathcal{L}\{f^{(n)}(t)\} = s^n F(s) - \sum_{k=0}^{n-1} s^{n-k+1} f^{(k)}(0) = s^n F(s) - \sum_{k=0}^{n-1} s^k f^{(n-k-1)}(0). \quad (2.12)$$

Jeśli dodatkowo pod uwagę zostanie wzięta transformata splotu funkcji:

$$\mathcal{L}\{f(t) * g(t)\} = F(s)G(s), \quad (2.13)$$

gdzie:

$$f(t) * g(t) = \int_0^t f(t-\tau)g(\tau)d\tau = \int_0^t f(\tau)g(t-\tau)d\tau, \quad (2.14)$$

to transformatę Laplace'a (2.10) uogólnia się dla pochodnych rzędu ułamkowego w sensie Riemanna-Liouville'a [11]:

$$\mathcal{L}\{ {}_0^{RL}D_t^r f(t) \} = s^r F(s) - \sum_{k=0}^{n-1} s^k [ {}_0^{RL}D_t^{r-k-1} f(t) ]_{t=0}, \quad (2.15)$$

gdzie  $n - 1 < r < n$ .

W podobny sposób wyprowadza się wzór transformaty Laplace'a dla pochodnej rzędu ułamkowego w sensie Caputo:

$$\mathcal{L}\{ {}_0^C D_t^r f(t) \} = s^r F(s) - \sum_{k=0}^{n-1} s^{r-k-1} f^{(k)}(0), \quad (2.16)$$

gdzie  $n - 1 < r < n$ .

Formuła (2.16) ma przewagę nad (2.15), gdyż łatwo w niej zinterpretować w sposób fizyczny wartości graniczne dla dolnego ograniczenia, gdzie  $t = 0$ . Jeśli jednak przyjęte zostaną zerowe warunki początkowe, to obie wersje upraszczają się do następującego wzoru:

$$\mathcal{L}\{ {}_0D_t^r f(t) \} = s^r F(s). \quad (2.17)$$

## 2.5 Aproksymacja ORA - Oustaloup Recursive Approximation

Do numerycznego obliczania pochodnej ułamkowego rzędu można wykorzystać bardzo popularną metodę, jaką jest aproksymacja ORA - Oustaloup Recursive Approximation. To rekurencyjny filtr, który bardzo dobrze przybliża pochodną ułamkowego rzędu [13, 19]. Metoda opiera się na aproksymacji funkcji postaci:

$$H(s) = s^r \quad (2.18)$$

dla wybranego zakresu częstotliwości  $\omega \in [\omega_b, \omega_h]$  przez funkcję wymierną:

$$\widehat{H}(s) = C_0 \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k}, \quad (2.19)$$

gdzie  $r \in \mathfrak{R}$ ,  $r$  znajduje się w przedziale  $[-1, 1]$  oraz  $N$  jest rzędem aproksymacji. Bieguny, zera i wzmocnienie filtra można określić za pomocą wzorów:

$$\begin{aligned} \omega'_k &= \omega_b \left( \frac{\omega_h}{\omega_b} \right)^{\frac{k + N + 0.5(1 - r)}{2N + 1}}, \\ \omega_k &= \omega_b \left( \frac{\omega_h}{\omega_b} \right)^{\frac{k + N + 0.5(1 + r)}{2N + 1}}, \\ C_0 &= \left( \frac{\sqrt{\omega_h \omega_b}}{\omega_h} \right)^r \prod_{k=-N}^N \frac{\omega_k}{\omega'_k}. \end{aligned} \quad (2.20)$$

Procedurę Matlab `ora_for()` tej metody można pobrać ze strony internetowej MathWorks, Inc. [3].

## 2.6 Regulator FOPID

PID, czyli regulator proporcjonalno-całkująco-różniczkujący, jest jednym z najbardziej rozpowszechnionych algorytmów regulacyjnych w praktyce. Wynika to z faktu, że od dłuższego czasu jest obiektem badań. W 1922 roku Nicolas Minorsky napisał jedną z pierwszych prac na ten temat [9]. Od tego czasu PID był poddawany różnym testom i okazał się jednym z najbardziej odpornych na zakłócenia, będąc jednocześnie łatwym w użyciu.

Transmitancja operatorowa tego regulatora w wersji idealnej wyrażana jest wzorem:



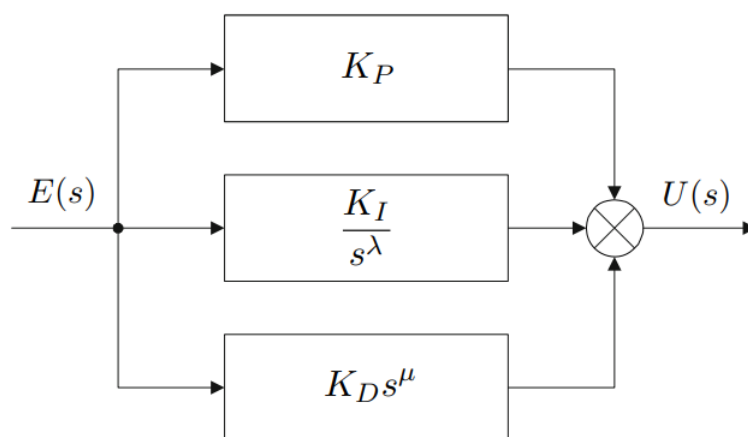
$$G_{PID} = K_P + \frac{K_I}{s} + K_D s, \quad (2.21)$$

gdzie  $K_P$  to wzmocnienie części proporcjonalnej,  $K_I$  to wzmocnienie części całkującej oraz  $K_D$  to wzmocnienie części różniczkującej.

Uogólnienie regulatora PID do rachunku ułamkowego to FOPID (Fractional Order PID). Posiada on dwa dodatkowe parametry, którymi są niecałkowite rzędy całkowania i różniczkowania. W związku z tym transmitancja regulatora FOPID przyjmuje następującą postać [16, 17]:

$$G_{FOPID} = K_P + \frac{K_I}{s^\lambda} + K_D s^\mu, \quad (2.22)$$

gdzie  $K_P$ ,  $K_I$ ,  $K_D$  pozostają bez zmian względem PID, natomiast  $\lambda$  to rząd całki oraz  $\mu$  to rząd różniczkowania. Struktura FOPID została przedstawiona na rysunku 2.1.



Rysunek 2.1: Struktura regulatora FOPID

FOPID jest bardziej elastyczny niż standardowy PID. Daje on także możliwość lepszego dopasowania do konkretnego sterowanego procesu. Dodatkowo, przy odpowiednich parametrach, może przyjmować postać szczególnych przypadków regulatora PID. Przykładowo podstawiając  $\lambda = 1$  oraz  $\mu = 1$ , można otrzymać klasyczny regulator PID. Jednak należy zaznaczyć, że przy FOPID mogą pojawić się problemy z likwidacją uchybu ustalonego. Wynika to z nieobecności czystego całkowania w regulatorze. Aproksymacje (CFE [1], PSE [6], ORA) zapewniają fizyczną realizowalność różniczkowania. Jest to jednak niemożliwe w przypadku przeprowadzania tej operacji dla pierwszego rzędu.

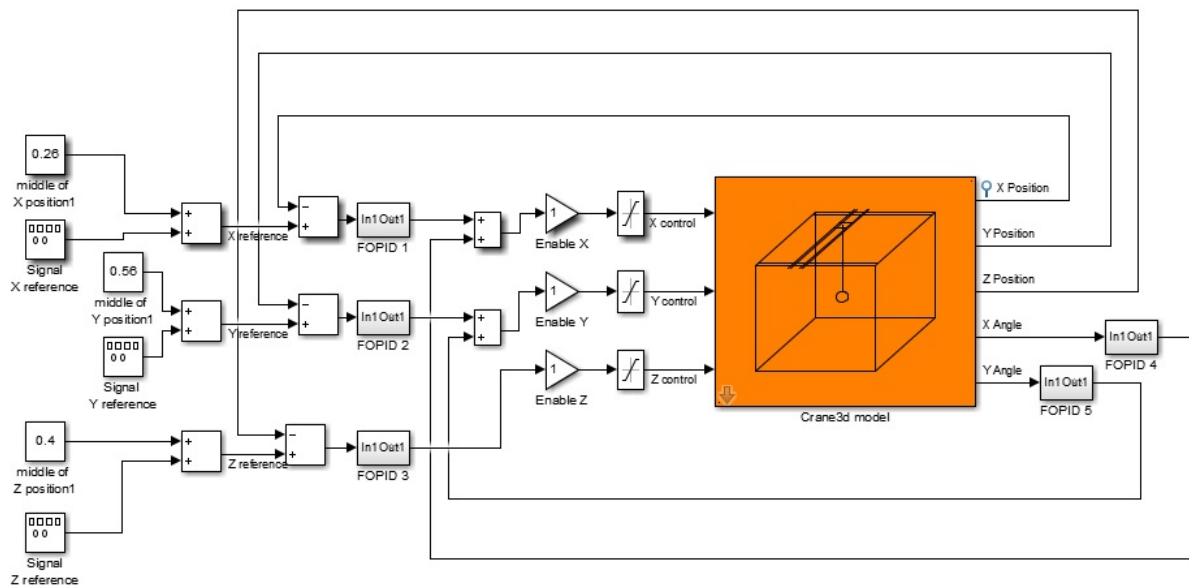


## Rozdział 3

# Wyniki badań

### 3.1 Suwnica 3D

Pierwszym obiektem, na którym przetestowano działanie regulatorów FOPID, była suwnica 3D. W tym celu przygotowano odpowiednią implementację w środowisku Matlab/Simulink. Schemat blokowy programu sterującego jest pokazany na rysunku 3.1.



Rysunek 3.1: Model symulacyjny układu sterowania suwnicą 3D

Do oceny jakości regulacji zaproponowano następującą funkcję kosztu:

$$J = \int_0^{\infty} (|e_x(t)| + |e_y(t)| + |e_z(t)| + |\alpha(t)| + |\beta(t)|) dt, \quad (3.1)$$

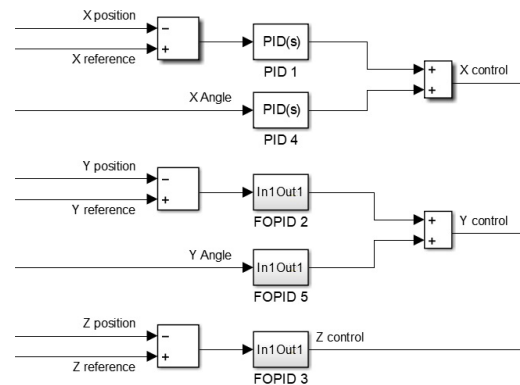
gdzie:

$e_x(t)$  - różnica między wartością zadaną i aktualną wartością na osi  $X$ ,  
 $e_y(t)$  - różnica między wartością zadaną i aktualną wartością na osi  $Y$ ,  
 $e_z(t)$  - różnica między wartością zadaną i aktualną wartością na osi  $Z$ ,  
 $\alpha(t)$  - kąt pomiędzy osią  $Z$  oraz osią  $X$ ,  
 $\beta(t)$  - kąt pomiędzy osią  $Z$  oraz osią  $Y$ .

Przygotowano między innymi następujące rodzaje układów sterowania:

**1. wariant** - dwa regulatory PID oraz trzy regulatory FOPID - rysunek 3.2.

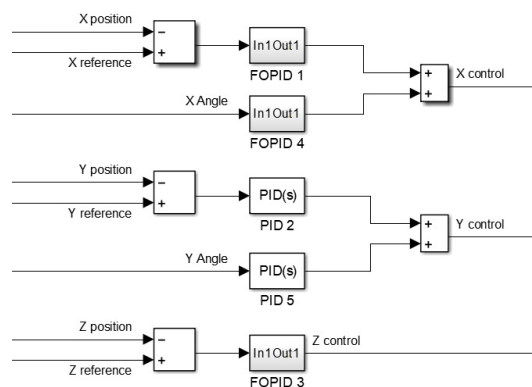
Kontrolery PID znajdują się tylko w pętli sprzężenia zwrotnego wyliczającej sterowanie dla osi  $X$ . W związku z tym są to regulatory korzystające z informacji o uchybie na osi  $X$  oraz wartości kąta  $\alpha$ .



Rysunek 3.2: Sterowanie suwnicą 3D - pierwszy wariant

**2. wariant** - dwa regulatory PID oraz trzy regulatory FOPID - rysunek 3.3.

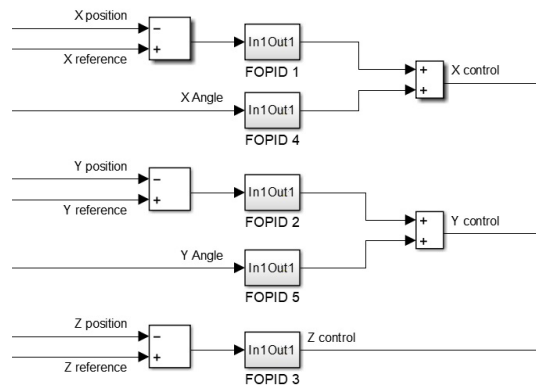
Kontrolery PID znajdują się tylko w pętli sprzężenia zwrotnego wyliczającej sterowanie dla osi  $Y$ . W związku z tym są to regulatory korzystające z informacji o uchybie na osi  $Y$  oraz wartości kąta  $\beta$ .



Rysunek 3.3: Sterowanie suwnicą 3D - drugi wariant

**3. wariant** - pięć regulatorów FOPID - rysunek 3.4.

Sterowanie dla wszystkich osi jest wyliczane za pomocą kontrolerów FOPID.

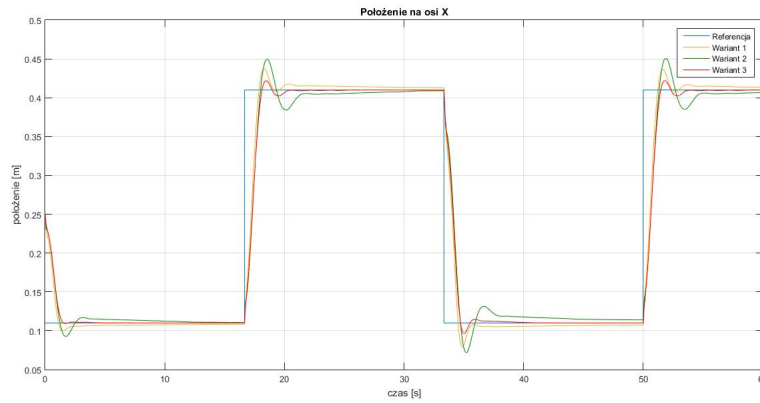
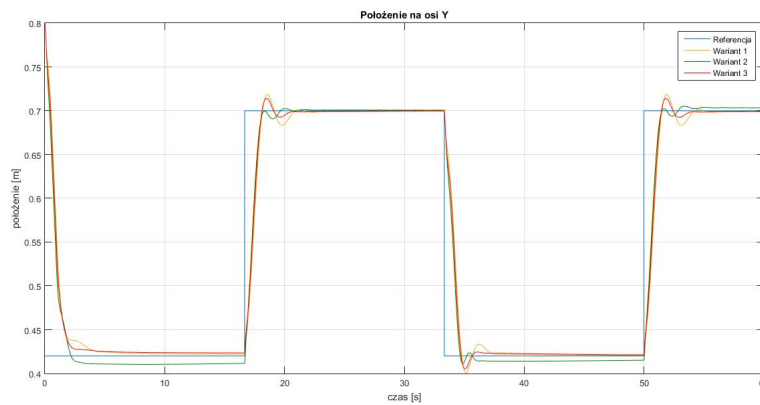
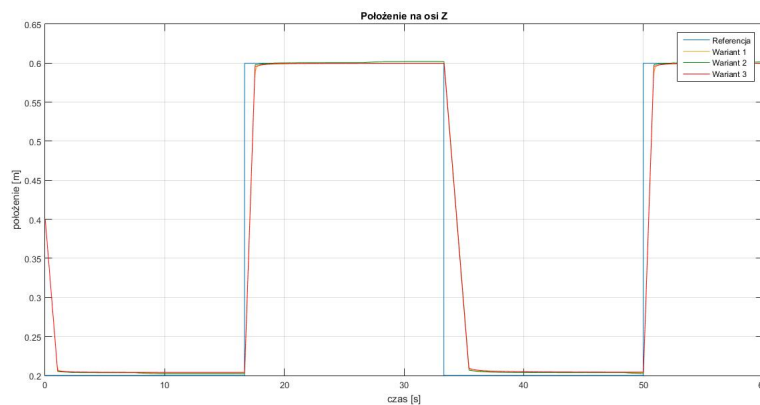


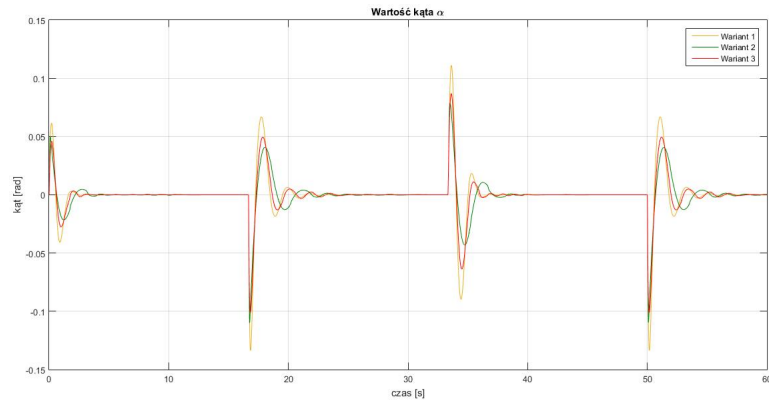
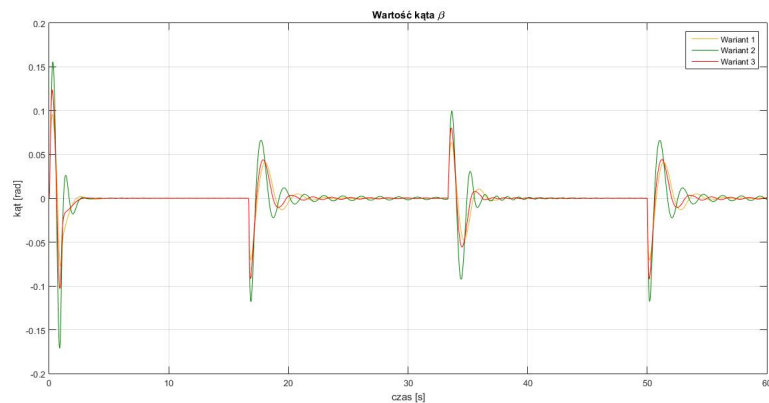
Rysunek 3.4: Sterowanie suwnicą 3D - trzeci wariant

Algorytm Grey Wolf Optimizer [10] korzystający z funkcji kosztu (3.1) został wykorzystany w celu znalezienia współczynników dla poszczególnych kontrolerów. Tabela 3.1 przedstawia wyznaczone wartości. W celu odpowiedniego zaimplementowania kontrolerów FOPID wykorzystano aproksymację rekurencyjną Oustaloupa. Zastosowany rząd aproksymacji to 8, zakres częstotliwości to przedział  $\omega \in [10^{-3}, 10]$ . Krok czasowy wyniósł 0.0005.

Tabela 3.1: Współczynniki regulatorów suwnicy 3D

1. wariant	P	I	$\lambda$	D	$\mu$
PID 1	131.52	6.58	—	64.49	—
FOPID 2	49.96	5.71	0.43	36.50	0.99
FOPID 3	154.38	1.55	0.051	8.98	0.83
PID 4	146.92	17.58	—	1.12	—
FOPID 5	41.27	35.11	0.50	43.48	0.045
2. wariant	P	I	$\lambda$	D	$\mu$
FOPID 1	17.55	2.15	0.19	24.83	0.73
PID 2	36.00	1.49	—	18.74	—
FOPID 3	131.61	5.77	0.12	2.04	0.21
FOPID 4	28.28	40.33	0.12	33.64	0.18
PID 5	34.27	5.42	—	1.39	—
3. wariant	P	I	$\lambda$	D	$\mu$
FOPID 1	48.74	17.96	0.99	12.41	0.99
FOPID 2	7.74	25.64	0.030	18.04	0.99
FOPID 3	116.40	5.75	0.074	32.66	0.058
FOPID 4	29.62	21.28	0.17	20.90	0.14
FOPID 5	28.95	17.23	0.041	8.18	0.31

Rysunek 3.5: Odpowiedź systemu - położenie na osi  $X$ Rysunek 3.6: Odpowiedź systemu - położenie na osi  $Y$ Rysunek 3.7: Odpowiedź systemu - położenie na osi  $Z$

Rysunek 3.8: Odpowiedź systemu - wartość kąta  $\alpha$ Rysunek 3.9: Odpowiedź systemu - wartość kąta  $\beta$ 

Rysunki 3.5 - 3.9 przedstawiają odpowiedź systemu podczas przeprowadzanych symulacji. W tabeli 3.2 zebrano wartości funkcji kosztu (3.1) dla poszczególnych wariantów. Wersja trzecia z pięcioma regulatorami FOPID osiągnęła najlepszy wynik. W wariacie drugim zanotowano największą wartość. Jest to związane przede wszystkim z niemożnością pełnego wytłumienia oscylacji kąta  $\beta$  oraz pojawieniem się uchybu ustalonego na osi  $Y$ . Warto wspomnieć, że na te wartości wpływało w głównej mierze sterowanie osi  $Y$  wyliczane tylko przez regulatory PID. Na tej podstawie można wyciągnąć wniosek, że słabiej poradziły sobie one z postawionym zadaniem. Różnica pomiędzy wariantami pierwszym oraz trzecim jest mniejsza i wynika głównie z pomiarów osi  $X$ . Opcja pierwsza potrzebowała zdecydowanie więcej czasu, aby pokonać ostatnie kilka centymetrów do wartości zadanej. Tutaj również warto zauważyć, że w wersji pierwszej sterowanie osi  $X$  też składało się tylko z regulatorów PID (dotyczy uchybu na osi  $X$  oraz kąta  $\alpha$ ).

Tabela 3.2: Wartości funkcji kosztu (3.1) dla suwnicy 3D

	Funkcja kosztu
1. wariant	7497.05
2. wariant	8066.55
3. wariant	7051.12

Ponadto przeprowadzono pomiary czasów wykonywania obliczeń przez regulatory dla trzeciego wariantu (wszystkie pięć regulatorów FOPID). W tabeli 3.3 zebrano wartości uśrednione i maksymalne. Regulatory FOPID 4 oraz FOPID 5 były zwykle odrobinę wolniejsze od pozostałych. Prawdopodobnie ma to związek z tym, że obliczenia dla tych kontrolerów mogą zacząć się nieco wcześniej, gdyż wykorzystują one nieprzetworzone dane wyjściowe z obiektu - kąty  $\alpha$  oraz  $\beta$ . W przypadku pozostałych regulatorów przeprowadzane są jeszcze obliczenia związane z odchyleniem od wartości zadanej. Dopiero wtedy sterowniki FOPID 1, FOPID 2 i FOPID 3 zaczynają wykonywać swoje obliczenia. W trakcie tych operacji FOPID 4 oraz FOPID 5 kończą swoje działania, przez co zwalniają pamięć obliczeniową dla pierwszych trzech regulatorów. Ostatecznie tabela 3.3 potwierdza, że został spory margines pomiędzy ekstremalnymi przypadkami i krokiem czasowym obliczeń wynoszącym 0.0005.

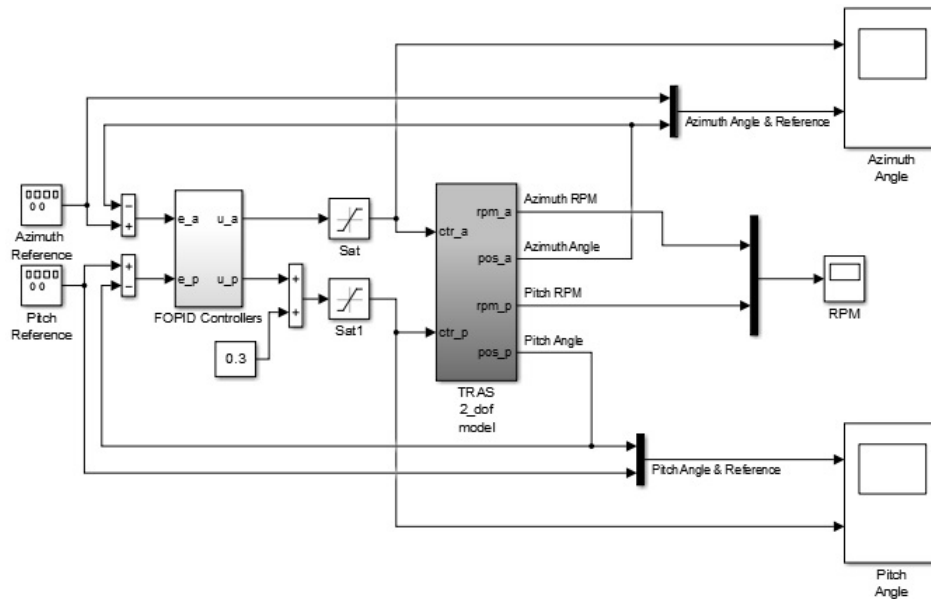
Tabela 3.3: Średni i maksymalny czas obliczeń regulatorów - trzeci wariant

Regulator	Średni czas obliczeń [s]	Maksymalny czas obliczeń [s]
FOPID 1	2.5029e-07	5.5610e-06
FOPID 2	2.8422e-07	4.7885e-06
FOPID 3	3.1299e-07	3.4473e-06
FOPID 4	3.5585e-07	3.0823e-06
FOPID 5	4.0239e-07	2.9723e-06

## 3.2 Dwurotorowy system aerodynamiczny

W celu przetestowania sterowania FOPID na przykładzie dwurotorowego systemu aerodynamicznego przygotowano układ sterowania w środowisku Matlab/Simulink przedstawiony na rysunku 3.10.





Rysunek 3.10: Model symulacyjny układu sterowania dwurotorowym systemem aerodynamicznym

Współczynniki wszystkich kontrolerów zostały wyznaczone poprzez zastosowanie algorytmu optymalizacyjnego Grey Wolf Optimizer [10]. W tym celu zdefiniowano następującą funkcję kosztu:

$$J = \int_0^{\infty} (|e_a(t)| + |e_p(t)|) dt, \quad (3.2)$$

gdzie:

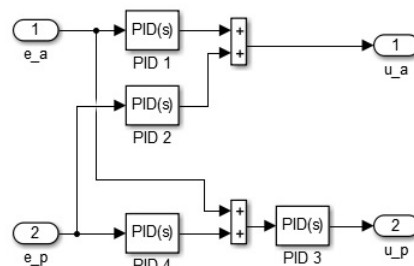
$e_a(t)$  - różnica między wartością zadaną i aktualną wartością kąta azymutowego,

$e_p(t)$  - różnica między wartością zadaną i aktualną wartością kąta nachylenia.

Przygotowano między innymi sterowanie hybrydowe w trzech wersjach. Różnią się one ilością regulatorów FOPID oraz PID. Wszystkie warianty przedstawiono poniżej:

**1. wariant** - cztery regulatory PID - rysunek 3.11.

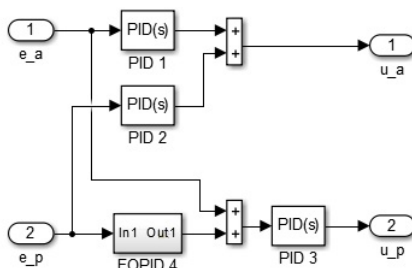
Wszystkie kontrolery występujące w tej wersji to PID.



Rysunek 3.11: Sterowanie hybrydowe - pierwszy wariant

**2. wariant** - trzy regulatory PID i jeden regulator FOPID - rysunek 3.12.

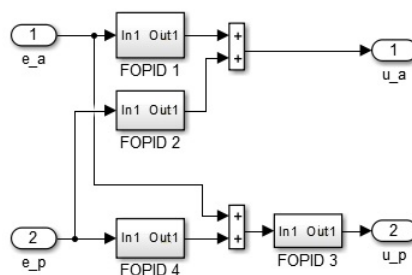
W pętli sterowania kaskadowego dla kąta nachylenia zastosowano regulator FOPID w położeniu nadrzędnym. Dostaje on informacje na temat różnicy pomiędzy aktualnym przechyleniem i wartością zadaną. Pozostałe kontrolery to PID.



Rysunek 3.12: Sterowanie hybrydowe - drugi wariant

**3. wariant** - cztery regulatory FOPID - rysunek 3.13.

Wszystkie kontrolery występujące w tej wersji to FOPID.

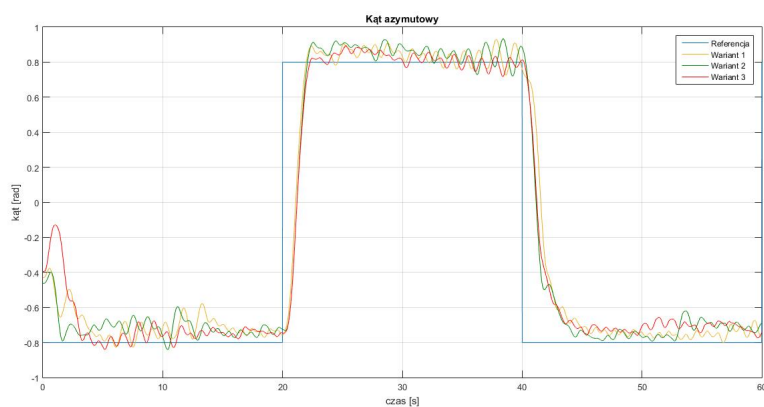


Rysunek 3.13: Sterowanie hybrydowe - trzeci wariant

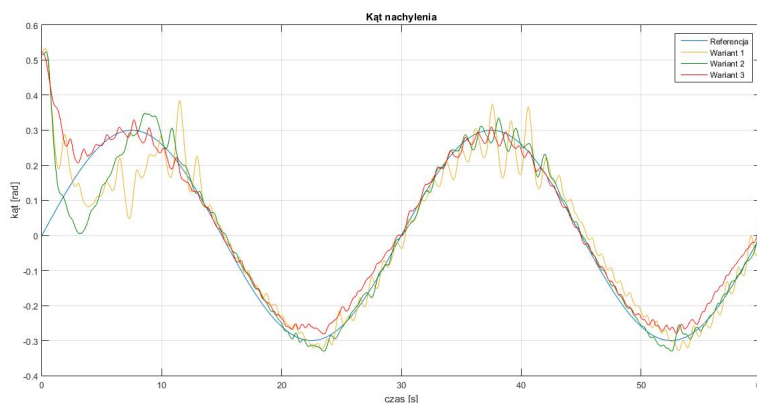
Do wyznaczenia nastaw wszystkich regulatorów zastosowano algorytm GWO [10], który korzystał z funkcji kosztu (3.2). Tabela 3.4 w zbiorczy sposób przedstawia wybrane wartości współczynników. Do implementacji regulatorów FOPID użyto aproksymacji rekurencyjnej Oustaloupa z rzędem aproksymacji równym 8 oraz zakresem częstotliwości  $\omega \in [10^{-3}, 10]$ . Eksperymenty były przeprowadzane ze stałym krokiem 0.01.

Tabela 3.4: Współczynniki regulatorów TRAS

1. wariant	P	I	$\lambda$	D	$\mu$
PID 1	163.25	0.98	—	111.72	—
PID 2	57.60	0.34	—	28.68	—
PID 3	19.10	1.45	—	22.04	—
PID 4	245.61	78.71	—	4.77	—
2. wariant	P	I	$\lambda$	D	$\mu$
PID 1	52.38	0.41	—	37.90	—
PID 2	6.85	10.19	—	37.86	—
PID 3	69.61	59.23	—	25.54	—
FOPID 4	17.89	9.45	0.57	63.77	0.012
3. wariant	P	I	$\lambda$	D	$\mu$
FOPID 1	49.30	4.34	0.99	6.27	0.023
FOPID 2	18.45	2.54	0.063	35.65	0.52
FOPID 3	19.24	0.032	0.052	1.40	0.95
FOPID 4	17.60	40.47	0.026	4.27	0.54



Rysunek 3.14: Odpowiedź systemu - kąt azymutowy



Rysunek 3.15: Odpowiedź systemu - kąt nachylenia

Rysunki 3.14 - 3.15 ukazują odpowiedź systemu podczas eksperymentów. Tabela 3.5 przedstawia wartości funkcji kosztu (3.2). Krok czasowy wynosił 0.01. Najlepszy wynik osiągnął wariant drugi z jednym regulatorem FOPID. Niewiele gorsza była wersja trzecia z czterema kontrolerami ułamkowymi. Najgorszy wynik zanotowano dla sterowania w wersji pierwszej. Najprawdopodobniej przyczyniły się do tego zwiększone oscylacje w przypadku pomiaru kąta nachylenia - zwłaszcza w okolicach maksimum. Należy zauważyć, że wszystkie warianty były stosunkowo bardzo odporne na nagłe zmiany wartości zadanej kąta azymutowego - przy sterowaniu równoległym miał on duży wpływ na wartość kąta nachylenia. W tej pętli sprzężenia zwrotnego przy sterowaniu hybrydowym zastosowano kaskadowy układ regulatorów. Pozwala to wysnuć wniosek, że, w przypadku systemu TRAS, to ułożenie jest bardziej odporne na dodatkowe zakłócenia.

Tabela 3.5: Wartości funkcji kosztu (3.2) dla systemu TRAS

	Funkcja kosztu
1. wariant	1150.44
2. wariant	1030.06
3. wariant	1043.12

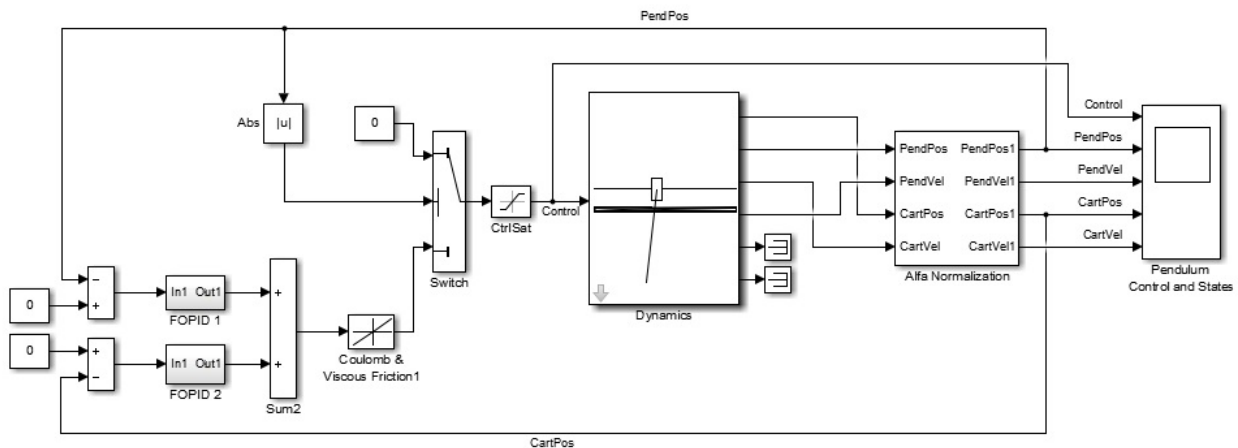
Pomiary czasów obliczeń dla wariantu trzeciego zostały przedstawione w tabeli 3.6. Regulator o numerze 3 miał najkrótsze czasy wykonania. Ma to związek z tym, że pozostałe kontrolery działały równolegle w tym samym czasie, wobec czego moc obliczeniowa procesora była podzielona. Z kolei regulator o numerze 3 musiał czekać na wyniki przygotowane przez kontroler o oznaczeniu 4. W związku z tym obliczenia dla FOPID 3 wykonywane są na samym końcu, gdy już procesor nie potrzebuje dzielić swoich zasobów. Ponadto maksymalne czasy obliczeń wszystkich regulatorów były na poziomie  $10^{-6}$ , podczas gdy krok czasowy wynosił 0.01. Można zatem stwierdzić, iż wciąż pozostał bardzo duży margines na wykonanie obliczeń potrzebnych do wyznaczenia sterowania.

Tabela 3.6: Średni i maksymalny czas obliczeń regulatorów - trzeci wariant

Regulator	Średni czas obliczeń [s]	Maksymalny czas obliczeń [s]
FOPID 1	6.3335e-07	4.9437e-06
FOPID 2	6.9608e-07	4.2425e-06
FOPID 3	2.0990e-07	1.3675e-06
FOPID 4	7.0620e-07	4.2500e-06

### 3.3 Wahadło odwrócone

Regulatory FOPID przetestowano również na przykładzie układu wahadła odwróconego. W tym celu przygotowano odpowiedni system sterowania w środowisku Matlab/Simulink przedstawiony na rysunku 3.16.



Rysunek 3.16: Model symulacyjny układu sterowania wahadłem odwróconym

W celu optymalizacji współczynników kontrolerów zastosowano algorytm GWO [10]. Cały proces podzielono na dwie części:

1. Optymalizacja samego regulatora odpowiedzialnego za kąt wahadła z wyłączonym kontrolerem powiązanim z położeniem wózka.

Na potrzeby problemu, jakim jest optymalizacja współczynników regulatora odpowiedzialnego za kąt wahadła, zastosowano następującą funkcję kosztu:

$$J_1 = \begin{cases} \int_0^{\infty} 100dt & \text{jeśli } |\alpha| \geq 0.31, \\ 0 & \\ \int_0^{\infty} (|e_{\alpha}(t)| + t|v_c(t)|) dt & \text{jeśli } |\alpha| < 0.31, \end{cases} \quad (3.3)$$

gdzie:

$\alpha$  - wartość kąta wahadła,

$e_\alpha$  - różnica między wartością zadaną i aktualną wartością kąta wahadła,

$v_c$  - prędkość wózka,

$t$  - czas.

Tego typu funkcja kosztu ma na celu szybko ustawić wahadło do zadanego położenia. Ponadto z każdą chwilą coraz ważniejsze staje się to, aby zminimalizować poruszanie wózka. Stała 100 ma za zadanie informować algorytm, że niedopuszczalna jest sytuacja, gdy wahadło wypada poza obszar wokół punktu zadanego.

2. Optymalizacja drugiego regulatora związanego z położeniem wózka, kontroler odpowiedzialny za kąt wzięty z punktu pierwszego.

Tym razem funkcja kosztu, z której korzystał algorytm optymalizacyjny, przybrała następującą postać:

$$J_2 = \begin{cases} \int_0^{\infty} 100dt & \text{jeśli } |\alpha| \geq 0.31, \\ \int_0^{\infty} (|e_\alpha(t)| + |e_c(t)|) dt & \text{jeśli } |\alpha| < 0.31, \end{cases} \quad (3.4)$$

gdzie:

$\alpha$  - wartość kąta wahadła,

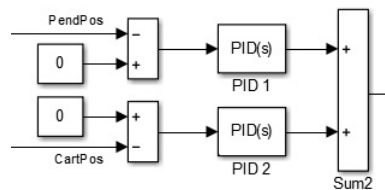
$e_\alpha$  - różnica między wartością zadaną i aktualną wartością kąta wahadła,

$e_c$  - różnica między wartością zadaną i aktualną wartością położenia wózka.

Ponownie, stała 100 ma za zadanie informować algorytm, że niedopuszczalna jest sytuacja, gdy wahadło wypada poza obszar wokół punktu zadanego.

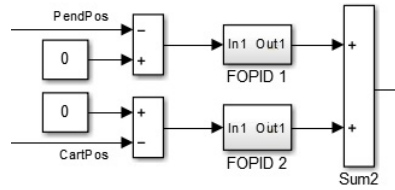
Przygotowano między innymi dwa rodzaje układów sterowania:

1. **wariant** - dwa kontrolery PID - rysunek 3.17.



Rysunek 3.17: Sterowanie wahadłem odwróconym - pierwszy wariant

2. **wariant** - dwa kontrolery FOPID - rysunek 3.18.

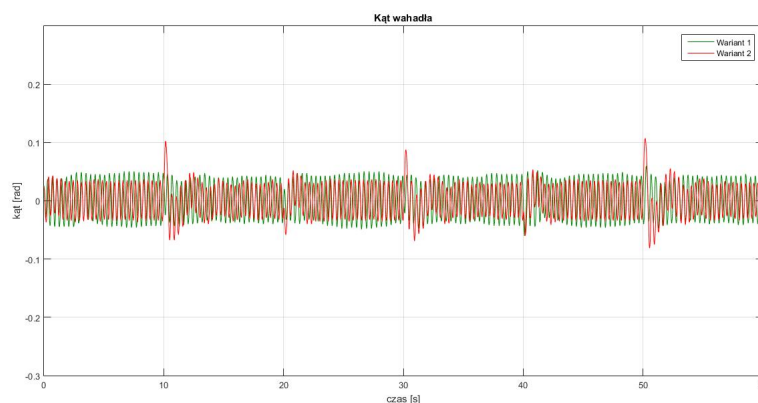


Rysunek 3.18: Sterowanie wahadłem odwróconym - drugi wariant

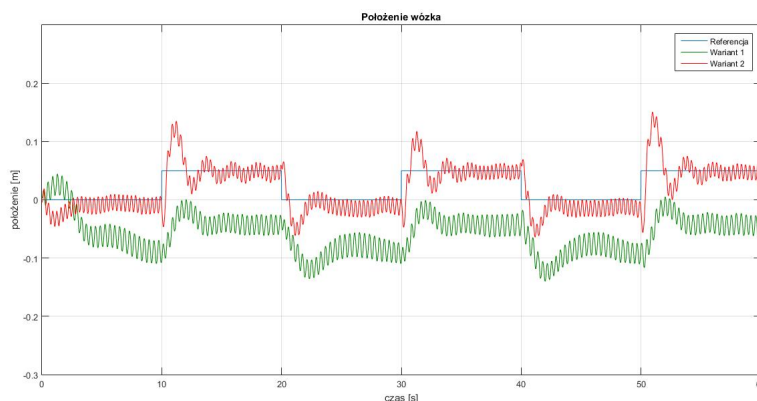
W celu znalezienia parametrów regulatorów wykorzystano algorytm optymalizacyjny GWO [10] korzystający z obu funkcji kosztu (3.3) oraz (3.4). Wybrane wartości przedstawiono w tabeli 3.7. Przy sterowaniu FOPID wykorzystano aproksymację rekurencyjną Oustaloupa z rzędem aproksymacji 8 oraz zakresem częstotliwości w przedziale  $\omega \in [10^{-3}, 10]$ . Pomiarzy przeprowadzono ze stałym krokiem 0.01.

Tabela 3.7: Współczynniki regulatorów wahadła odwróconego

1. wariant	P	I	$\lambda$	D	$\mu$
PID 1	18.84	37.99	—	0.21	—
PID 2	3.90	0.80	—	0.23	—
2. wariant	P	I	$\lambda$	D	$\mu$
FOPID 1	31.99	0.73	0.61	0.093	0.66
FOPID 2	1.14	1.66	0.017	2.39	0.61



Rysunek 3.19: Odpowiedź systemu na zmienne wartości zadane - kąt wahadła



Rysunek 3.20: Odpowiedź systemu na zmienne wartości zadane - położenie wózka

Rysunki 3.19 - 3.20 przedstawiają odpowiedź systemu podczas eksperymentów ze zmienną wartością zadaną dla wózka. W tabeli 3.8 spisano wartości funkcji kosztu (3.4). Wariant drugi osiągnął zdecydowanie lepszy wynik. W wersji pierwszej należy wziąć pod uwagę fakt, że zmiana wartości zadanej z 0.05 na 0 była pozytywna dla tego wariantu, gdyż znajdowała się bliżej wcześniejszego uchybu ustalonego. To obniżyło wyliczoną wartość funkcji kosztu.

Tabela 3.8: Wartości funkcji kosztu (3.4) dla wahadła odwróconego ze zmiennymi wartościami zadanymi

	Funkcja kosztu
1. wariant	519.55
2. wariant	343.42

Tabele 3.9 - 3.10 przedstawiają zebrane pomiary czasów wykonywania obliczeń przez oba warianty sterowania w eksperymencie. Regulatory FOPID były zdecydowanie wolniejsze niż ich klasyczne odpowiedniki. W przypadku PID 1 (pierwszy wariant) oraz FOPID 2 (drugi wariant) maksymalny czas obliczeń był większy o rząd wielkości względem typowej wartości. Mimo wszystko wynik będący na poziomie  $10^{-6}$  wciąż jest świetnym osiągnięciem. To aż cztery rzędy wielkości mniej niż krok, który wynosił 0.01.

Tabela 3.9: Średni i maksymalny czas obliczeń regulatorów - pierwszy wariant

Regulator	Średni czas obliczeń [s]	Maksymalny czas obliczeń [s]
PID 1	5.6440e-08	5.6995e-07
PID 2	1.0769e-07	5.7745e-07

Tabela 3.10: Średni i maksymalny czas obliczeń regulatorów - drugi wariant

Regulator	Średni czas obliczeń [s]	Maksymalny czas obliczeń [s]
FOPID 1	1.3151e-07	7.9371e-07
FOPID 2	2.1874e-07	1.4162e-06



## Rozdział 4

# Podsumowanie

W dysertacji zbadano działanie regulatorów Fractional Order PID (FOPID) podczas sterowania trzech obiektów. Były to: suwnica 3D, dwurotorowy system aerodynamiczny oraz wahadło odwrócone. Jakość sterowania została porównana względem klasycznych odpowiedników PID, między innymi na podstawie porównania wskaźników jakości regulacji zarówno bezpośrednich, jak i całkowych. W związku z tym, że wszystkie systemy są nieliniowe i ich dynamika jest stosunkowo szybka, to zaistniała możliwość doświadczalnego zbadania, czy regulatory FOPID mogą być stosowane w sterowaniu tą klasą obiektów. Oznacza to między innymi konieczność spełnienia twardych wymagań czasu rzeczywistego.

Suwnica 3D była systemem, który wymagał największej ilości regulatorów - aż pięciu. Wiązało to się z potrzebą minimalizacji uchybów na osiach  $X$ ,  $Y$ ,  $Z$  oraz kątów  $\alpha$ ,  $\beta$ . Ilość sygnałów wejściowych była ograniczona do trzech. W związku z tym wystąpiła potrzeba zgrupowania regulatorów. Kontroler związany z uchybem na osi  $X$  połączono z drugim, który korzystał z wartości  $\alpha$ . Podobnie postąpiono z odchyleniem na osi  $Y$  oraz kątem  $\beta$ . Zgrupowane regulatory połączono równolegle. Niezależnie od nich obliczane było sterowanie związane z uchybem na osi  $Z$ . Po dokonaniu optymalizacji współczynników kontrolerów z użyciem algorytmu Grey Wolf Optimizer (GWO) symulacyjnie sprawdzono działanie obiektu. W wersji z pięcioma regulatorami FOPID osiągnięto najlepsze wartości zastosowanej funkcji kosztu (3.1). Przesterowania na osiach  $X$ ,  $Y$ ,  $Z$  były stosunkowo małe. Także szybko następowała stabilizacja wokół zadanych wartości. W przypadku pomiarów kątów tłumienie oscylacji również było bardzo sprawne. Obserwacje potwierdziły wartości funkcji kosztu.

Kolejnym badanym obiektem był dwurotorowy system aerodynamiczny. Dwa wyjścia wykorzystywano do obliczenia sterowania dla dwóch wejść. Jednak przez zjawisko sprzężenia krzyżowego zaistniała potrzeba zastosowania czterech regulatorów. Wiąże się to z tym, że każdy z rotorów wpływa zarówno na kąt nachylenia, jak i kąt azymutowy. Wszystkie współczynniki kontrolerów zoptymalizowano z wykorzystaniem algorytmu Grey Wolf Optimizer (GWO). W sterowaniu hybrydowym wariant z samymi regulatorami PID zaczął zauważalnie odstawać pod względem wartości funkcji kosztu. Warto zauważyć, że przy tego typu sterowaniu kąt nachylenia był bardziej odporny na zakłócenia związane z nagłą zmianą referencji dla wartości kąta azymutowego.

Na końcu przeprowadzono badania na wahadle odwróconym. Do celów sterowania wykorzystywano dwa wyjścia - położenie wózka i kąt odchylenia wahadła. Z uwagi na to, że obiekt przyjmował tylko jedną

wartość wejściową, zdecydowano się zastosować sterowanie równoległe. Przygotowano między innymi wariant z samymi regulatorami FOPID oraz drugi tylko z ich klasycznymi odpowiednikami. Współczynniki zoptymalizowano za pomocą algorytmu Grey Wolf Optimizer (GWO). Jednak proces doboru nastaw podzielono na dwa etapy. W pierwszym brano pod uwagę tylko kąt odchylenia wahadła. W drugim zoptymalizowano nastawy regulatora wózka przy sztywnych współczynnikach sterownika związanego z położeniem wahadła (wzięte z poprzedniej fazy). Oba warianty zachowywały się w dość zbliżony sposób. Jednak największą zauważalną różnicą było powstanie dużego uchybu ustalonego związanego z położeniem wózka w wersji z regulatorami PID. W wyniku tego zachowania wartość funkcji kosztu (3.4) znacząco wzrosła dla tej opcji. Biorąc pod uwagę powyższe wyniki, można stwierdzić, iż wariant z regulatorami FOPID lepiej spisuje się w rzeczywistości.

Oprócz jakości sterowania pod względem funkcji kosztu sprawdzono również czasy wykonania obliczeń podczas wyznaczania sygnałów sterujących przez kontrolery. Zgodnie z przewidywaniami regulatory PID działały szybciej w stosunku do FOPID, jednak różnice były niewielkie. Mała rozpiętość pomiędzy czasami obu wersji kontrolerów wskazuje na to, że komputer starał się optymalizować wykonywanie obliczeń tak, aby wszystkie wartości zostały przekazane do obiektu w zbliżonym momencie. W wersji hybrydowej dla suwnicy 3D zanotowano większe różnice, przede wszystkim dotyczące regulatorów podrzędnych w pętli kaskadowej. Należy jednak wziąć pod uwagę, że taki kontroler, oznaczony w tych wariantach numerem 3, musiał czekać na wykonanie innych operacji. Toteż zanim zaczynał swoją pracę, to większość innych regulatorów kończyła swoje działania, przez co zwalniały one moc obliczeniową w komputerze. Jeśli jednak pod uwagę zostaną wzięte pomiary z wszystkich trzech obiektów, to za każdym razem w najgorszym przypadku wartości maksymalne notowano na poziomie  $10^{-6}$ . Jest to bardzo dobry wynik, biorąc pod uwagę stosowane kroki czasowe - 0.01 dla systemu TRAS i wahadła odwróconego oraz 0.0005 dla suwnicy 3D. Zatem powstała różnica pozwala wnioskować, że regulatory FOPID stosujące aproksymację Oustaloup Recursive Approximation (ORA) bardzo szybko wykonują swoje obliczenia i są w stanie spełnić twarde wymagania czasu rzeczywistego dla szybkich obiektów.

# Bibliografia

- [1] R. Caponetto, G. Dongola, L. Fortuna, and I. Petras. *Fractional Order Systems: Modeling And Control Applications*. World Scientific Series On Nonlinear Science Series A. World Scientific Publishing Company, 2010.
- [2] M. Caputo. Linear Models of Dissipation whose Q is almost Frequency Independent—II. *Geophysical Journal International*, 13(5):529–539, 1967.
- [3] Y. Q. Chen. Oustaloup-Recursive-Approximation for Fractional Order Differentiators (<https://www.mathworks.com/matlabcentral/fileexchange/3802-oustaloup-recursive-approximation-for-fractional-order-differentiators>), MATLAB Central File Exchange, 2023. Retrieved 5.05.2023.
- [4] S. Das. *Functional Fractional Calculus for System Identification and Controls*. Springer Berlin, Heidelberg, 2010.
- [5] P. J. Davis. Leonhard Euler’s Integral: A Historical Profile of the Gamma Function: In Memoriam: Milton Abramowitz. *The American Mathematical Monthly*, 66(10):849–869, 1959.
- [6] L. Dorčák. *Numerical Models for the Simulation of the Fractional-Order Control Systems*. Uef-04-94, The Academy of Sciences, Inst. of Experimental Physic, Košice, Slovakia, 1994.
- [7] A. A. Kilbas. Hadamard-type fractional calculus. *Journal of the Korean Mathematical Society*, 38(6):1191–1204, 2001.
- [8] A.A. Kilbas, H.M. Srivastava, and J.J. Trujillo. *Theory and Applications of Fractional Differential Equations*. North-Holland Mathematics Studies. Elsevier Science, 2006.
- [9] N. Minorsky. Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309, 1922.
- [10] S. Mirjalili, S. M. Mirjalili, and A. Lewis. Grey Wolf Optimizer. *Advances in Engineering Software*, 69:46–61, 2014.
- [11] A. Obrączka. *Sterowanie procesów cieplnych z wykorzystaniem modeli niecałkowitego rzędu*. Doctoral dissertation, AGH University of Science and Technology, 2015.
- [12] K. Oldham and J. Spanier. *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order*. Dover Books on Mathematics. Dover Publications, 2006.

- [13] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot. Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(1):25–39, 2000.
- [14] I. Petras. *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation*. Nonlinear Physical Science. Springer Berlin, Heidelberg, 2011.
- [15] I. Podlubny. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, volume 198 of *Mathematics in Science and Engineering*. Academic Press, 1999.
- [16] I. Podlubny. Fractional-order systems and  $PI^\lambda D^\mu$ -controllers. *IEEE Transactions on Automatic Control*, 44(1):208–214, 1999.
- [17] I. Podlubny, I. Petráš, B. M. Vinagre, P. O’Leary, and L. Dorčák. Analogue Realizations of Fractional-Order Controllers. *Nonlinear Dynamics*, 29(1):281–296, 2002.
- [18] M. I. Syam and M. Al-Refai. Fractional differential equations with Atangana–Baleanu fractional derivative: Analysis and applications. *Chaos, Solitons & Fractals: X*, 2:100013, 2019.
- [19] D. Xue, Y. Q. Chen, and D. P. Atherton. *Linear Feedback Control: Analysis and Design with MATLAB*. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2007.