

**INSTYTUT PODSTAW KONSTRUKCJI MASZYN**  
**Wydział Mechaniczny Technologiczny**  
**POLITECHNIKA ŚLĄSKA W GLIWICACH**

**PRACA DYPLOMOWA MAGISTERSKA**

**System sterowania prototypu manipulatora o strukturze  
kinematycznej 6R bazujący na logice rozmytej**

**Marek ŚLIŹ**  
**Kierunek studiów: AiR**  
**Specjalizacja: Projektowanie i Eksploatacja Maszyn**

GLIWICE 2014


## Spis treści

1.	Wstęp.....	9
2.	Cel i zakres pracy .....	11
3.	Przegląd literatury .....	12
4.	Logika rozmyta .....	16
4.1	Fuzyfikacja.....	16
4.2	Inferencja (wnioskowanie).....	17
4.3	Defuzyfikacja .....	21
5.	Regulator PID.....	22
5.1	Charakterystyki sterowań P, I, D .....	22
5.2	Strojenie parametrów regulatora.....	23
6.	Układ sterujący manipulatorem .....	26
6.1	Układ sterowania .....	26
6.2	Niska warstwa oprogramowania.....	28
7.	Aplikacja sterująca .....	31
8.	Projekt systemu sterowania.....	39
9.	Implementacja systemu .....	44
10.	Weryfikacja działania systemu.....	49
11.	Podsumowanie .....	67
	Bibliografia.....	68



## 1. Wstęp

Manipulatory przemysłowe stanowią obecnie nieodzowną część przemysłu [2]. Dzięki swojej niezawodności, powtarzalności ruchów oraz brakowi potrzeby odpoczynku, skutecznie wyparły człowieka z wykonywania niektórych prac. Na rynku dostępne są rozwiązania robotów takich firm jak np. FANUC oraz KUKA [2]. Wyzwaniem dla współczesnej techniki jest miniaturyzacja tych rozwiązań z zachowaniem większości założeń stawianych robotom przemysłowym [2]. Poza problemami miniaturyzacji równie ważne są problemy sterowania na które wpływ ma wiele czynników takich jak zastosowane napędy, momenty bezwładności, zakresy ruchów, ilość stopni swobody itp.

Z punktu widzenia teorii sterowania manipulator przemysłowy jest nieliniowym wielowymiarowym układem dynamicznym a jego system stanowią układy składające się z elementów. O dynamice manipulatora przemysłowego decyduje dynamika elementów składowych układów sterowania, układów napędowych i manipulatorów wraz z wzajemnym ich oddziaływaniem [1].

Na rysunku 1 przedstawiono prototyp manipulatora wraz z sterownikiem dla którego zaprojektowano system sterowania. W konstrukcji zdecydowano się na zastosowanie przekładni wykonanych z tworzywa sztucznego oraz stali. Trzy dolne osie manipulatora wyposażono w przekładnie zębate o trybach stalowych. Przekładnie tego typu charakteryzują się małymi rozmiarami oraz dobrą wytrzymałością, dodatkowo ostatni stopień przekładni jest łożyskowany. Przełożenie przekładni wynosi 1:280. Dwie następne osie wyposażono w napędny o przekładniach wykonanych z tworzywa sztucznego. Przełożenia ich kolejno wynoszą 1:160 oraz 1:180. Ostatnią z osi, wyposażono w mikrosilnik DC Pololu o przełożeniu 1:298. Główną zaletą tego silnika wraz z przekładnią, są jego małe wymiary wynoszące 10 mm x 12 mm x 26 mm. W górnych osiach starano się unikać stosowania elementów wykonanych ze stali. W układach napędowych zastosowano szczotkowe silniki prądu stałego. Silniki osi pierwszej - czwartej zasilane są napięciem 12V, silnik osi piątej i szóstej kolejno 5V i 3,3V.



Rys. 1. Manipulator wraz z sterownikiem

Na potrzeby minimalizacji masy do budowy zastosowano tworzywo sztuczne PMMA. Jego atutem jest możliwość łatwego formowania kształtu po podgrzaniu do określonej temperatury. Części narażone na duże obciążenie, takie jak platforma obrotowa oraz główne mocowanie ramienia, wykonano z aluminium. Pozwoliło to zwiększyć sztywność całego zespołu. Sworznie łączące oś pierwszą z platformą obrotową oraz osie drugą i trzecią z ramieniem wykonano ze stali. Całą konstrukcję przymocowano do stalowej płyty w celu zapewnienia stateczności. Cała konstrukcja waży około 1,35 kg a wymiary wynoszą 280 mm wysokości oraz 250 mm długości w pozycji przedstawionej na rysunku 1.

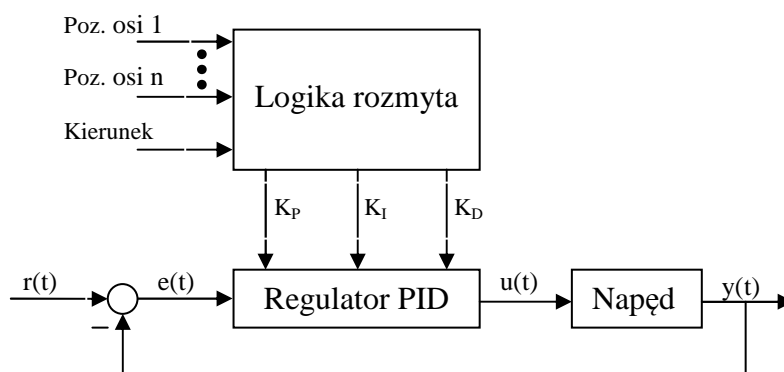
Manipulator może być sterowany bezpośrednio za pomocą przycisków umieszczonych na sterowniku lub po podłączeniu go do komputera za pomocą dedykowanej aplikacji, która umożliwia tworzenie sekwencji ruchowych, rysowanie wykresów wymuszeń, prądu oraz kalibrację.

## 2. Cel i zakres pracy

W niniejszej pracy, z powodu trudności związanych ze zbudowaniem modelu dynamicznego manipulatora opracowany zostanie alternatywny system sterowania.

Celem pracy jest zaprojektowanie oraz weryfikacja systemu sterowania z wykorzystaniem logiki rozmytej dla miniatury manipulatora przemysłowego o sześciu obrotowych stopniach swobody opracowanego przez M. Śliżę. W proponowanym rozwiązaniu, logika rozmyta jest stosowana w celu wyznaczenia optymalnych nastaw regulatora PID w zależności od aktualnej i docelowej pozycji. Tego typu system powinien zapewnić lepszą kontrolę oraz działanie manipulatora. W opisywanym przypadku istnieją trudności z opracowaniem odpowiedniego modelu dynamiki z powodu braku danych. W takim przypadku można próbować przybliżać te wartości lub wyznaczyć je metodami laboratoryjnymi. W pracy zaproponowana została alternatywa dla tego rozwiązania, system wykorzystujący zalety wnioskowania rozmytego oraz dokładnego sterowania regulatora PID. Schemat systemu przedstawiono na rysunku 2.

Zakres pracy obejmuje m.in. przegląd literatury na temat wykorzystania logiki rozmytej w systemach sterowania manipulatorami, przeprowadzenie analizy możliwych rozwiązań systemu, opracowanie algorytmu regulatora oraz weryfikację poprawności zaprojektowanego systemu podczas działania manipulatora.



Rys. 2. Schemat regulatora rozmytego

gdzie:

Poz. osi 1 - Poz. osi n, Kierunek – dane wejściowe logiki rozmytej,

$r(t)$  – wielkość zadana,

$e(t)$  – uchyb,

$u(t)$  – sygnał sterujący,

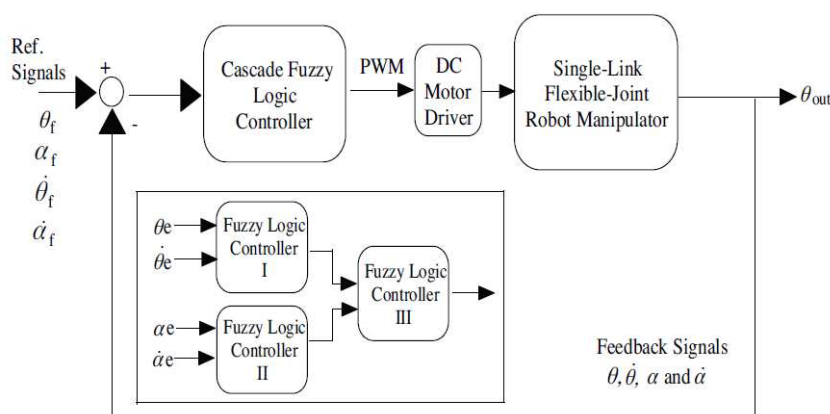
$y(t)$  - wyjście z obiektu.

### 3. Przegląd literatury

Literatura na temat wykorzystania logiki rozmytej w systemach sterowania manipulatorów o podobnej budowie co omawiany jest stosunkowo skąpa.

Pod nazwą regulator rozmyty (ang. *fuzzy logic controller* – FLC) rozumiemy prawo sterowania, które jest opisane przez system o bazie reguł zapisanej w postaci JEŻELI – TO przy nieokreślonych predyktantach. Przykładem wykorzystania jest regulacja temperatury w pomieszczeniach, lodówkach, żelazkach elektrycznych, regulacja obrotów silników elektrycznych w niektórych urządzeniach, i inne. Jeżeli regulator spełnia postawione przed nim zadania to jego stosowanie jest całkowicie uzasadnione, zwłaszcza jeśli uwzględni się jego niską cenę [3].

Autorzy I.H. Akyuz, Z. Bingul, S. Kizir w swoim artykule opisują działanie FLC. Jego zadaniem jest usunięcie wibracji i zmniejszenie czasu osiągnięcia pozycji przez elastyczne ramię manipulatora. Opisany system sterowania składa się z trzech regulatorów rozmytych połączonych kaskadowo. Wejściami do pierwszych dwóch regulatorów jest: błąd kąta obrotu, jego pochodna, błąd kąta odbicia oraz jego pochodna. Wyjścia z tych dwóch regulatorów są wejściem do trzeciego który generuje sygnał do napędu manipulatora. Autorzy podczas generowania wymuszeń skokowych, co stanowi podstawę weryfikacji działania układu opisują, że uzyskane przeregulowania są mniejsze niżeli  $0,12^\circ$ . Dodatkowo autorzy opisują, że system wykorzystujący logikę rozmytą jest bardzo stabilny, odporny na błędy i elastyczny. W tym celu przeprowadzili dodatkowe badania zmieniając parametry manipulatora takie jak długość ramienia, sztywność sprężyny oraz badali odpowiedź regulatora rozmytego [11]. Na rysunku 3 przedstawiono opisywany model FLC.



Rys. 3. Struktura systemu kaskadowo połączonych FLC [11]

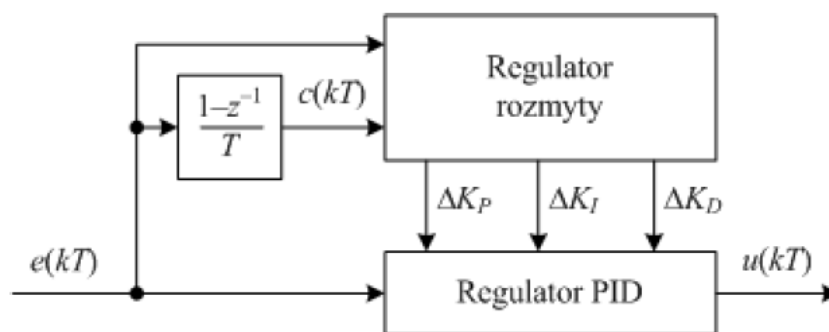


W pracy M. Tomera wykonano rozmyty samonastrajalny regulator cyfrowy PID [10]. Łączy on prostotę regulatora PID oraz zalety wnioskowania rozmytego. Zaprojektowany układ składa się z regulatora rozmytego i regulatora PID. Zadaniem regulatora rozmytego jest strojenie trzech wzmocnień regulatora PID na podstawie uchybu i szybkości zmian uchybu. Zaprojektowany regulator zastosowany został do sterowania poziomem wody w dolnym zbiorniku w układzie połączonych kaskadowym dwóch zbiorników. Rozmyty regulator PID ma poniższe właściwości:

- liniową strukturę jak konwencjonalny regulator PID, lecz samonastrajalne wzmocnienia sterowania,  $K_P$ ,  $K_I$ ,  $K_D$ . Są one nieliniowymi funkcjami sygnałów wejściowych.
- Regulator PID zaprojektowany jest w oparciu o teorię klasycznego regulatora cyfrowego PID.

Ogólny schemat regulatora przedstawiony został na rysunku 4. W wymienionej pracy regulator rozmyty składa się z bloku: rozmywania, bazy reguł, wnioskowania rozmytego, wyostrzania. Posiada on dwa wejścia oraz trzy wyjścia i jest regulatorem trójkanałowym. Funkcje przynależności są trójkątne a przestrzeń rozmyta każdej zmiennej lingwistycznej podzielona została na siedem wartości. Wejściami są szybkość zmian uchybu oraz wartość uchybu.

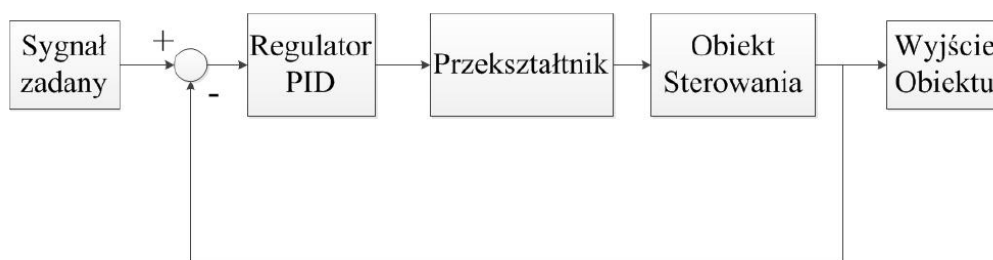
Baza reguł została zbudowana na podstawie obserwacji charakterystyk dynamicznych oraz analiz wielu badań symulacyjnych [10]



Rys. 4. Struktura rozmytego regulatora PID [10]

Efektem końcowym zastosowania tego regulatora było sprowadzenie do zera maksymalnego przeregulowania oraz skrócenie czasu regulacji. Natomiast wadą była konieczność przygotowania odpowiedniej bazy reguł, do utworzenia której przeprowadzono liczne symulacje na modelu matematycznym obiektu. Trafność bazy reguł zależy bezpośrednio od dokładności tego modelu [10].

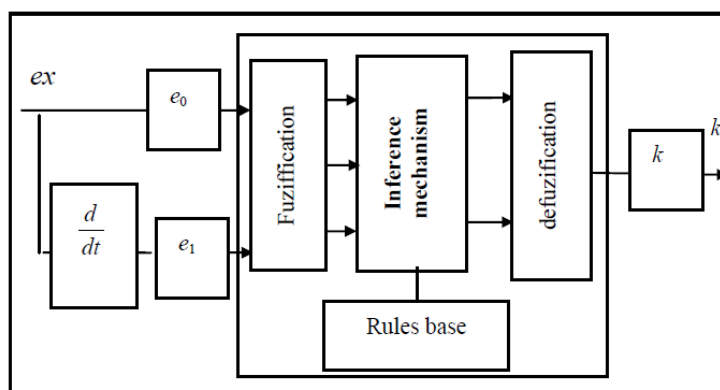
Z kolei Ł. Niewiara oraz K. Zawirski w swoim artykule przedstawili koncepcję auto-strojenia regulatora PID z wykorzystaniem bloku logiki rozmytej. Proces auto-strojenia regulatora typu PID pozwala na minimalizację wpływu czynnika ludzkiego przy doborze nastaw. Takie podejście przyspiesza dobór odpowiednich nastaw, które generowane są automatycznie podczas kolejnych iteracji. Podejście to wykorzystuje zalety nieprecyzyjnego opisu wskaźników jakości procesu regulacji za pomocą odpowiednich zmiennych lingwistycznych oraz baz reguł [13]. Ogólny schemat układu zaprezentowano na rysunku 5.



Rys. 5. Schemat układu auto-strojenia regulatora PID [13]

Badania symulacyjne przeprowadzone przez autorów wykazały, że zaprojektowany układ po podaniu wstępnych nastaw i około 15 iteracjach osiągnął zadowalająco małe przeregulowania [13].

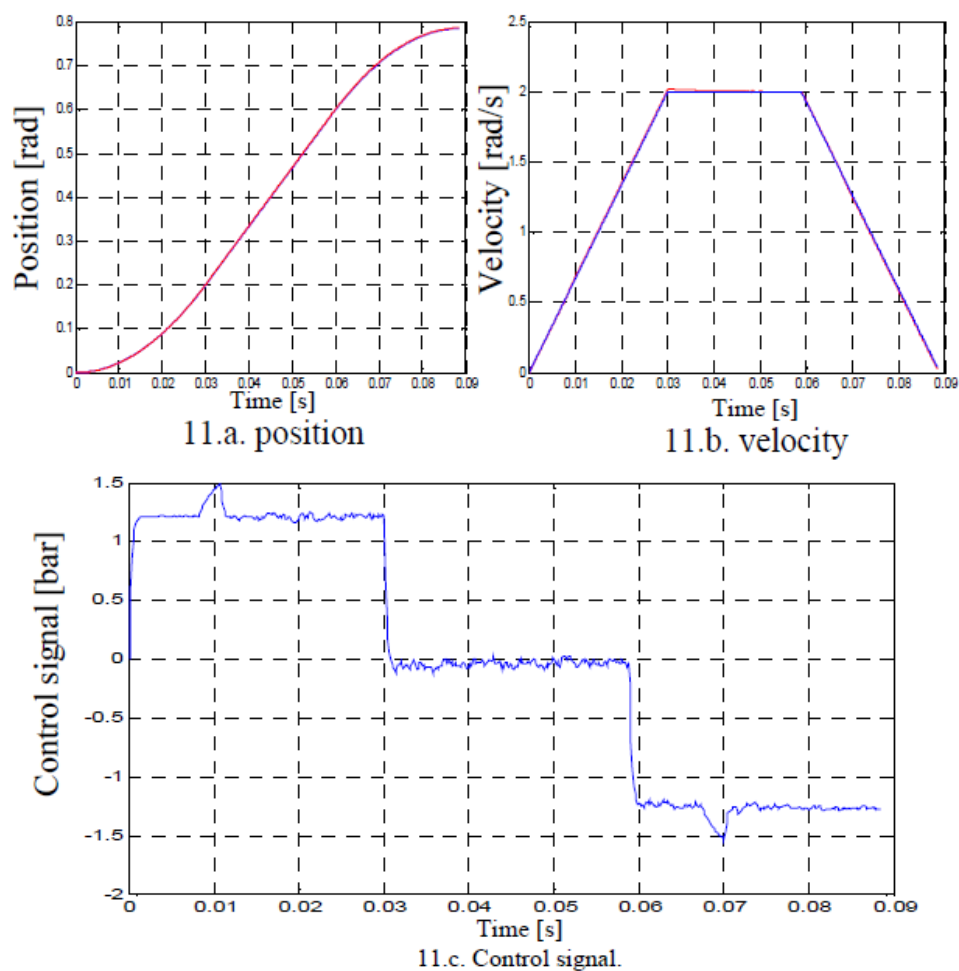
W artykule autorstwa A. Rezoug, S. Boudoua oraz F. Hamerlain'a opisany został regulator rozmyty sterujący prędkością i położeniem pneumatycznych silników liniowych o działaniu zbliżonym do mięśni. Model systemu FLC został przedstawiony na rysunku 6.



Rys. 6. Model systemu logiki rozmytej [12]

Model został przygotowany w FLT (ang. *Fuzzy Logic Toolbox* - pakiet programu Matlab). Wejściami do układu jest błąd położenia oraz jego pochodna. Przy tworzeniu reguł wykorzystana

została wiedza eksperdzka na temat zachowywania się obiektu. Wyniki przeprowadzonych testów są zadowalające. Jedną z największych zalet sterownika jest brak konieczności opracowywania modelu dynamiki ramion. Na rysunku 7 przedstawiono końcowe wyniki badań autorów. Wymuszenia badanego napędu miały charakter skokowy [12].



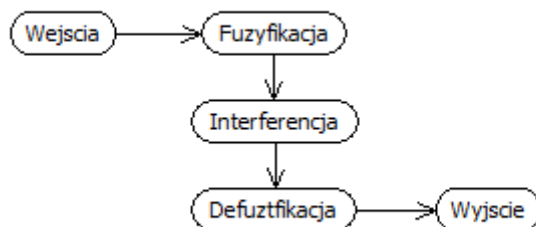
Rys. 7. Wyniki badań trzeciej osi manipulatora

## 4. Logika rozmyta

Pojęcie zbioru rozmytego zostało wprowadzone przez L. A. Zadeha w 1965r. Celem była chęć modelowania złożonych procesów, w szczególności obejmujących udział czynnika ludzkiego. W logice klasycznej element może należeć do zbioru (1) lub do niego nie należeć (0). Natomiast w logice rozmytej ten sam element należy do zbioru  $\langle 0,1 \rangle$  przyjmując dowolną wartość z tego zakresu. Wprowadzając pewną dozę niedokładności, zyskujemy zdolność do modelowania złożonych procesów [3].

Za pierwsze zastosowanie logiki rozmytej uważane jest użycie jej do regulacji systemu wytwarzania pary w elektrowni zrealizowane przez Assiliana w 1974r. Wytwornica pary okazała się zbyt złożonym i nie liniowym procesem, że zastosowanie tradycyjnych metod sterowania nie pozwoliło na rozwiązanie problemu. Jednak to zastosowanie nie spopularyzowało logiki rozmytej [4].

Za przełom uważa się układ sterowania metra Sendai opracowany przez koncern Hitachi. Układ automatycznie zmniejszał prędkość pociągu przy wjeździe na stację i zapewniał dokładne jego zatrzymanie w ściśle określonym miejscu. Układ ten został zaprojektowany na podstawie wiedzy maszynisty a jego próby i udoskonalenia trwały 2 lata [4]. Struktura typowego modelu rozmytego przedstawiona jest na rysunku 8.

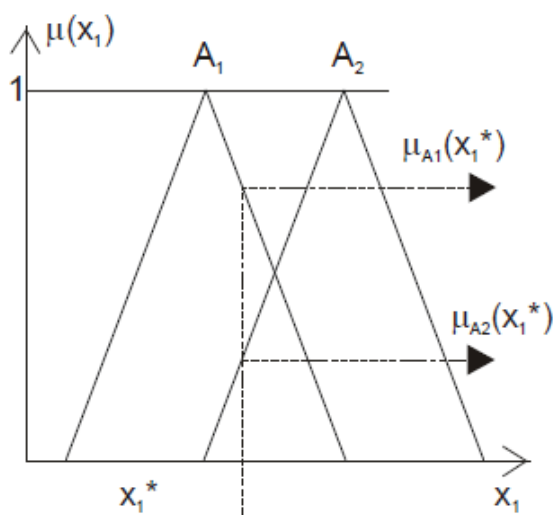


Rys. 8. Struktura modelu rozmytego

### 4.1 Fuzyfikacja

Fuzyfikacja jest pierwszym stopniem logiki rozmytej, przyjmująca ostre wartości oraz rozmywająca je. W jej ramach obliczane są stopnie przynależności wartości wejść modelu do odpowiadających im zbiorów rozmytych. Aby obliczyć stopnie przynależności do poszczególnych zbiorów, funkcje te muszą być precyzyjnie zdefiniowane jakościowo (rodzaj funkcji), jak i ilościowo (parametry, współczynniki funkcji). Zarówno parametry, jak i kształt

funkcji przynależności mają duży wpływ na dokładność modelu. Zbiory rozmyte opisują najczęściej pojęcia lingwistyczne używane w życiu codziennym jak na przykład „chłodno”, „ciepło”, „bardzo wolno” itp [8]. Takie opisywanie zbiorów sprawia, że logika rozmyta staje się bardziej naturalna, bliższa opisywaniu zdarzeń, przedmiotów w sposób jaki robi to człowiek w normalnym życiu. Na rysunku 9 przedstawiono przykładowy zbiór rozmyty.



Rys. 9. Przykładowy zbiór rozmyty

W procesie fuzyfikacji ostry wektor  $X^*$  wejść przekształcony zostaje w wektor  $\mu$  stopni przynależności, będący wyjściem z bloku fuzyfikacji [4]. Funkcje przynależności mogą mieć różne kształty (trapezoidalny, funkcji Gaussa, itp.)

## 4.2 Inferencja (wnioskowanie)

Blok inferencja oblicza na podstawie wejściowych stopni przynależności  $\mu_{A1}(x_i)$ ,  $\mu_{A2}(x_i)$  wynikową funkcję przynależności  $\mu_{wyn}(y)$ . Funkcja ta może mieć często złożony kształt. Aby zrealizować operację inferencji (wnioskowania) blok musi zawierać następujące elementy:

- bazę reguł,
- mechanizm interferencji,
- funkcję przynależności wyjścia  $y$ .

W bazie reguł niej zawarta jest wiedza eksperdzka na temat układu. Zawiera ona reguły logiczne określające zależności przyczynowo skutkowe istniejące w systemie pomiędzy zbiorami rozmytymi wejść i wyjść. Przykładowa baza reguł może mieć postać:

Tabela 1. Przykładowa baza reguł [10]

$\Delta K_{Pk}$		$E_i$						
		UD	US	UM	ZE	DM	DS	DD
$C_j$	UD	UD	UD	UM	ZE	DM	DS	DD
	US	UD	UD	US	US	UM	ZE	ZE
	UM	UD	US	UM	UM	UM	ZE	DM
	ZE	US	UM	UM	UM	ZE	DM	DM
	DM	US	UM	UM	ZE	ZE	DM	DM
	DS	UM	ZE	ZE	ZE	DM	DS	DS
	DD	ZE	UM	ZE	DM	DS	DS	DD

Wówczas każdą regułę można zapisać w postaci:

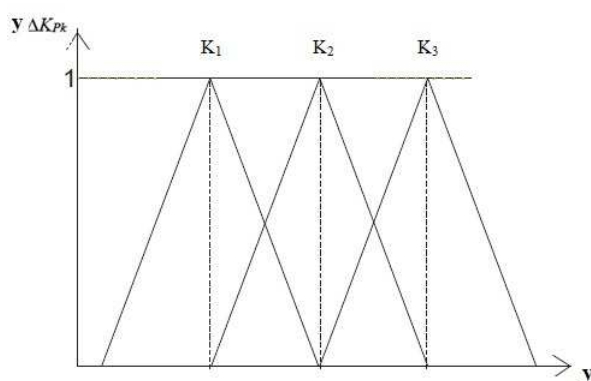
R1 : **JEŚLI** ( $E_i = ZE$ ) **I** ( $C_j = ZE$ ) **TO** ( $\Delta K_p = K_1$ ).

R2 : **JEŚLI** ( $E_i = ZE$ ) **I** ( $C_j = DD$ ) **TO** ( $\Delta K_p = K_2$ ).

R3 : **JEŚLI** ( $E_i = DD$ ) **I** ( $C_j = ZE$ ) **TO** ( $\Delta K_p = K_1$ ).

R4 : **JEŚLI** ( $E_i = DD$ ) **I** ( $C_j = DD$ ) **TO** ( $\Delta K_p = K_3$ ).

Natomiast na rysunku 10 przedstawiono przykładową funkcję wyjścia ( $K_1$  – mały,  $K_2$  – średni,  $K_3$  – duży).



Rys. 10. Przykładowe funkcje przynależności

Zadaniem mechanizmu inferencyjnego jest obliczanie wynikowej funkcji przynależności  $\mu_{\text{wyn}}(y)$ . Składa się ona z trzech części:

- Części obliczającej stopień spełnienia przesłanek poszczególnych reguł.

- Części obliczającej stopień aktywacji konkluzji poszczególnych reguł.
- Części określającej wynikową postać funkcji przynależności wyjścia  $\mu_{\text{wyn}}(y)$ .

Popularnymi metodami wnioskowania są metody Mamdaniego oraz Larensa. Metoda Mamdaniego wykorzystuje operacje minimum i maksimum. Wielkości wejściowe  $x_0, y_0$  pochodzące np. z czujników pomiarowych można traktować jako singletony (zbiór rozmyty który zawiera jeden element) rozmyte. Mamdani zaproponował wyznaczenie wagi reguł jako:

$$w_1 = \min[m_{A_1}(x_0), m_{B_1}(y_0)], \quad (1)$$

gdzie:

$m_{A_1}(x_0), m_{B_1}(y_0)$  - są wartościami funkcji przynależności liczb  $x_0, y_0$  do zbiorów  $A_1, B_1$ .

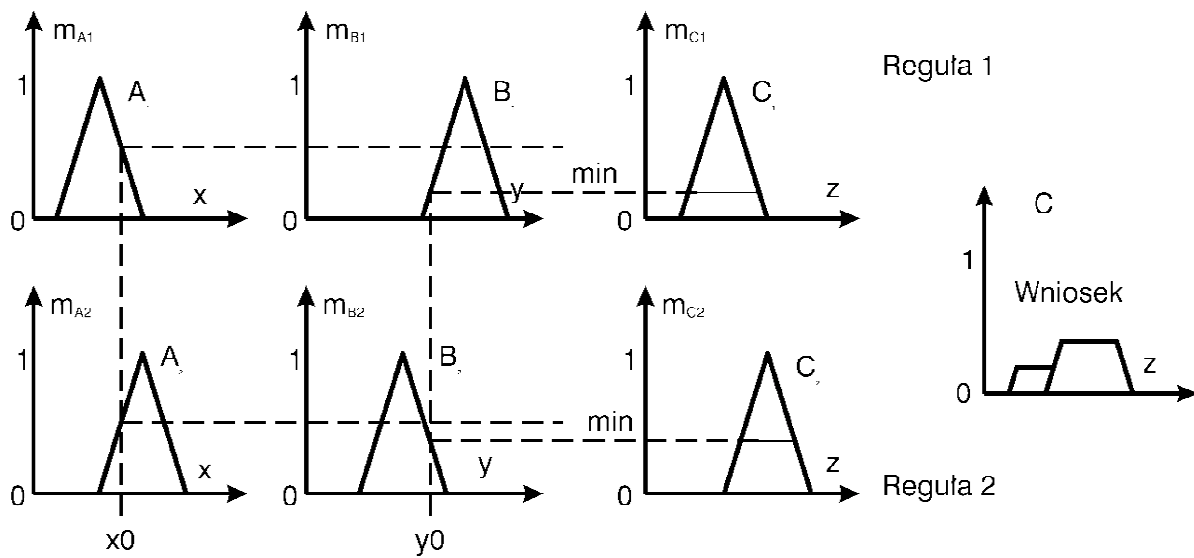
Decyzję stanowi zbiór rozmyty  $C_1$  o funkcji przynależności:

$$m_{C_1}(z) = \min[w_1, m_{C_1}(z)]. \quad (2)$$

Wniosek z obu reguł stanowi zbiór rozmyty  $C$  o funkcji przynależności:

$$m_C = \max[m_{C_1}, m_{C_2}] = \max [\min(w_1, m_{C_1}), \min(w_2, m_{C_2})]. \quad (3)$$

Ze względu na tę zależność, metodę nazywa się czasem max-min. Powyższą regułę najważniejszą w praktyce zobrazowano na rysunku 11. Jest ona powszechnie stosowana w teorii sterowania. Metoda Larsena jest bardzo podobna do wnioskowania Mamdaniego. Wagi reguł oblicza się identycznie jak minimum z wartości (1) ale decyzję stanowi zbiór rozmyty  $C_1$ , o funkcji przynależności będącej iloczynem [8]:



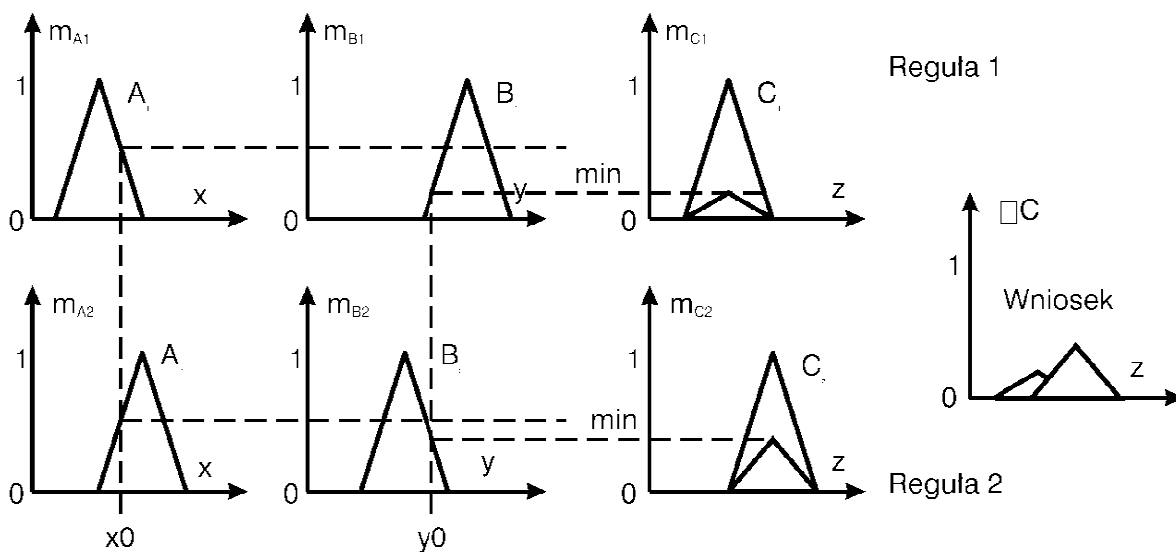
Rys. 11. Wnioskowanie według Mamdaniego

$$m_{C1}(z) = w_1 \times m_{C1}(z). \quad (4)$$

Wniosek z obu reguł stanowi zbiór rozmyty C o funkcji przynależności:

$$m_C = \max[w_1 \times m_{C1}(z), w_2 \times m_{C1}(z)]. \quad (5)$$

Powyższą regułę nazwaną czasem max-dot zobrażowana na rysunku 12. Praktyka pokazuje, że ta metoda jest równie dobra jak wcześniej opisywana jednakże jest rzadziej wykorzystywana [8].



Rys. 12. Wnioskowanie według Larsena



### 4.3 Defuzyfikacja

W tym bloku, na podstawie wynikowej funkcji przynależności wyjścia  $\mu_{\text{wyn}}(y)$ , obliczana jest ostra wartość wyjścia będąca skutkiem podania ostrych wartości  $x_1, x_2$  na model. Operacja ta realizowana jest przez mechanizm defuzyfikacji, który definiuje metodę obliczeń. Istnieje dość liczny zbiór metod obliczeniowych, są to między innymi [4]:

- metoda środka maksimum,
- metoda pierwszego maksimum,
- metoda ostatniego maksimum,
- metoda środka ciężkości,
- metoda środka sum,
- metoda wysokości.

## 5. Regulator PID

Regulator PID jest obecnie najczęściej stosowaną strategią sterowania, spotykaną w układach regulacji automatycznej. Ich ogromną zaletą jest prosta i łatwości implementacji [10]. Regulator opisany jest zależnością:

$$G_R(s) = K_P \left( 1 + \frac{1}{sT_I} + sT_D \right) = K_P + \frac{K_I}{s} + sK_D, \quad (6)$$

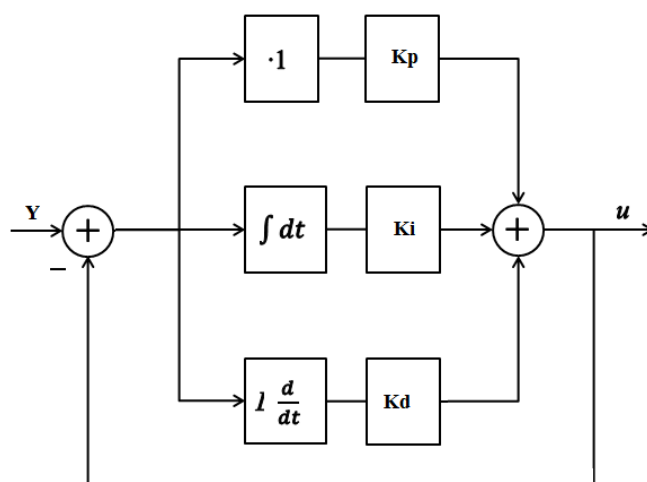
gdzie:

$K_P$  - współczynnik wzmocnienia regulatora,

$T_I$  - stała czasowa całkowania,

$T_D$  - stała czasowa różniczkowania.

Na rysunku 13 przedstawiona została graficzna interpretacja regulatora i jego działania.



Rys. 13. Schematyczne przedstawienie regulatora PID

### 4.1 Charakterystyki sterowań P, I, D

Sterowanie proporcjonalne z nastawą  $K_P$  ma wpływ na zmniejszanie czasu narastania i będzie zmniejszało uchyb w stanie ustalonym, lecz nigdy nie będzie go eliminowało. Sterowanie całkujące z nastawą  $K_I$  ma wpływ na eliminowanie uchybu w stanie ustalonym, lecz pogarsza odpowiedź w stanie przejściowym. Sterowanie różniczkujące z nastawą  $K_D$  ma wpływ na

zwiększenie stabilności układu, zmniejszając przeregulowanie i poprawiając odpowiedź przejściową [15].

## 4.2 Strojenie parametrów regulatora

Dla każdego procesu istnieje możliwość ścisłej analizy teoretycznej układów sterowania, która pozwala na dokładne wyznaczenie właściwości projektowanego układu, jednakże w praktyce często nie znamy wszystkich potrzebnych parametrów do obliczenia nastaw. W tym przypadku dobór nastaw regulatora polega na takim ustalaniu parametrów, aby otrzymana odpowiedź była satysfakcjonująca i mieściła się w granicy dozwolonego błędu [6].

- Pomiar własności dynamicznych obiektu regulacji metodą skoku jednostkowego.

Optymalne nastawy regulatorów opracowywane są zazwyczaj w postaci tabel, wykresów, wzorów. W takim przypadku obiekty dzieli się na dwie klasy: obiekty statyczne i astatyczne, których charakterystyki aproksymuje się następującymi transmitancjami :

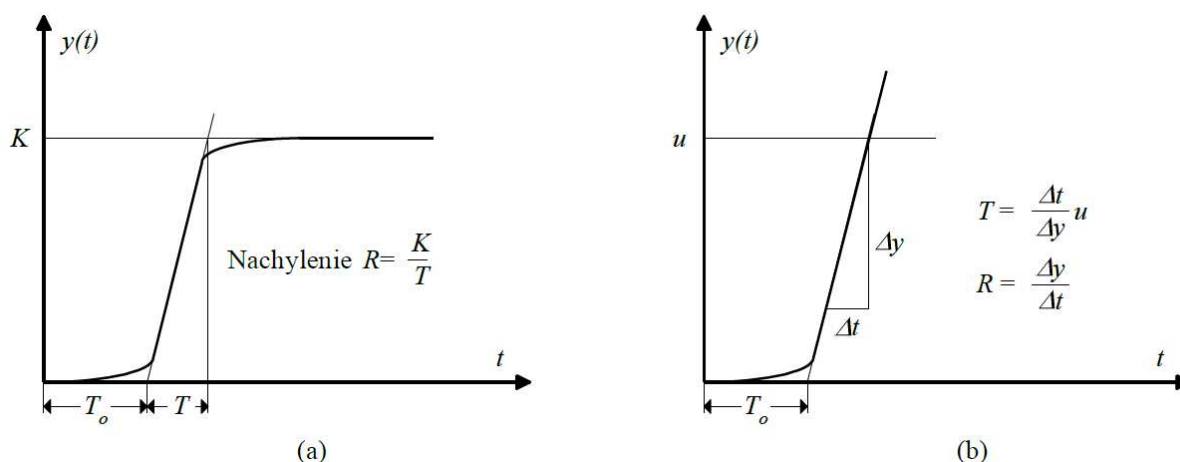
- dla obiektu statycznego:

$$G(s) = \frac{Ke^{-sT_0}}{Ts+1} \quad (7)$$

- dla obiektu astatycznego:

$$G(s) = \frac{Ke^{-sT_0}}{s} = \frac{e^{-sT_0}}{Ts}; K = \frac{1}{T} \quad (8)$$

Sposób wyznaczania parametrów modelu  $K$ ,  $T$ ,  $T_0$  odwzorowującego obiekt na podstawie odpowiedzi na skok jednostkowy przedstawiono na rysunku 14.



Rys. 14. Sposób określania parametrów na podstawie jednostkowej odpowiedzi procesu

(a) dla obiektu statycznego, (b) dla obiektu astatycznego

### 1. Metody doboru nastaw według Zieglera – Nicholasa.

Na bazie obserwacji eksperymentalnych strojeń regulatorów PID w procesach przemysłowych Ziegler i Nichols zaproponowali dwie metody doboru nastaw. Metody te wymagają pomiaru odpowiedzi układu i są powszechnie stosowane dla układów typu 0 najczęściej spotykanych w układach sterowania (np. sterowanie poziomem cieczy w zbiorniku, temperaturą, ciśnieniem, przepływem, itd.). Metody doboru nastaw opracowane przez Zieglera i Nicholasa opierają się na odpowiedzi skokowej i możliwie są do zastosowania w tych układach, w których odpowiedź skokowa układu otwartego jest bez przeregulowań tak jak ta pokazana na rysunku 14a. Ziegler i Nichols opracowali dwie metody strojenia regulatorów dla takich modeli [15].

- Metoda bazująca na odpowiedzi skokowej

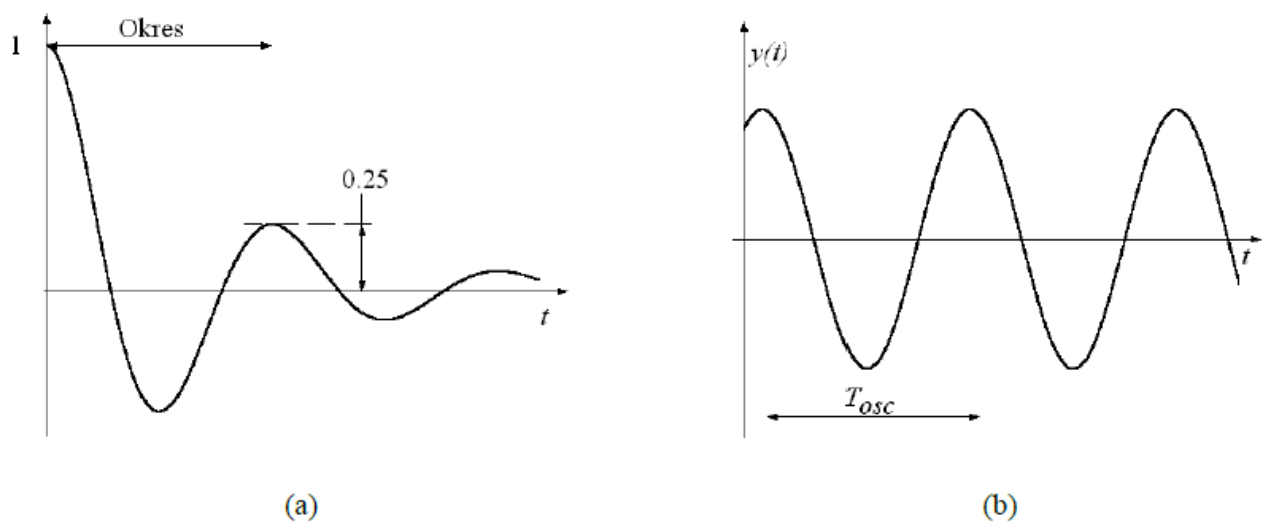
Autorzy zauważyli, że odpowiedź skokowa większości układów sterowania ma kształt podobny do tego z rysunku 14a. Krzywa może być otrzymana z danych uzyskanych poprzez symulację dynamiczną obiektu.

W tej metodzie wybór parametrów regulatora opiera się na współczynniku zanikania równym w przybliżeniu 0,25. Oznacza to, że dominująca składowa przejściowa zanika do jednej czwartej swojej wartości maksymalnej po jednym okresie oscylacji, co przedstawiono na rysunku 15a. Metoda ta daje dobre rezultaty, gdy spełniony jest warunek:

$$0.15 < \frac{T_0}{T} < 0.6 \quad (9)$$

- Metoda wzmocnienia krytycznego

W tej metodzie kryterium strojenia parametrów opiera się na ocenie układu znajdującego się na granicy stabilności. Ten algorytm strojenia może być stosowany dla układów typu 0 posiadających rząd  $>2$ . Może być również zastosowany do samostrojenia regulatorów PID. Wzmocnienie krytyczne otrzymuje się poprzez stopniowe zwiększanie wzmocnienia proporcjonalnego aż do zauważenia oscylacji o stałej amplitudzie. Zjawisko to przedstawiono na rysunku 15b [16].



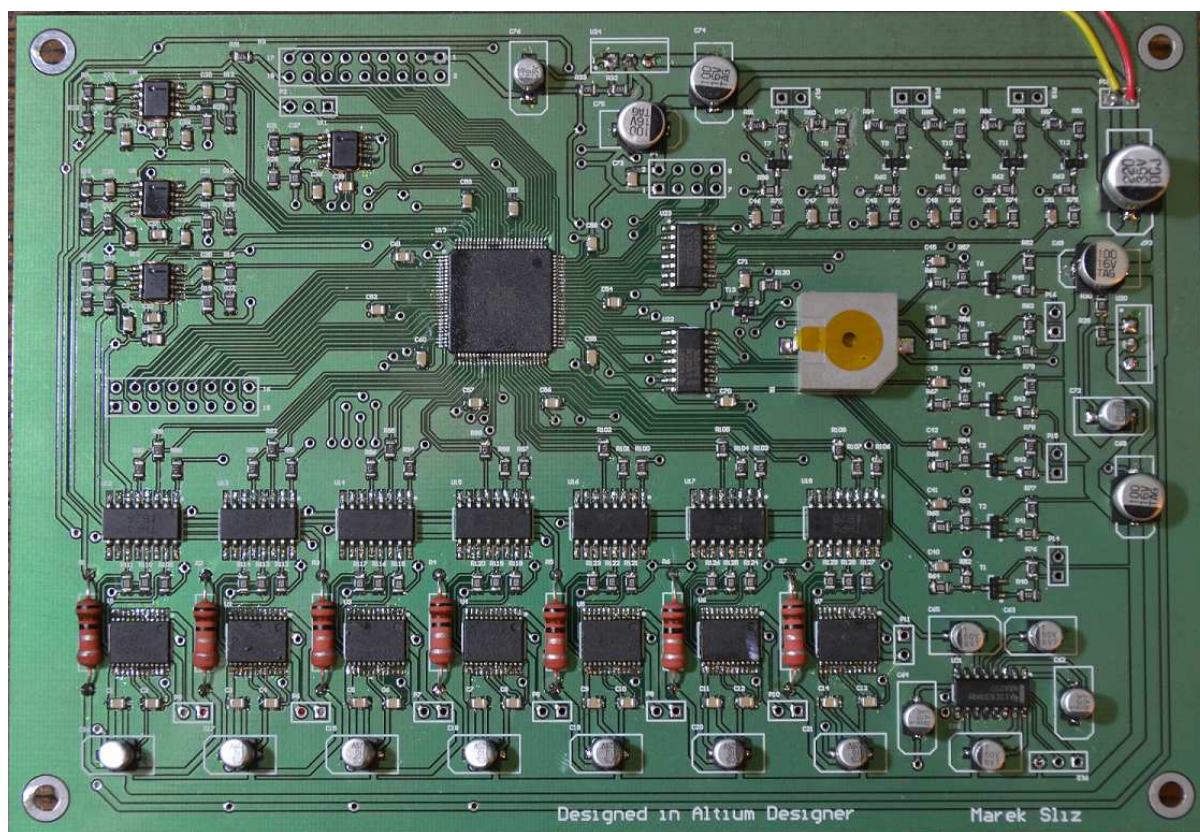
Rys. 15. Przebiegi czasowe (a) kwadratowy współczynnik zanikania, (b) układ na granicy stabilności [15]

## 6. Układ sterujący manipulatorem

Obecnie układem kontrolującym pracę manipulatora jest dedykowana płytki PCB, której zaprojektowanie oraz wykonanie było tematem pracy przejściowej. Wygląd układu został przedstawiony na rysunku 16.

### 5.1 Układ sterowania

Układem kontrolującym pracę manipulatora jest mikrokontroler ATxmega 128A1 taktowany wewnętrznym rezonatorem 32MHz. Mikrokontroler posiada 128 Kbajtów pamięci flash, osiem 16-sto – bitowych timerów, dwa 16-sto kanałowe przetworniki ADC o rozdzielczości 12 - bitów oraz wielopoziomowe przerwania.



Rys. 16. Widok zaprojektowanej płytki PCB

ATxmega to 8-bitowe mikrokontrolery z rdzeniem AVR. Od innych rodzin AVR odróżnia je bogactwo urządzeń peryferyjnych, niższe napięcie pracy, budowa zorientowana na oszczędności

energii oraz wiele specjalistycznych podsystemów znacznie przyspieszających wykonywanie programu. [12].

Wyjściem mocy są mostki H TB6612. Wymieniony układ pozwala na maksymalne zasilanie silnika do 15V. Maksymalna częstotliwość przełączania wynosi 100 kHz, a maksymalny prąd wyjściowy to 1,2A na jeden kanał. Doysterowania jednego mostka potrzebne są trzy sygnały PWM oraz dwa odpowiadające za kierunek. W celu zabezpieczenia mikrokontrolera przed wysokim napięciem wszystkie sygnały są izolowane galwanicznie transoptorami tranzystorowymi TLP281-4. Mostki H umożliwiają również pomiar prądu silników. W tym celu badany jest spadek napięcia na każdym z rezystorów połączonych szeregowo pomiędzy nóżkę GND mostka H oraz masę układu. Spadek napięcia jest proporcjonalny do prądu pobieranego przez silnik. Do jego wzmocnienia zastosowano cztery dwukanałowe wzmacniacze operacyjne LM358. Po wzmocnieniu sygnał trafia do wejścia ADC mikrokontrolera.

Jako sprzężenie zwrotne położenia wykorzystano optyczne enkodery inkrementalne o różnych rozdzielczościach zamontowane na wałach silników. Dążono do uzyskania minimalnej rozdzielczości wynoszącej 0,1%. Większa rozdzielczość nie miała uzasadnienia ze względu na luzy przekładni.

Ważnym aspektem jest również zabezpieczenie manipulatora przed uszkodzeniami. W tym celu badany jest prąd pobierany przez silnik. Po przekroczeniu zadanego poziomu sygnalizowany jest błąd. Dodatkowo przewidziano wyposażenie sterownika w przycisk bezpieczeństwa odcinającego zasilanie. Sygnalizowanie błędów odbywa się za pomocą sygnału dźwiękowego poprzez buzzer oraz za pomocą czerwonej diody LED. Układ umożliwia również sterowanie manipulatorem za pomocą podłączonych do sterownika klawiszy.

Cały układ został zamknięty w metalowej obudowie przedstawionej na rysunku 17. Na przednim panelu umieszczono między innymi złącze DB15, które może służyć do podłączenia dodatkowego osprzętu. Wyprowadza ono dziewięć nieużywanych pinów mikrokontrolera napięcia 3.3V, 5V, 12V oraz GND oraz jest złączem programatora.



Rys. 17. Sterownik manipulatora

Na górnej części obudowy znajdują się przyciski sterujące, przycisk bezpieczeństwa, wyświetlacz oraz diody LED sygnalizujące pracę. Na tylnym panelu umieszczono złącza służące do podłączenia manipulatora. Wymiary zewnętrzne sterownika manipulatora wynoszą 160x50x160 mm.

## 5.2 Niska warstwa oprogramowania

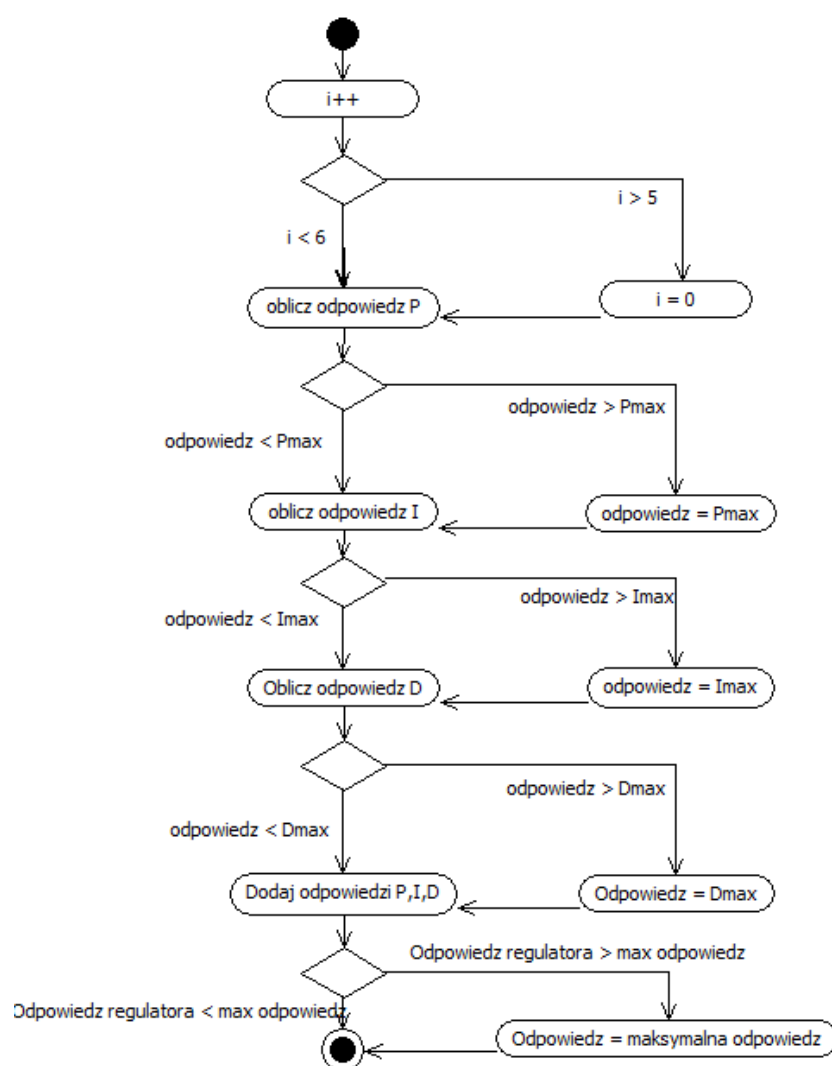
Program dla mikrokontrolera jest pisany w języku C w środowisku Eclipse. Wymienione środowisko cechuje się dużą uniwersalnością oraz możliwościami. Na mikrokontrolerze rozwiązywane są następujące zadania:

- obliczanie odpowiedzi regulatora PID,
- obliczanie aktualnej pozycji,
- monitorowanie pobieranego prądu,
- kalibracja manipulatora,
- wymiana danych z aplikacją sterującą.



W opisywanym manipulatorze każda z osi kontrolowana jest poprzez regulator PID. Częstotliwość sygnałów pochodzących z enkoderów wynosi około 1kHz. Działanie regulatora zostało schematycznie przedstawione na rysunku 18.

Przedstawiony algorytm wykonywany jest z częstotliwością 1.2 KHz, co daje częstotliwość 200 Hz odświeżania odpowiedzi regulatora dla każdego z napędów. Obliczona wartość zostaje zapisana do rejestru odpowiedzialnego za poziom wypełnienia sygnału PWM, który warunkuje prędkość obrotową napędu.



Rys. 18. Schemat działania regulatora PID

Regulator PID dzięki swojej prostocie nie wymaga skomplikowanych obliczeń, działania na liczbach zmiennoprzecinkowych są pominięte przez co nie wymaga on dużych zasobów obliczeniowych.

Wstępne wzmocnienia regulatorów dobrano zgodnie z ogólnymi wytycznymi przedstawionymi w literaturze [15]. Wykonano następujące kroki:

1. Wyznaczenie odpowiedzi układu regulacji z obiektem, ale bez regulatora i określenie co powinno zostać poprawione.
2. Dodanie sterowania proporcjonalnego w celu poprawienia czasu narastania i zmniejszenia uchybu w stanie ustalonym.
3. Dodanie sterowania różniczkującego w celu poprawienia przeregulowania.
4. Dodanie sterowania całkującego w celu wyeliminowania uchybu w stanie ustalonym.
5. Dostrojenie każdej z nastaw  $K_P$ ,  $K_I$ ,  $K_D$  aż do uzyskania braku przeregulowań.

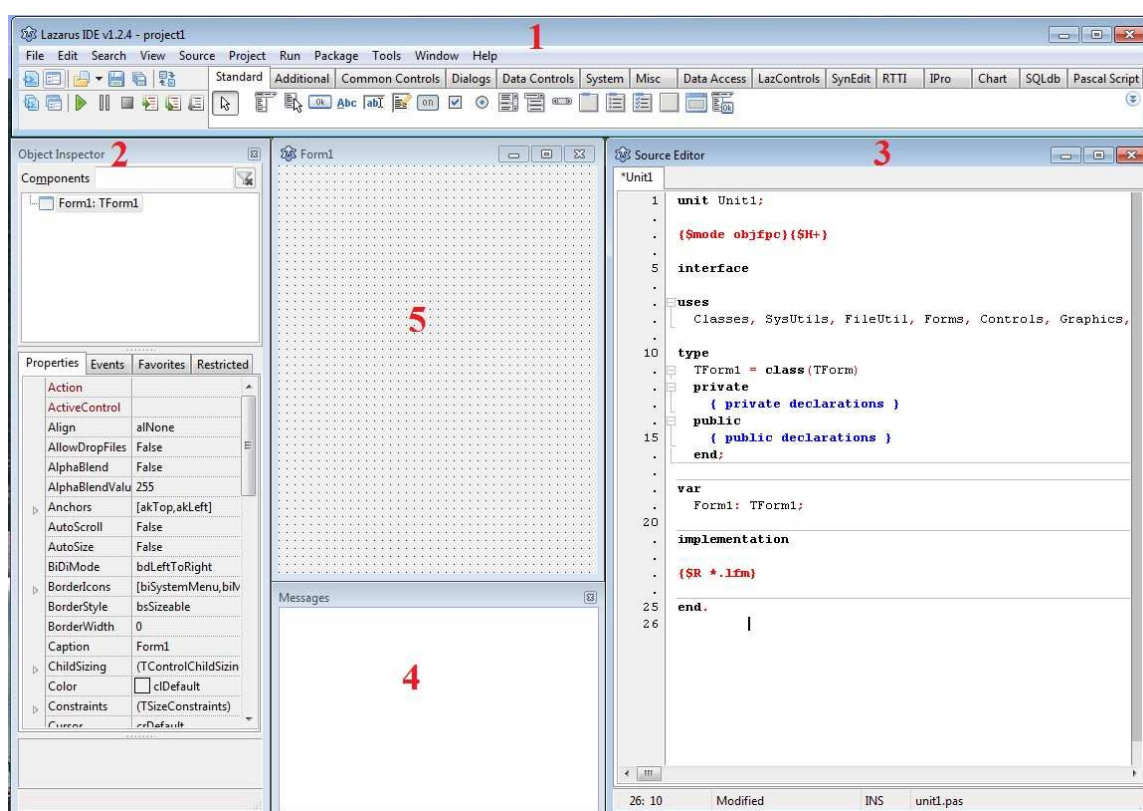
Dobre wzmocnienia regulatorów zostały przedstawione w tablicy 1.

Tablica 1. Wzmocnienia regulatorów

	Oś 1	Oś 2	Oś 3	Oś 4	Oś 5	Oś 6
$K_P$	55	55	60	55	25	19
$K_I$	1	2	2	1	3	3
$K_D$	80	120	110	115	85	119

## 7. Aplikacja sterująca

Do kontroli manipulatora powstała dedykowana aplikacja napisana w Lazarusie v1.2.4. Jest to w pełni darmowe środowisko na licencji GNU GPL, oparte jest o kompilator Free Pascal. Lazarus wzorowany jest na Delphi, wizualnym środowisku programistycznym [17]. Zaletą tego programu jest bardzo przejrzyste IDE (ang. *Integrated Development Environment*), które zostało przedstawione na rysunku 19. Aplikacje pisane w tym środowisku są zdarzeniowe to znaczy, że każdej kontrolce dostępnej w programie przypisanych jest co najmniej kilka możliwych zdarzeń po nastąpieniu, których wykonywana jest przypisana im procedura.

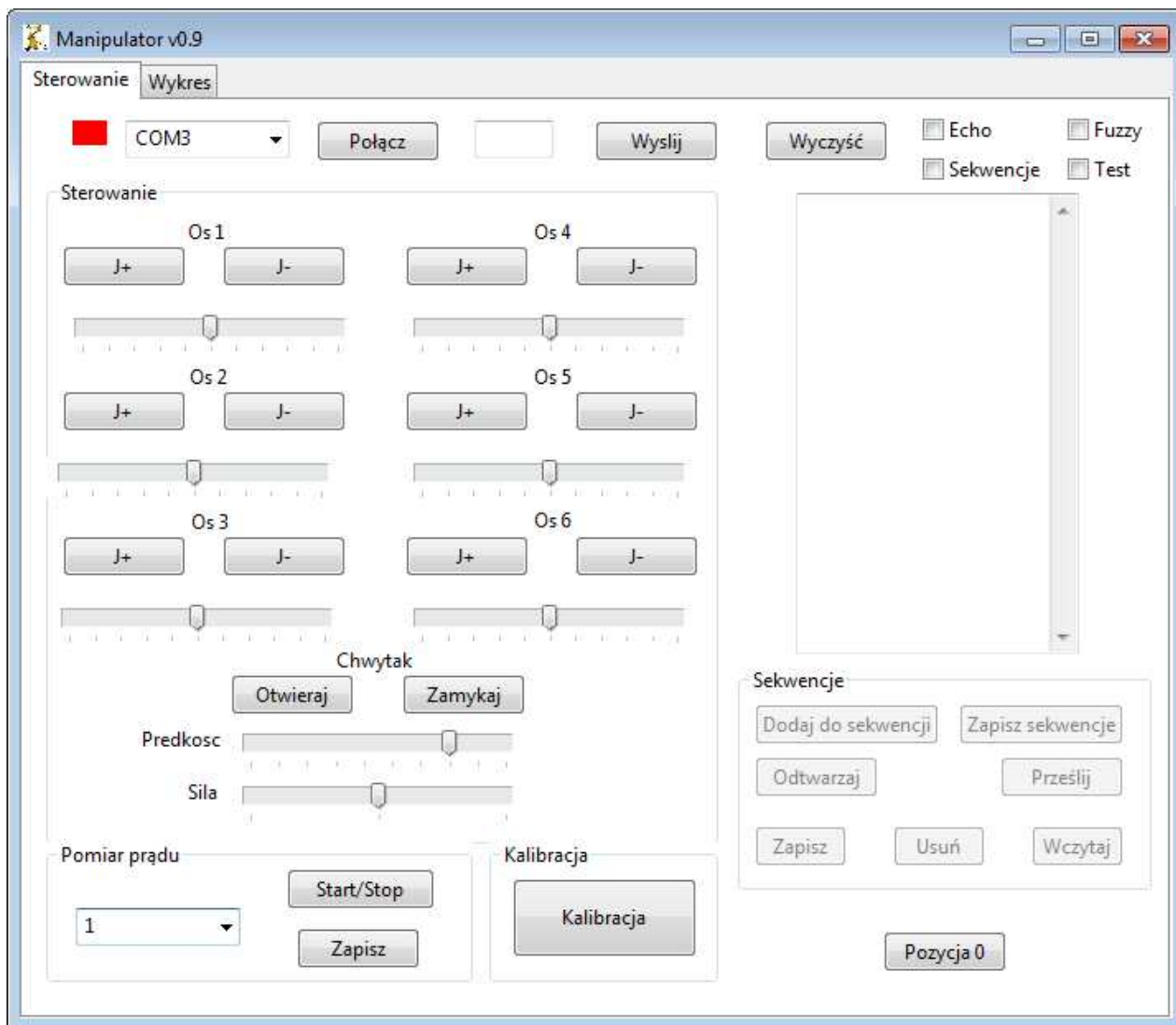


Rys. 19. Lazarus IDE

Okno główne składa się z:

- 1 - głównego panelu,
- 2 - inspektora obiektu
- 3 - edytora źródeł,
- 4 - okna wiadomości,
- 5 - formularza.

Napisana aplikacja uruchamiana jest z pliku .exe. Przy przenoszeniu jej na inny komputer nie jest wymagane instalowanie dodatkowego oprogramowania. Interfejs aplikacji przedstawiono na rysunkach 20 oraz 25.



Rys. 20. Aplikacja sterująca

W pierwszej zakładce mamy możliwość wybrania portu COM, do którego podłączony jest manipulator i utworzenia połączenia, wciskając przycisk „Połącz”. Kolorowy kwadracik z lewej strony jest informacją na temat połączenia, czerwony – brak połączenia, zielony – połączono. Ponadto widoczne jest pole tekstowe, do którego mamy możliwość wpisywania kodu rozkazu, który po wciśnięciu klawisza „Wyślij” zostanie wykonany przez manipulator. Taki mechanizm jest pomocny podczas diagnozowania nieprawidłowości w działaniu aplikacji. Przycisk „Wyczyść” służy do usuwania komunikatów wyświetlanych w poniższym oknie.

W prawym rogu dostępne mamy cztery pola wyboru opcji. Pierwsze „Echo” służy do podsłuchiwania przesyłanych rozkazów pomiędzy manipulatorem a aplikacją. Kolejne „Sekwencja” uaktywnia przyciski odpowiedzialne za tworzenie i odtwarzanie sekwencji. Działanie kolejnych pól wyboru „Fuzzy” oraz „Test” zostanie omówione w rozdziale 9. W sekcji „Sterowanie” mamy dostęp do kontrolowania wszystkich osi oraz chwytaka. Za pomocą suwaków ustawiana jest prędkość poruszania się poszczególnych ramion. Ponad to dla chwytaka przeznaczone są dwa suwaki. Suwak „Prędkość” ustawia prędkość działania chwytaka a suwak „Siła” reguluje siłę jego zamykania.

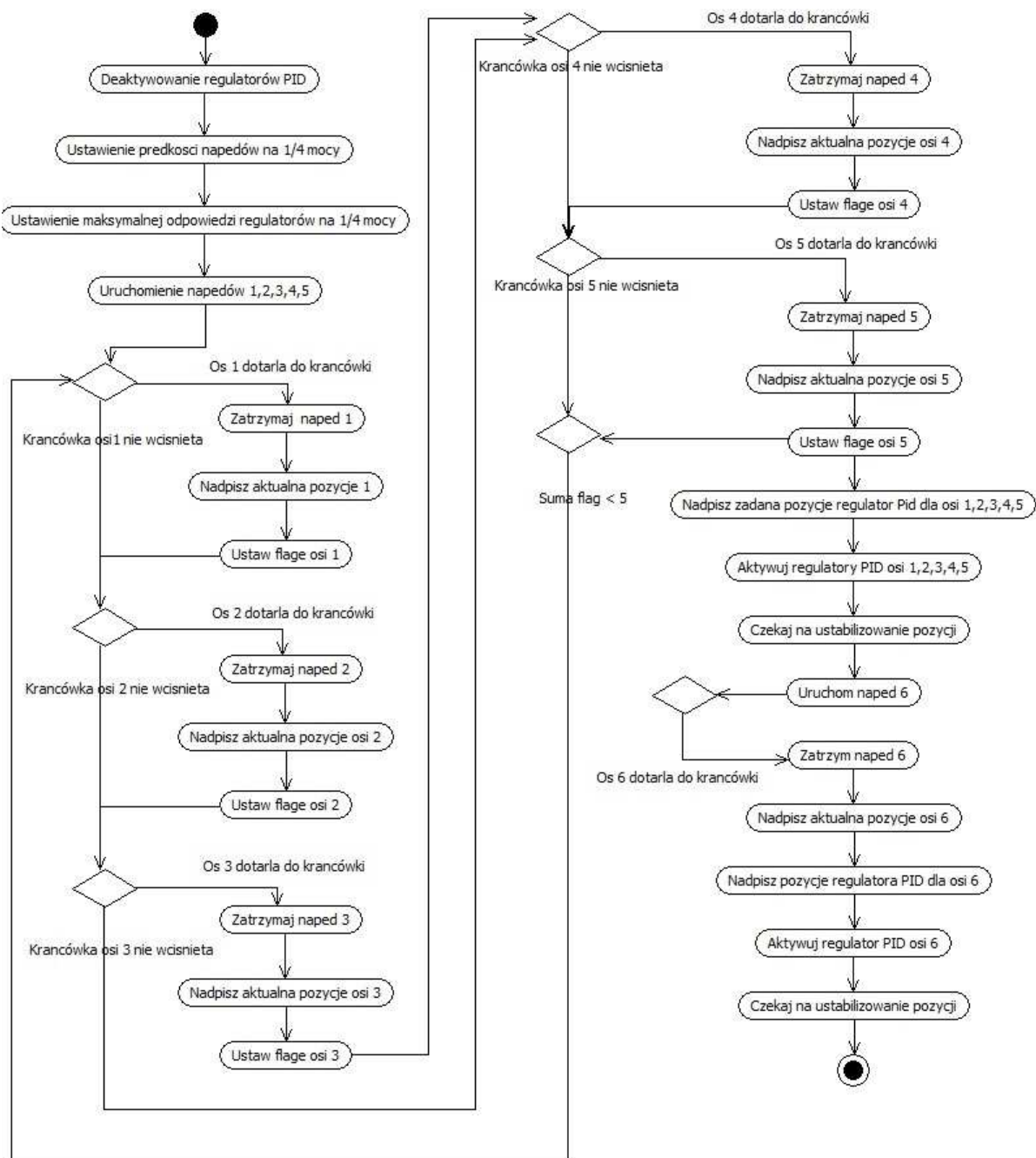
Poniżej w sekcji „Pomiar prądu” mamy możliwość pomiaru prądu na wybranym silniku. Aby dokonać pomiaru należy wybrać numer osi z listy rozwijanej, następnie rozpocząć pomiar naciskając na przycisk „Start/Stop”. W tym momencie przy pomocy przycisków z sekcji „Sterowanie” należy poruszyć osią. Pomiar jest wykonywany z częstotliwością 100 Hz. Przycisk „Start/Stop” zostanie wciśnięty aż do jego ponownego naciśnięcia, co spowoduje zakończenie pomiarów. Następnie należy przy pomocy przycisku „Zapisz” wskazać ścieżkę oraz nazwę pliku, w którym mają być zapisane dane z pomiarów.

W kolejnej sekcji „Kalibracja” znajduje się tylko jeden przycisk wysyłający do manipulatora rozkaz uruchomienia algorytmu kalibracji. Przed każdym jego uruchomieniem (jeśli to jest konieczne) należy ręcznie ustawić osie manipulatora w pozycji około „0”. Manipulator w tej pozycji został przedstawiony na rysunku 21. Wiąże się to z tym, że w pierwszej fazie działania algorytmu, regulatory PID są wyłączone i do napędów wpisana jest komenda obrotu w zadanym kierunku aż do natrafienia na krańcówkę. W przypadku gdyby oś była już za krańcówką i uruchomiony zostałby algorytm kalibracji doprowadziłoby to do uszkodzenia osi. Po każdym uruchomieniu manipulatora dla jego poprawnej pracy konieczna jest kalibracja pozycji. W manipulatorze każda z osi ma własną krańcówkę, przy czym krańcówki osi piątej i szóstej podłączone są do tego samego przewodu sygnałowego. Wadą tego rozwiązania jest brak możliwości kalibracji tych dwóch osi jednocześnie. Algorytm kalibracji przedstawiony jest na rysunku 22.



Rys. 21. Manipulator w pozycji „0”

Przycisk „Pozycja 0” zostaje aktywowany po wykonaniu kalibracji, wtedy po jego naciśnięciu manipulator wraca do pozycji „0”.

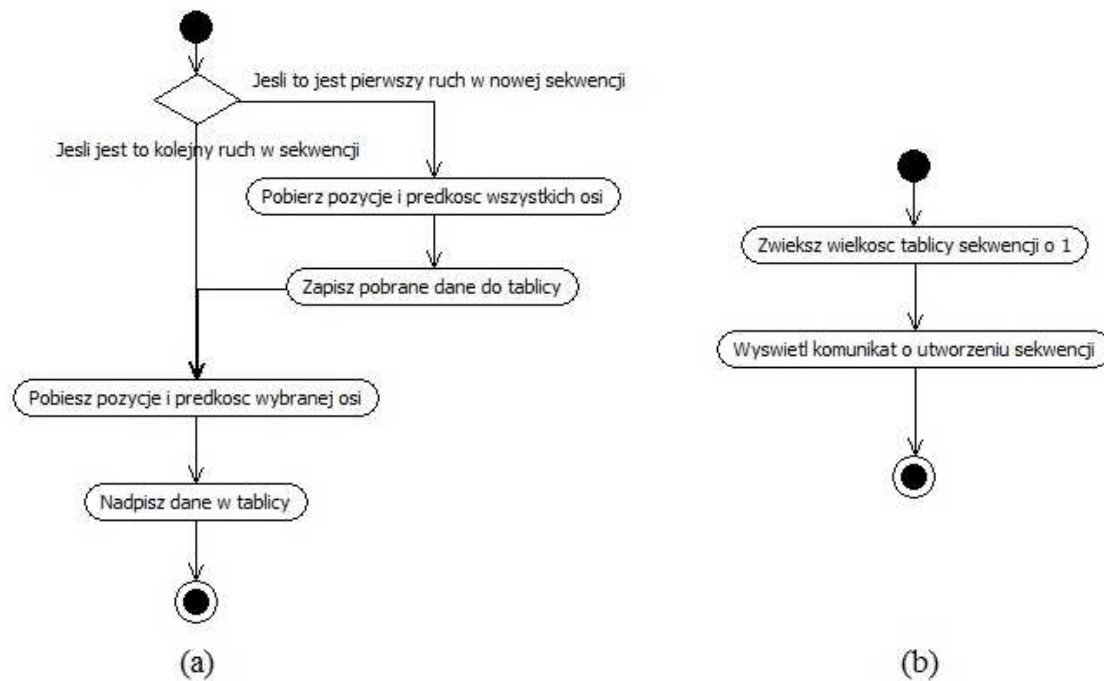


Rys. 22. Algorytm kalibracji

Po najeździe osi na krancówkę jej pozycja zostaje nadpisana, następnie przed uruchomieniem regulatorów „pozycje zadane” również zostają nadpisane. Ten zabieg ma na celu wprowadzenie uchybu regulacji tak, aby regulator jak najwcześniej przejął kontrolę nad napędem i przyjął pozycję „0”.

W sekcji „Sekuencje” mamy możliwość programowania sekwencji oraz kolejek. Pojęcie sekwencji oznacza tablicę z zapisaną prędkością poruszania się oraz docelowym położeniem dla wszystkich osi. Natomiast przez pojęcie kolejka rozumiany jest zbiór sekwencji odtwarzany

kolejno po sobie. W omawianej sekcji mamy do dyspozycji siedem przycisków. Pierwszy z nich „Dodaj do sekwencji”, zapisuje pozycje i prędkość sterowanej przed chwilą osi. Dodatkowo jeśli jest to pierwszy ruch w nowej sekwencji to algorytm pobiera aktualną pozycję i prędkość wszystkich ramion. Przycisk „Zapisz sekwencje” zwiększa wielkość tablicy sekwencji o jeden. Schematyczne działanie tych przycisków przedstawiono na rysunku 23.

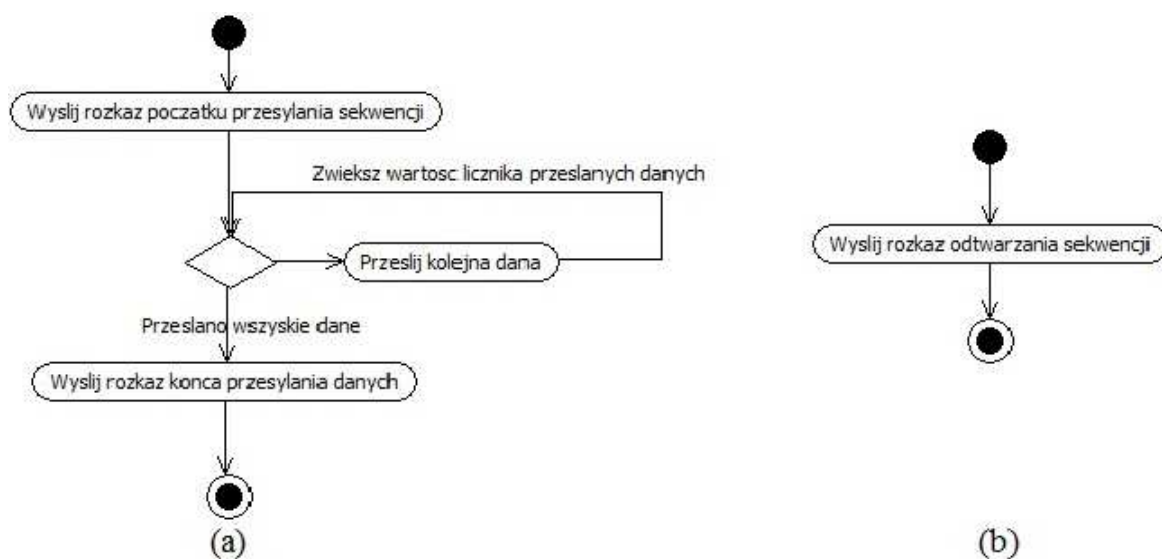


Rys. 23. Schemat działania przycisków, (a) „Dodaj do sekwencji”, (b) „Zapisz sekwencje”

Kolejny przycisk „Prześlij”, początkowo wysyła rozkaz informujący mikrokontroler o początku przesyłania sekwencji. Następnie zostają wysyłane wszystkie utworzone sekwencje. Koniec przesyłania tablicy również jest sygnalizowany rozkazem, co powoduje że mikrokontroler opuszcza pętle, w której odbiera dane. Procedura przycisku „Odtwarzaj” wysyła do manipulatora rozkaz rozpoczęcia odtwarzania przesłanych sekwencji. Schematyczne działanie tych przycisków przedstawiono na rysunku 24.

Algorytm zaimplementowany na mikrokontrolerze odczytuje wartości z przesłanej tablicy, ustawiając odpowiednio zadaną pozycję oraz prędkość poruszania się. Kolejno mikrokontroler oczekuje aż wszystkie osie się ustabilizują i przetwarza kolejną tablicę, aż do ostatniej sekwencji. Przyciski „Zapisz” i „Wczytaj” służą do zapisania aktualnej kolejki ruchów do pliku tekstowego lub jej wczytanie. W razie popełnienia błędu podczas tworzenia sekwencji istnieje możliwość jej usunięcia poprzez kliknięcie przycisku „Usuń”.

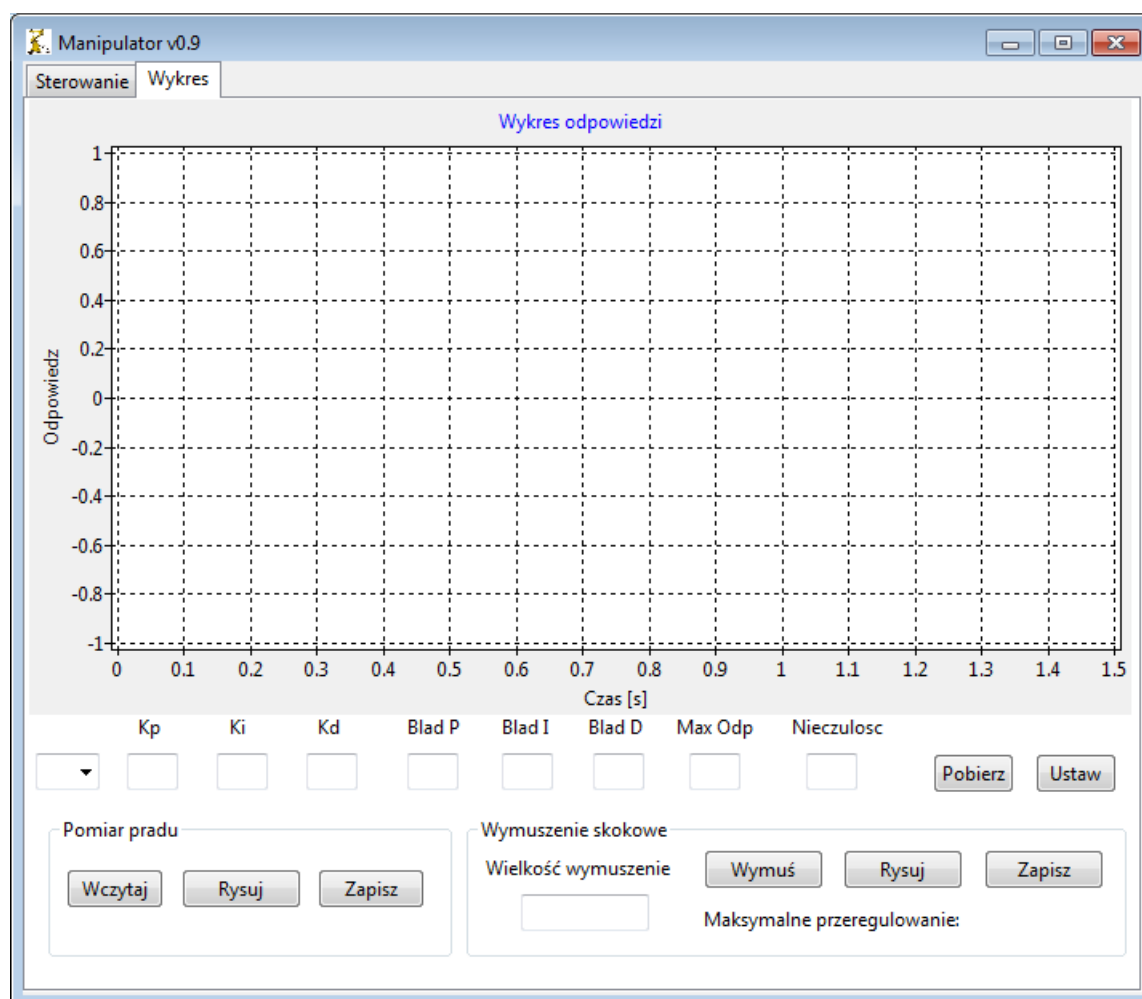




Rys. 24. Schemat działania przycisków (a) „Prześlij”, (b) „Odtwarzaj”

W drugiej zakładce aplikacji przedstawionej na rysunku 25, mamy możliwość rysowania odpowiedzi skokowej regulatora dla wszystkich osi. Aby narysować wykres należy wybrać jedną z osi z listy rozwijanej, następnie wpisać wielkość wymuszenia w pole „Wielkość wymuszenia” i wcisnąć przycisk „Wymuś” wykonane zostanie wtedy wymuszenie. Częstotliwość próbkowania wynosi podobnie jak dla pomiaru prądu 100Hz. Możliwe jest wpisanie w pole „Wielkość wymuszenia” wartości ujemnej spowoduje to ruch w przeciwnym kierunku a maksymalne zmierzone przeregulowanie zostanie wpisane za napisem „Maksymalne przeregulowanie:”. Klikając kolejno przycisk „Rysuj”, wygenerowany zostaje wykres odpowiedzi, skala wykresu dobierana jest automatycznie. Możliwe jest zapisanie wykresu do jednego z wybranych formatów graficznych poprzez wciśnięcie przycisku „Zapisz”.

W sekcji „Pomiar prądu” mamy możliwość wczytania wcześniej zapisanego pliku zawierającego dane z pomiaru prądu silnika, narysowania wykresu prądu oraz zapisania wykresu do pliku.



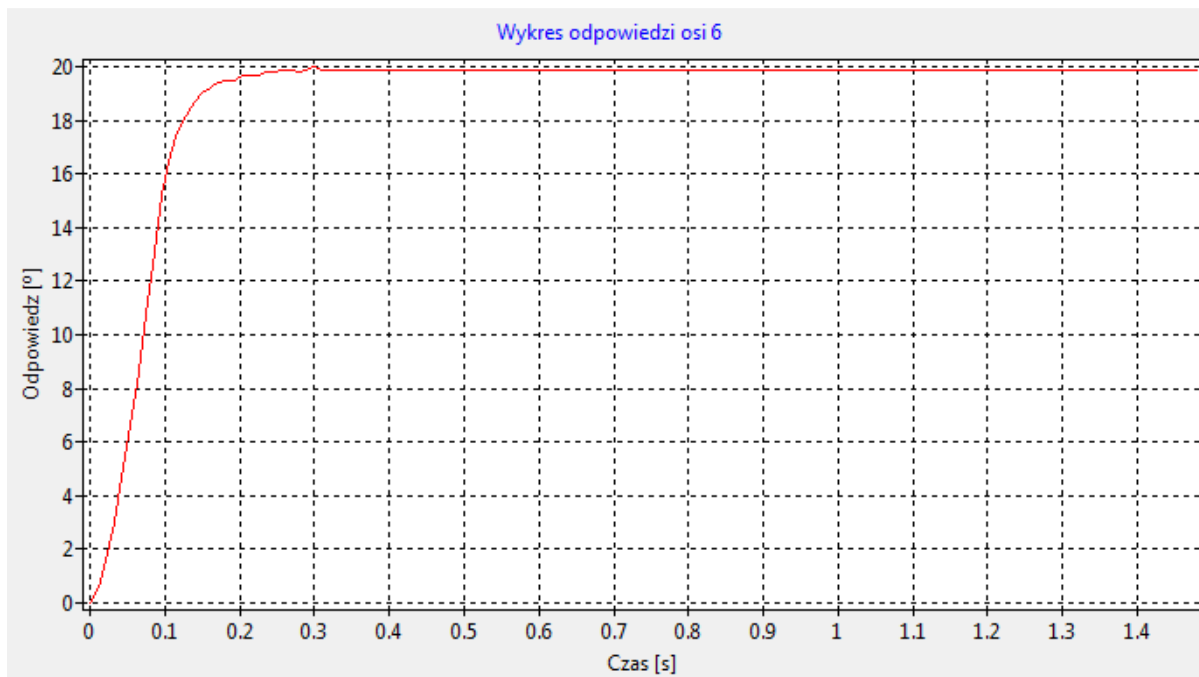
Rys. 25. Aplikacja sterująca

Aplikacja umożliwia nam dodatkowo podgląd aktualnie ustawionych parametrów regulatora PID oraz ich zmianę. Po wybraniu osi z rozwijanej listy i wciśnięciu przycisku „Pobierz” wypełnione zostaną pola tekstowe. Pierwsze trzy są to wzmocnienia części proporcjonalnej, całkującej oraz różniczkującej. Kolejnymi trzema są maksymalne wartości błędów jakie mogą być obliczone przez regulator dla jego kolejnych części. Do pola „Max Odp” wpisywana jest wartość maksymalnej odpowiedzi regulatora. Z kolei „Nieczułość” określa maksymalny uchyb, dla którego regulator pozostanie w spoczynku.

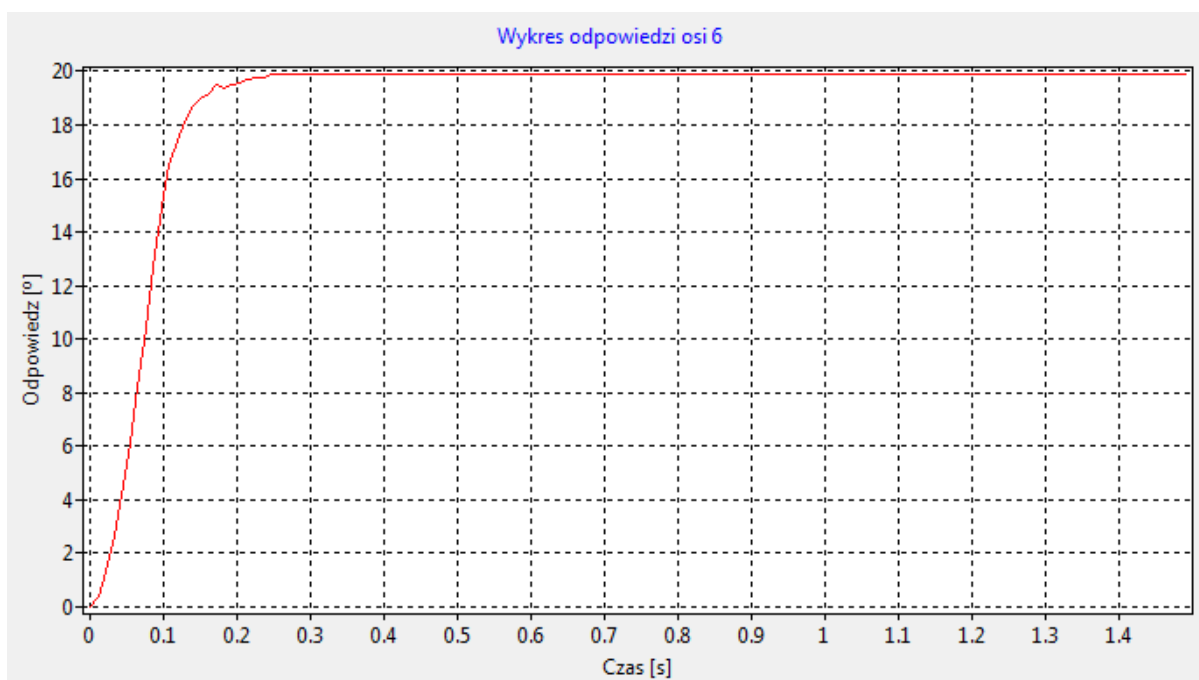
## 8. Projekt systemu sterowania

Podczas normalnej pracy, wartości sił działających na manipulator mogą się gwałtownie zmieniać. Na tego typu zmiany najbardziej narażona jest oś 2 manipulatora, na której wytwarza się największy moment. W takim przypadku istotną wadą regulatora PID jest brak sprzężenia zwrotnego informującego o wzajemnym położeniu osi względem siebie lub o momentach jakie obecnie działają na napędy. Taka sytuacja może skutkować przeregulowaniami (gdy moment jest dodatni) lub nieosiągnięciem przez napęd zadanej pozycji (gdy moment jest ujemny). Takie działanie manipulatora jest niedopuszczalne, ponieważ mogłoby doprowadzić do uszkodzenia manipulowanego przedmiotu lub samego manipulatora. Jednym z rozwiązań tego problemu byłaby zmiana sposobu regulacji z prędkościowej na momentową. Jednakże w tym przypadku dla celów badawczych postanowiono zbudować rozmyty regulator PID. Podobne działanie FLC przedstawione zostało w literaturze [10], z tą różnicą że w tym przypadku część rozmyta będzie generowała nowe wzmocnienia dla poszczególnych członów. W takim przypadku zasada działania regulatora PID nie uległa zmianie.

Początkowo rozważano koncepcję budowy regulatorów rozmytych na mikrokontrolerze. Jednakże po wstępnej analizie odrzucono ten pomysł ze względu na ilość obliczeń i działań na liczbach zmiennoprzecinkowych jakie musiałby wykonać mikrokontroler. Postanowiono wykonywać obliczenia związane z logiką rozmytą na komputerze PC i przysyłać wyniki w postaci wzmocnień do mikrokontrolera. Modele rozmyte zostały zbudowane w programie Matlab w pakiecie FLT. Udostępnia on wygodną bibliotekę do tworzenia logiki rozmytej wraz z graficznym interfejsem. Zdecydowano że regulatory rozmyte będą stosowane dla wszystkich osi poza ostatnią. Oś szósta jest wyposażona w silnik z największą przekładnią 1:298 oraz nie jest narażona na duże zmiany momentu bezwładności, dzięki czemu parametry tego regulatora mogą być stałe. W celu weryfikacji tych wzmocnień skorzystano z napisanej aplikacji i przeprowadzono kilka wymuszeń skokowych, wybrane przedstawiono na rysunkach 26 oraz 27.

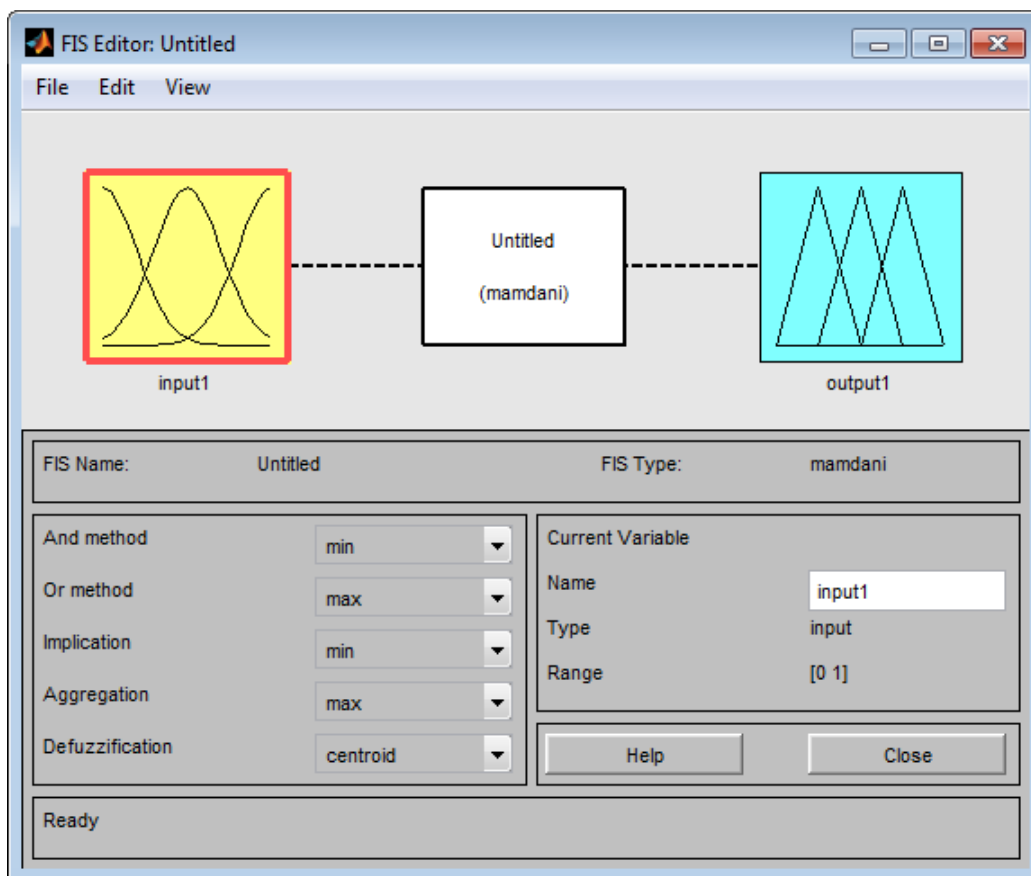


Rys. 26. Wykres odpowiedzi dla osi 6



Rys. 27. Wykres odpowiedzi dla osi 6

Oś szóstą cechuje się dużą stabilnością, maksymalne zarejestrowane przeregulowanie wynosi 0,1°.



Rys. 28. Okno główne FLT

FLT zapewnia możliwość tworzenia i edytowania logiki rozmytej w środowisku Matlab, integrację z Simulinkiem lub nawet generowania kodu w języku C. Dużym atutem jest możliwość obsługi LR (logiki rozmytej) za pomocą graficznego interfejsu (GUI graphical user interface), który w znaczny sposób ułatwia konfigurację i przejrzystość budowanych modeli LR. Moduł zapewnia trzy kategorie narzędzi i możliwości dostępu [14]:

- dostęp do funkcji poprzez linię komend,
- interaktywne graficzne narzędzia,
- bloki i przykłady dla Simulinka.

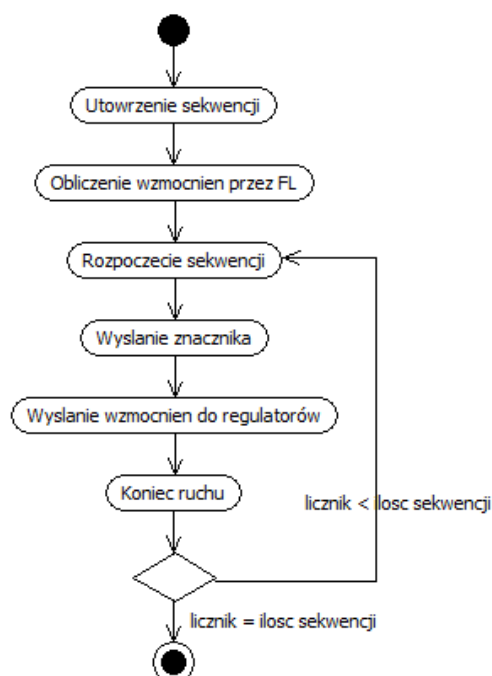
Schemat ideowy działania algorytmu przedstawiono na rysunku 29. Projektowane modele rozmyte obliczane będą wykorzystując metodę Mamdaniego. Każdy z modeli posiada trzy wyjścia nazwane odpowiednio  $K_P$ ,  $K_I$ ,  $K_D$  opisane funkcją trójkątną. Modele różnią się ilością wejść, jest to uwarunkowane wpływem działania ramion manipulatora na siebie. Wyróżniono trzy zmienne lingwistyczne dla wejść „pozycja osi”:

- pozycja ujemna (U),
- pozycja zero (Z),

- pozycja dodatnia (D).

Dla wejścia „kierunek” wyznaczono dwie zmienne lingwistyczne:

- góra (G),
- dół (D).



Rys. 29. Projekt algorytmu

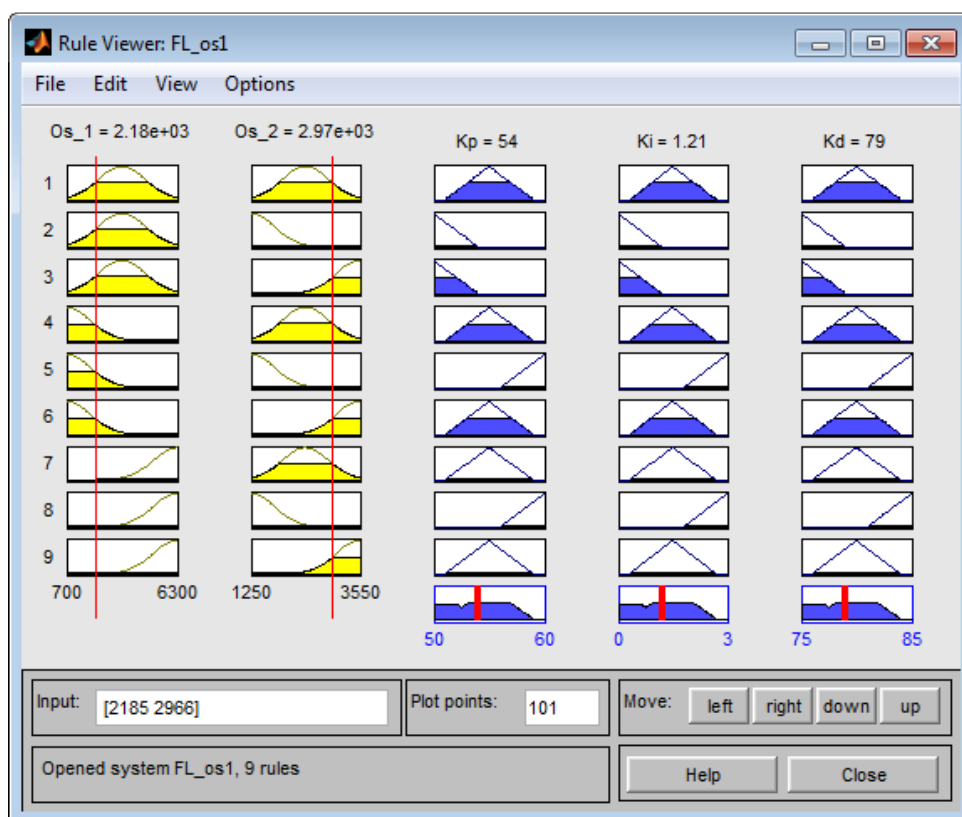
Taki podział zmiennych lingwistycznych miał na celu ograniczenie rozmiaru bazy reguł. Gdyby podział zmiennych lingwistycznych pozycji osi zawierał pięć zmiennych łączna ilość reguł dla wszystkich modeli wynosiłaby 2575 reguł. Taka ilość reguł byłaby zbyt pracochłonna do wprowadzenia, jak również znacznie wydłużyłaby czas obliczania odpowiedzi, dlatego zaproponowano podział na trzy zmienne, wówczas ilość wszystkich reguł wynosi 369. Natomiast każdą z trzech zmiennych wyjściowych podzielono na trzy zmienne lingwistyczne:

- mała wartość współczynnika (M),
- wartość współczynnika bez zmian (B),
- duża wartość współczynnika (D).

Funkcje przynależności posiadają kształt trójkątny.

Model logiki rozmytej pierwszej osi posiada dwa wejścia. Pierwsze wejście jest położeniem osi pierwszej, zakres wejściowy pokrywa się z maksymalnym zakresem ruchu tej osi. Drugim wejściem jest położenie osi drugiej, która ma największy wpływ na moment bezwładności tej osi.

Reszta ramion została pominięta z powodu nikłego wpływu na oś pierwszą. Na rysunku 30 przedstawiono przykładowy wynik obliczeń logiki rozmytej dla tej osi w FLT.



Rys. 30. Symulacja modelu FL dla osi 1

Model logiki rozmytej dla osi drugiej i trzeciej posiada trzy wejścia. Pierwsze z nich jest położeniem osi drugiej, następne to położenie osi trzeciej, a ostatnim jest kierunek poruszania się. W tych modelach jak i następnych konieczne jest uwzględnienie kierunku poruszania się ramienia ponieważ konieczne jest uwzględnienie momentu powstającego od sił grawitacji.

Następne modele osi czwartej i piątej, posiadają dodatkowe dwa wejścia, są to położenia osi czwartej oraz piątej.

Do obsługi każdego z modelu napisany został w Matlabie krótki skrypt. Jego zadaniem jest wprowadzenie danych do obliczeń, ich uruchomienie oraz zaokrąglenie wyniku do liczb całkowitych. Takie rozwiązanie upraszcza uruchomienie logiki rozmytej z poziomu aplikacji sterującej.

## 9. Implementacja systemu

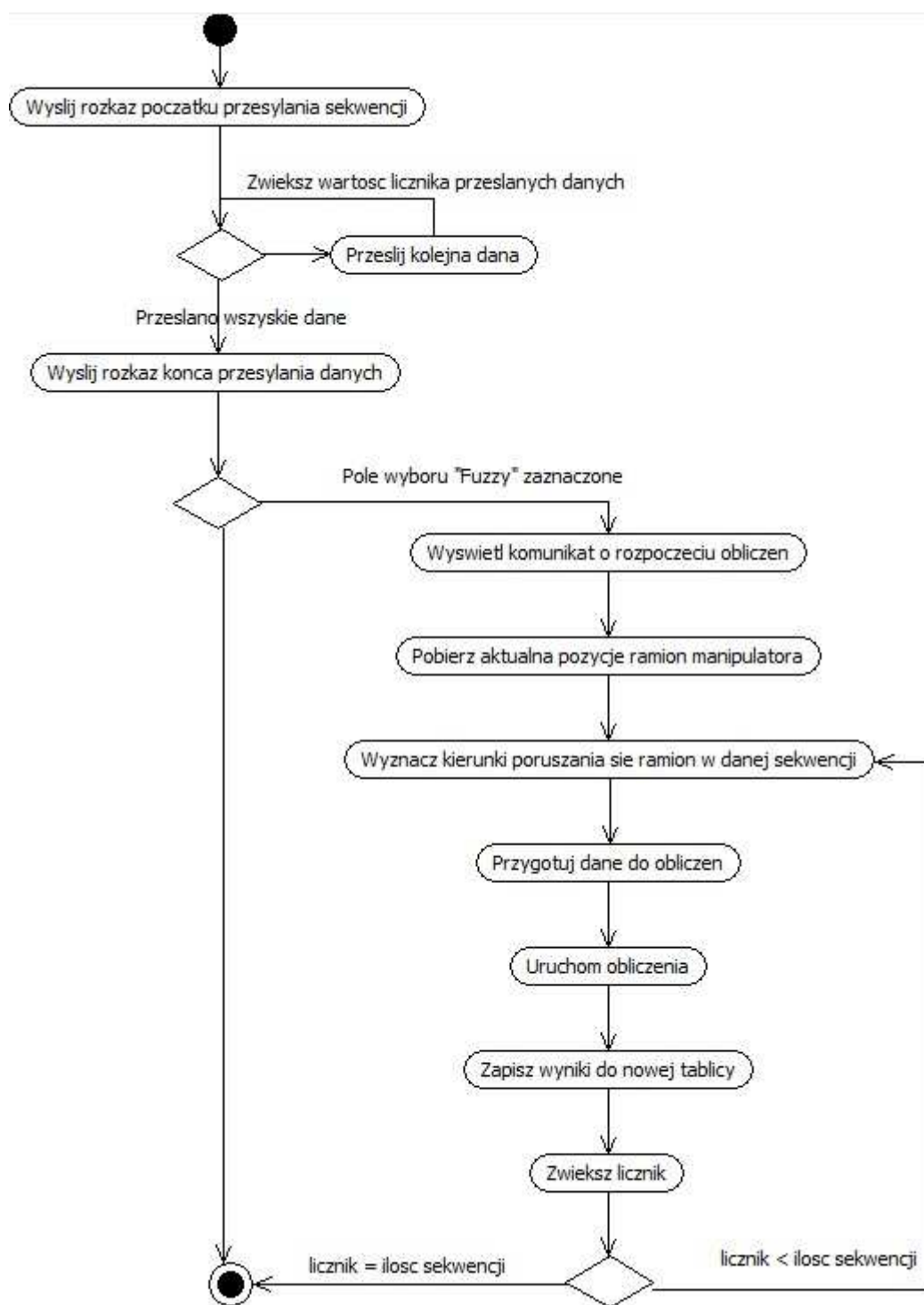
Komunikację pomiędzy aplikacją sterującą manipulatorem a Matlabem rozwiązano za pomocą komponentu OLE. Mechanizm OLE (ang. *Object Linking nad Embedding*), czyli łączenie i osadzanie obiektów, umożliwia przenoszenie obiektów pomiędzy aplikacjami z jednoczesnym zachowaniem połączenia pomiędzy obiektami a ich aplikacjami macierzystymi. Metoda wykorzystywania obiektów istniejących na zewnątrz aplikacji przy użyciu technologii OLE, polegająca na obsłudze tych obiektów zgodnie z zasadami programowania obiektowego, określana jest często jako automatyzacja OLE. Mechanizm ten jest oparty na modelu COM (ang. *Component Object Model*) stanowiącym interfejs programistyczny firmy Microsoft. W środowisku Lazarus można korzystać z tej technologii za pomocą odpowiednich komponentów. Wadą takiego rozwiązania jest konieczność instalowania potrzebnych programów na komputerze. Technika OLE pozwala również na korzystanie z programów innych producentów Adobe, OpenOffice, MS Office, Autocad itd. Takie rozwiązanie pozwala na łatwą wymianę danych pomiędzy aplikacją sterującą a Matlabem [6]. Na rysunku 31 przedstawiono procedurę uruchamiającą obliczenia LR dla jednej z osi.

```
begin
  FL := format('OS1 = FL_os1(%s,%s);',[inttostr(FL_input[i][0]),inttostr(FL_input[i][1])]);
  Matlab.Execute(FL);
  Matlab.Visible := True;
  MReal := VarArrayCreate([0,2], varDouble); //create array [0...x]
  MImage:= VarArrayCreate([0,0], varDouble); //create array [x]
  Matlab.GetFullMatrix('OS1','base',VarArrayRef(MReal),VarArrayRef(MImage));
  for l:=0 to 2 do
    begin
      FL_output[i][l] := strtoint(MReal[l]);
    end;
  VarClear(MReal);
  VarClear(MImage);
end;
```

Rys. 31. Procedura uruchamiająca obliczenia FL

Testy wykazały, że każdorazowe obliczenie wzmocnień regulatorów zajmuje około 1 sekundy. Niestety przy tak długim czasie obliczeń nie możliwa jest jego ciągła praca, podczas sterowania za pomocą przycisków w sekcji „Sterowanie”, dlatego postanowiono zastosować opisywany regulator podczas sterowania kolejkowego. W tym celu konieczne było zmodyfikowanie działania algorytmu tworzenia sekwencji. Zmiany dokonano w procedurze przycisku „Prześlij”. Schemat działania został przedstawiony na rysunku 32.



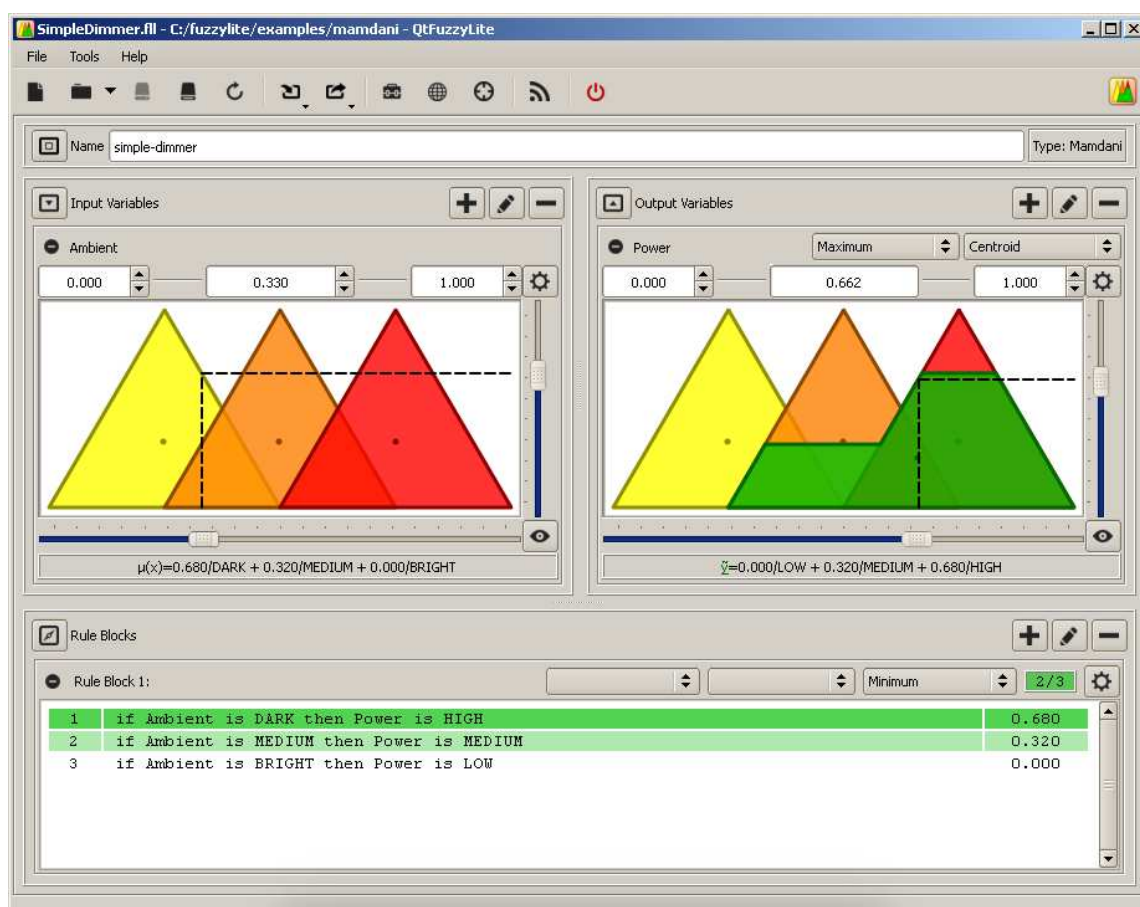


Rys. 32. Schemat tworzenia sekwencji

Po zaznaczeniu pola wyboru „Fuzzy” utworzony zostaje obiekt OLE o nazwie Matlab oraz uruchomiona zostaje jego aplikacja konsolowa. Dalsza część algorytmu związanego z logiką rozmytą wykonuje się po wciśnięciu przycisku „Przeslij”. Wówczas po przesłaniu wszystkich sekwencji pojawia się komunikat o rozpoczęciu obliczeń logiki rozmytej. Dla pierwszej sekwencji

pobierane jest aktualne położenie manipulatora. Jest to konieczne, aby wyznaczyć kierunek poruszania się osi w pierwszej wykonywanej sekwencji co jest jedną z danych wejściowych do modeli logiki rozmytej. Następnie w odpowiedniej kolejności przesyłane są pozycje z tablicy sekwencji do Matlaba i uruchamiane są obliczenia dla danej sekwencji. Jej wynik zapisywany jest do tablicy. Po wykonaniu obliczeń zostaje wyświetlona informacja o przebiegu obliczeń. Następnie po wciśnięciu przycisku „Odtwarzaj”, manipulator wysyła znacznik rozpoczęcia sekwencji i zaczyna się poruszać. Aplikacja po odebraniu znacznika rozpoczyna przesyłanie pierwszego zestawu obliczonych wzmocnień dla regulatorów PID. Wzmocnienia zostają wpisane zanim zakończy się ruch w danej sekwencji. Takie działanie algorytmu zapewnia właściwe wzmocnienia regulatora przy dojeżdżaniu do pozycji, co skutkuje zmniejszeniem przeregulowań napędu.

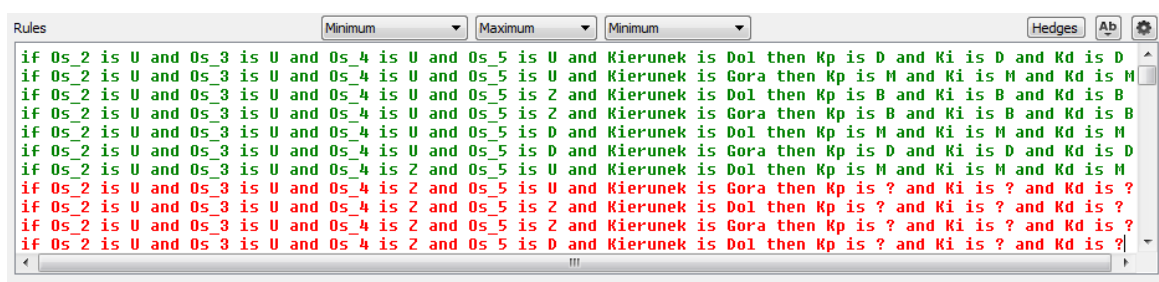
Baza reguł dla wszystkich modeli logiki rozmytej powstała na podstawie wiedzy eksperta oraz licznych doświadczeń przeprowadzonych na manipulatorze. Była to najbardziej czasochłonna część pracy. W celu przyspieszenia implementacji bazy reguł korzystano z programu Fuzzylite 4.0.



Rys. 33. Okno główne programu Fuzzylite 4.0

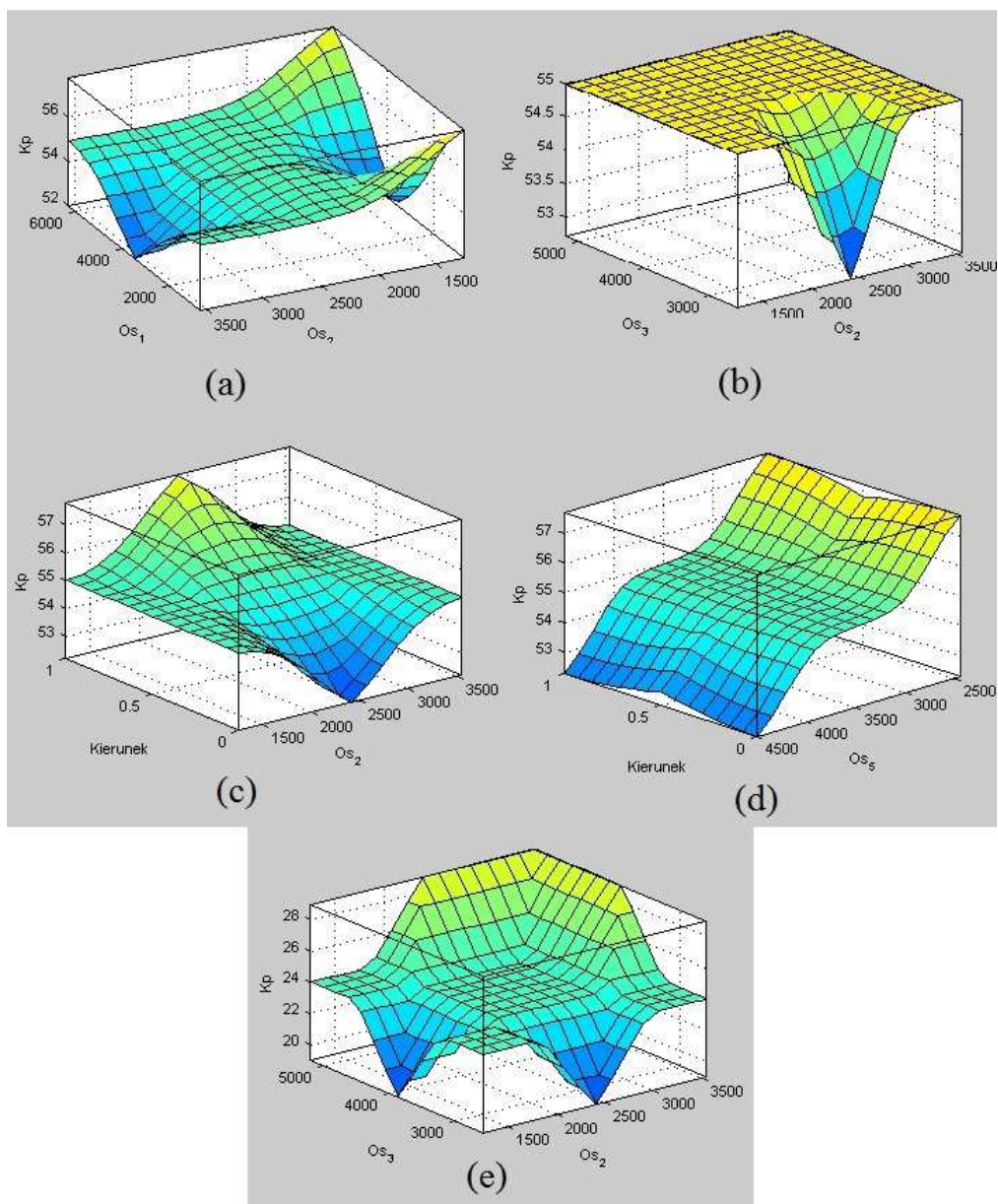
Fuzzylite jest to darmowy program napisany w języku C++ działający na różnych platformach (Windows, Linux, Mac, iOS). Pozwala on na łatwe tworzenie modeli logiki rozmytej w szybki i wygodny sposób. Sam program nie wymaga instalowania oraz nie korzysta z innych programów zewnętrznych. Fuzzylite podobnie jak FLT posiada interfejs graficzny, który ułatwia jego obsługę. Na rysunku 33 przedstawiono widok głównego okna programu [18].

Ogromną zaletą tego programu jest możliwość automatycznego wygenerowania kombinacji reguł. Do tak przygotowanej bazy należy jedynie wpisać odpowiednie wartości wyjścia. Częściowo wypełnioną bazę reguł w programie Fuzzylite przedstawiono na rysunku 34.



Rys. 34. Przykład wygenerowanych reguł

Po zdefiniowaniu bazy reguł zaimportowano modele do FLT, gdzie sprawdzono jej poprawność oraz wygenerowane zostały powierzchnie reprezentujące dane wejściowe i wyjściowe. Przykładowe powierzchnie zostały przedstawione na rysunku 35.



Rys. 35. Powierzchnie bazy reguł dla osi 1 (a), dla osi 2 (b), dla osi 3 (c), dla osi 4 (d), dla osi 5 (e)

## 10. Weryfikacja działania systemu

Po implementacji systemu przeprowadzono weryfikację jego działania. Polega ona na przeprowadzeniu dwóch kolejek ruchowych po trzy i dwie sekwencje a następnie analizie wymuszeń skokowych dla każdej z osi. Starano się, aby pozycje które wybrano były różnorodne i skrajne. Pozycje przedstawiono na rysunku 37.

Wszystkie człony manipulatora zostały ustawione w pozycji, w której znajdowały się podczas odtwarzanej sekwencji. Pozycja osi która jest aktualnie badana pomniejszona jest o wielkość wymuszenia skokowego. Następnym krokiem jest wpisanie wcześniej obliczonych wartości wzmocnień regulatora dla danego położenia i przeprowadzenie wymuszenia skokowego. Wielkość wymuszenia jest umowna i wynosi dla każdej z osi 200 kroków. W tablicy 2 przedstawiono wielkości wymuszeń w przeliczeniu na stopnie.

Tablica 2. Wielkość wymuszenia

	Przełożenie	Ilość otworów w tarczy enkodera	Wielkość wymuszenia	Wielkość wymuszenia w [°]
Oś 1	1:280	20	200	12,9
Oś 2	1:280	20	200	12,9
Oś 3	1:280	20	200	12,9
Oś 4	1:160	32	200	14
Oś 5	1:180	20	200	20
Oś 6	1:298	12	200	20

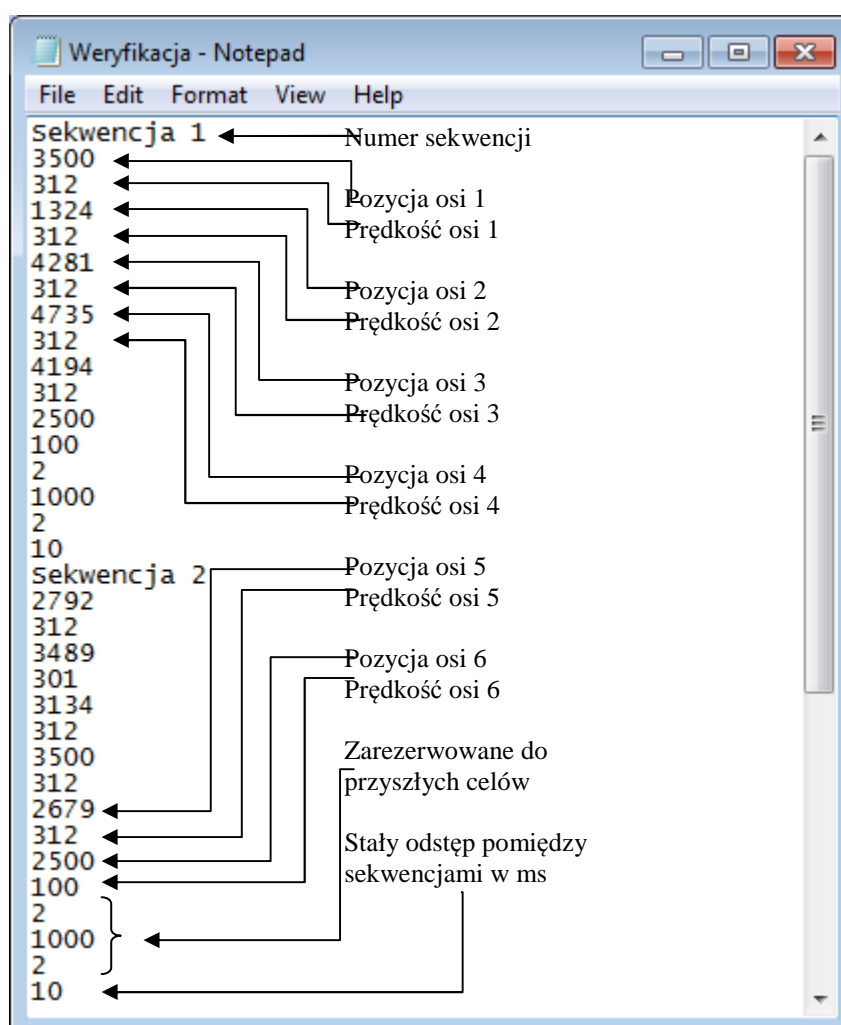
Taka wartość skoku zapewnia możliwość rozwinięcia pełnej mocy przez silnik. Następnie wymuszenia zostaną przeanalizowane na wykresie, dopuszczalny jest błąd z zakresu 1%. Wielkość błędu wynika z konstrukcji manipulatora, który w głównej mierze powstał poprzez ręczne wytwarzanie części, stąd mogące się wkradać niedokładności.

Aby aplikacja umożliwiała wyświetlanie obliczonych wartości wzmocnień należy zaznaczyć pole wyboru „Test”, wówczas wyświetlone zostaną obliczone wzmocnienia, które są aktualnie przesyłane do manipulatora. Z kolei dokładna pozycja ramion manipulatora oraz wszystkie parametry sekwencji dostępne są w pliku po zapisaniu sekwencji na dysk. Struktura pliku przedstawiona jest na rysunku 36.

Pozycja, w której musi się znajdować ramie manipulatora zostanie osiągnięta poprzez wpisanie odpowiedniej wartości w pole tekstowe, które umożliwia wysyłanie bezpośrednich rozkazów. Zasady wpisywania wartości do pola tekstowego, aby oś osiągnęła daną pozycję przedstawiono w tablicy 3.

Tablica 3. Zasada wysyłania danych

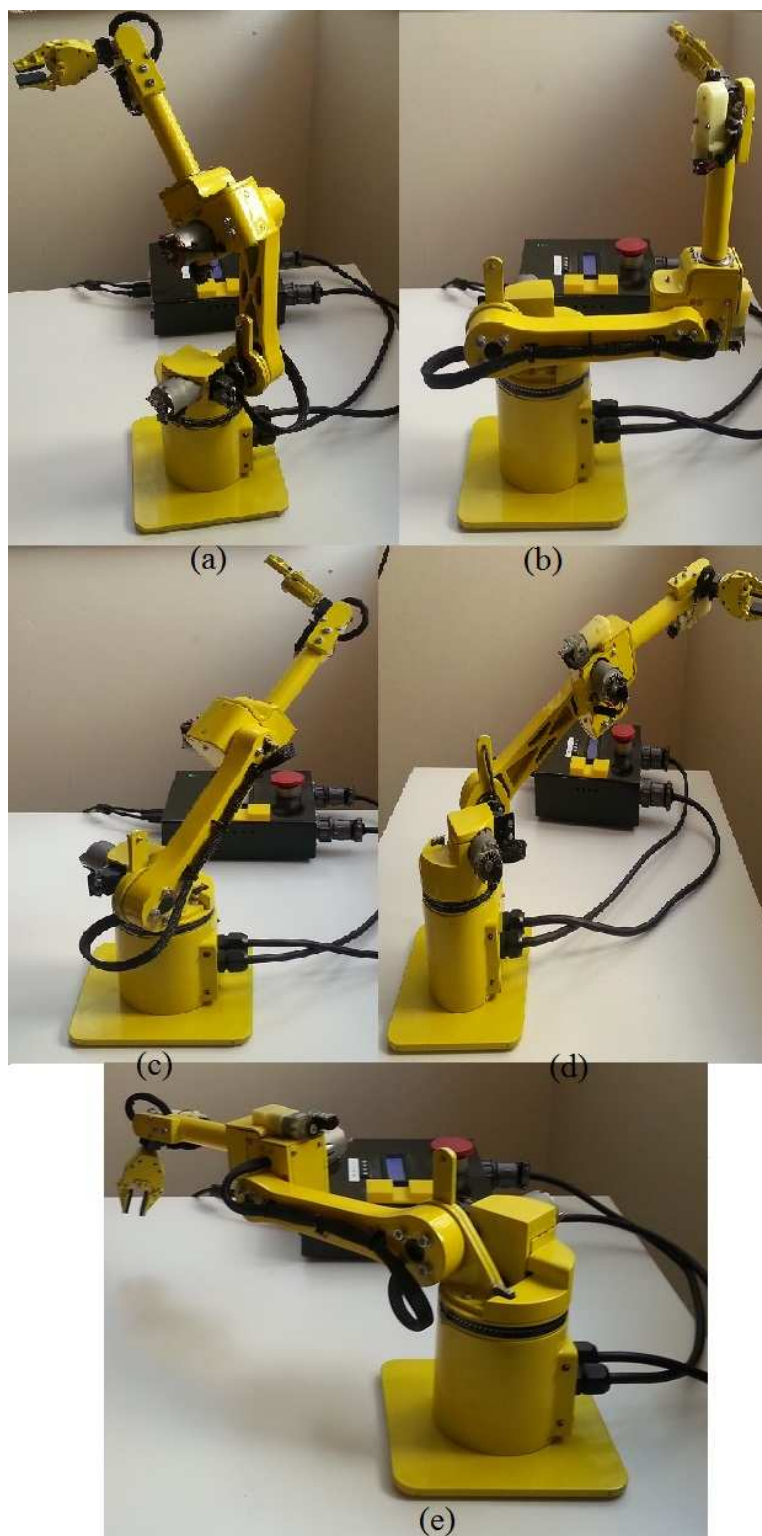
Oś	Docelowa pozycja	Wielkość wymuszenia	Wpisywana wartość
1	3500	X	$3500 \pm X$
2	1750	X	$1750 \pm X + 7000$
3	5000	X	$5000 \pm X + 14000$
4	3500	X	$3500 \pm X + 21000$
5	3500	X	$3500 \pm X + 28000$
6	2500	X	$2500 \pm X + 36000$



Rys. 36. Struktura pliku sekwencji



Pozycje z rysunków 37(b),37(e) oraz 37(a) są odtwarzane w pierwszej kolejce, natomiast pozycje z rysunków 37(c),37(d) w drugiej. Wyznaczone wzmacnienia zostały przedstawione w tablicach 4-8.



Rys. 37. Sekwencje użyte do weryfikacji. Kolejka I (a), (b), (e) kolejka II (c),(d)

Tablica 4. Wygenerowane wzmocnienia osi 1

		Pozycja osi 1	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Kolejka I	Sekwencja I	3500	52	1	77
	Sekwencja II	2792	53	1	78
	Sekwencja III	5507	55	2	80
Kolejka II	Sekwencja I	1120	55	1	80
	Sekwencja II	4279	55	1	80

Tablica 5. Wygenerowane wzmocnienia osi 2

		Pozycja osi 2	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Kolejka I	Sekwencja I	1324	57	2	117
	Sekwencja II	3489	53	1	113
	Sekwencja III	2385	53	1	113
Kolejka II	Sekwencja I	3071	54	1	114
	Sekwencja II	1866	54	1	114

Tablica 6. Wygenerowane wzmocnienia osi 3

		Pozycja osi 3	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Kolejka I	Sekwencja I	4281	55	2	115
	Sekwencja II	3134	54	1	114
	Sekwencja III	2696	55	1	115
Kolejka II	Sekwencja I	3290	56	2	116
	Sekwencja II	3064	55	2	115

Tablica 7. Wygenerowane wzmocnienia osi 4

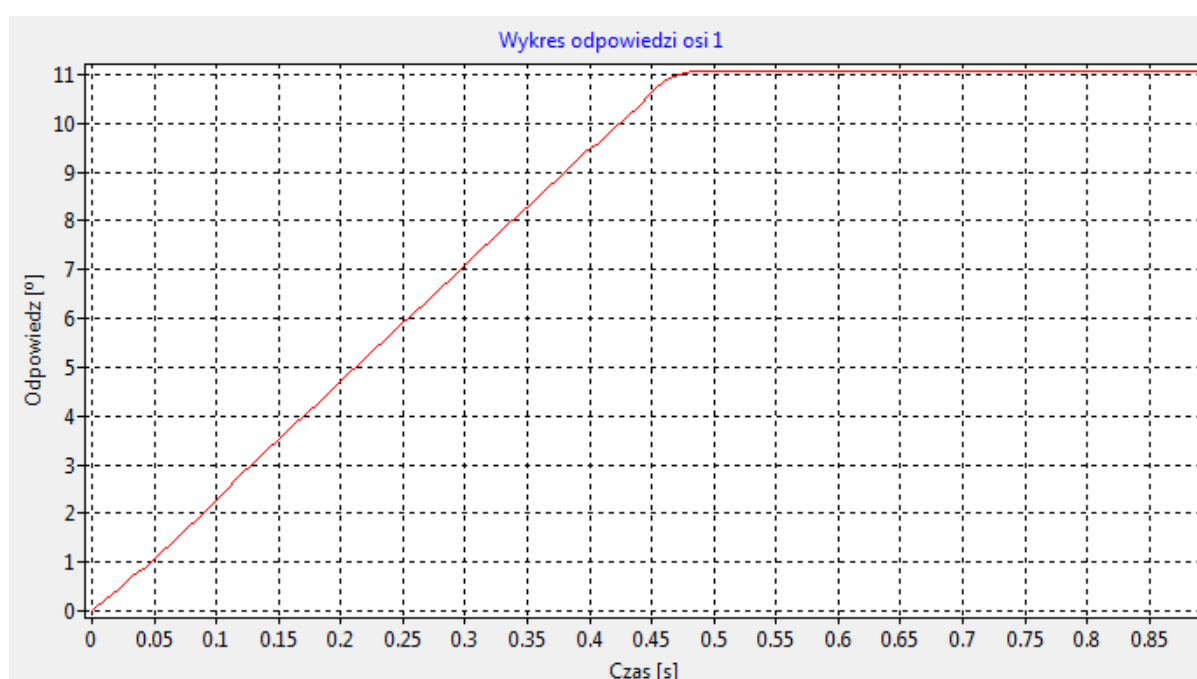
		Pozycja osi 4	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Kolejka I	Sekwencja I	4735	55	1	115
	Sekwencja II	3500	54	1	114
	Sekwencja III	2191	55	2	115
Kolejka II	Sekwencja I	4546	55	1	115
	Sekwencja II	2805	55	1	115



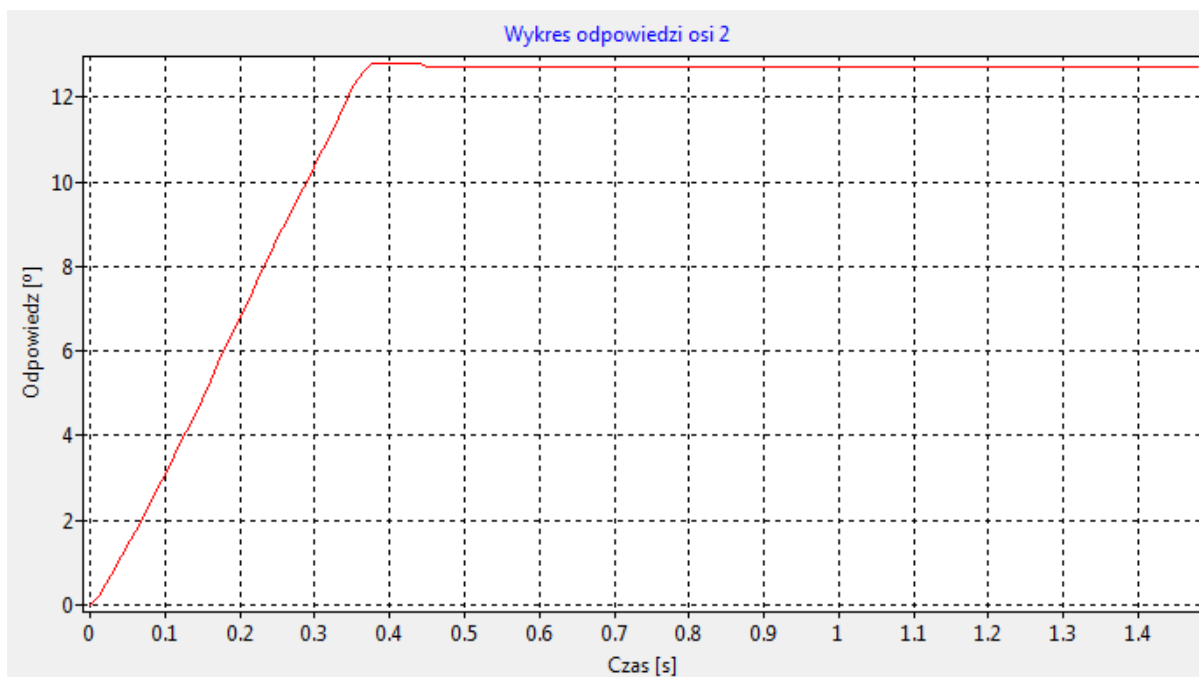
Tablica 8. Wygenerowane wzmocnienia osi 5

		Pozycja osi 5	$K_p$	$K_i$	$K_d$
Kolejka I	Sekwencja I	4194	21	3	84
	Sekwencja II	2679	23	3	86
	Sekwencja III	2600	19	2	75
Kolejka II	Sekwencja I	4080	25	3	92
	Sekwencja II	4893	24	3	89

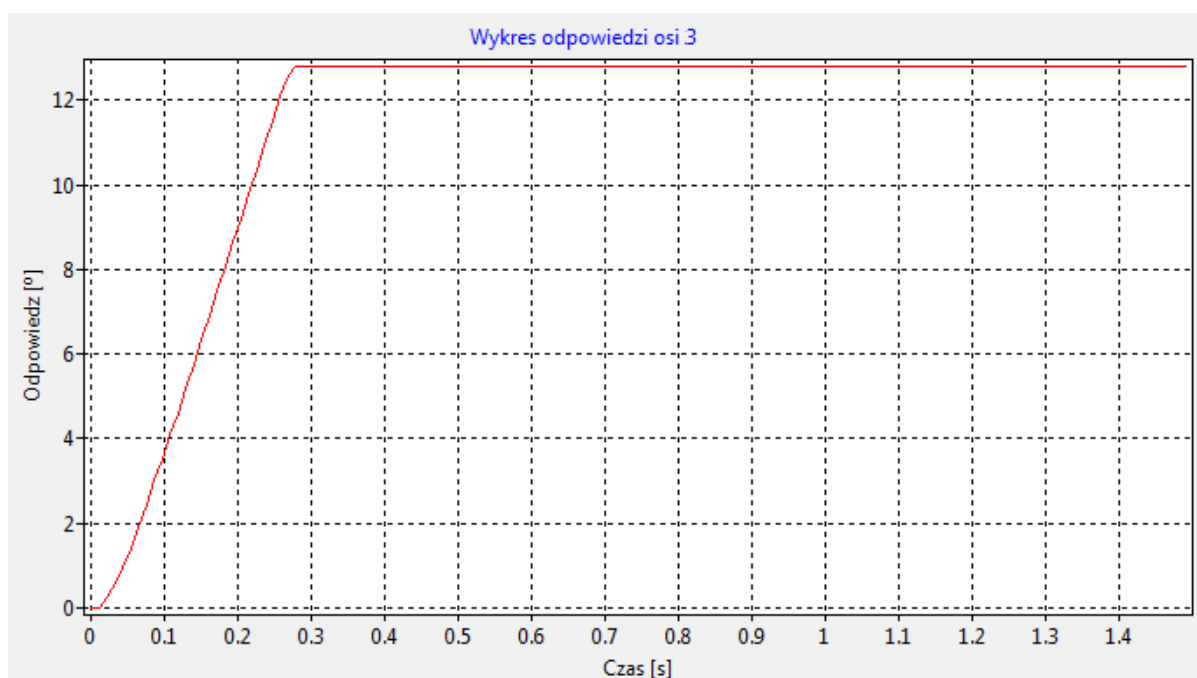
Korzystając z napisanej aplikacji wygenerowane zostały odpowiedzi skokowe wszystkich napędów. Wyniki zostały przedstawione na rysunkach 38 - 62.



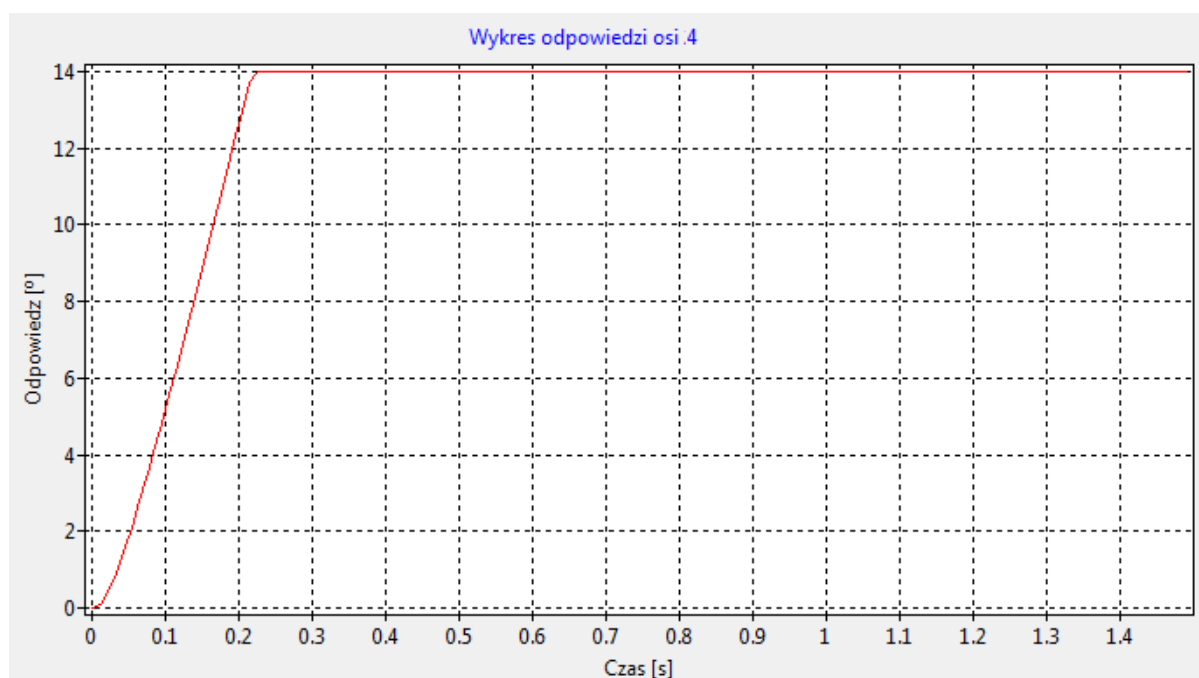
Rys. 38. Wykres odpowiedzi dla osi 1, sekwencja (a)



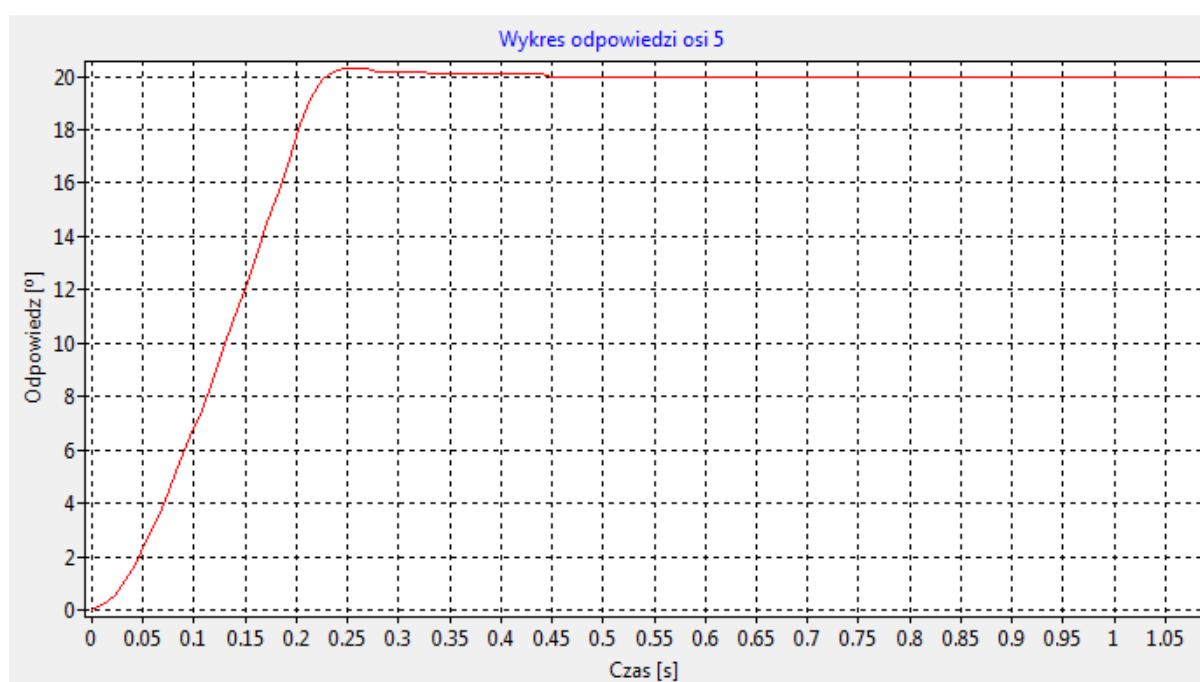
Rys. 39. Wykres odpowiedzi dla osi 2, sekwencja (a)



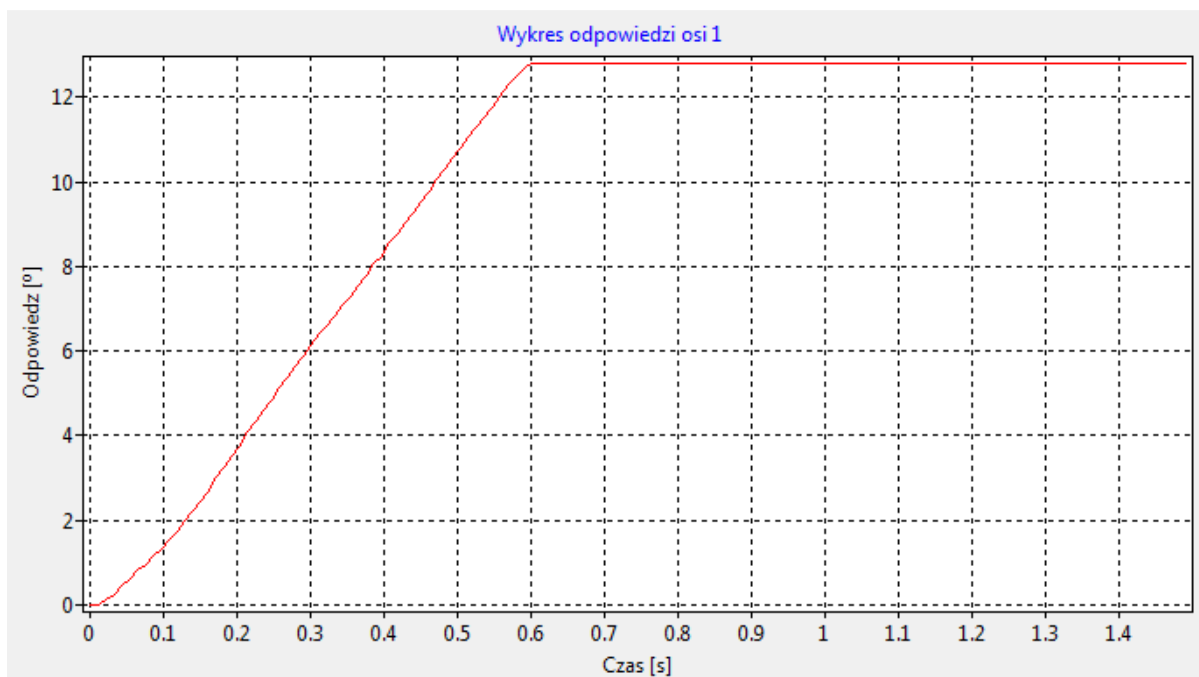
Rys. 40. Wykres odpowiedzi dla osi 3, sekwencja (a)



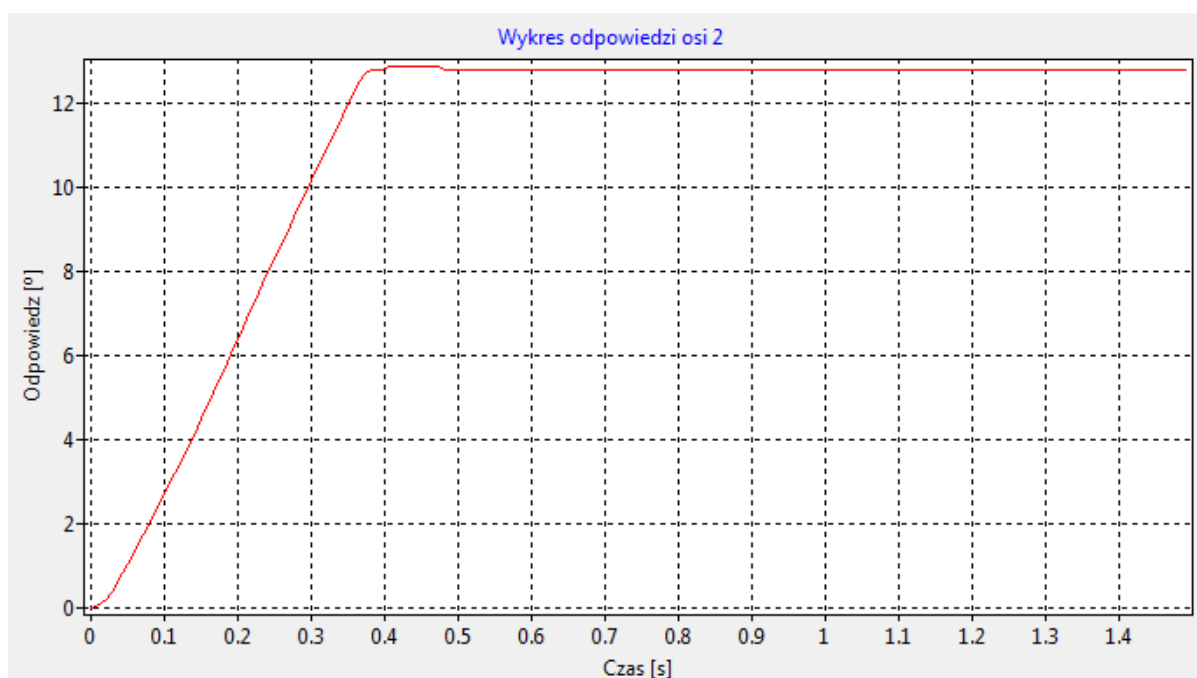
Rys. 41. Wykres odpowiedzi dla osi 4, sekwencja (a)



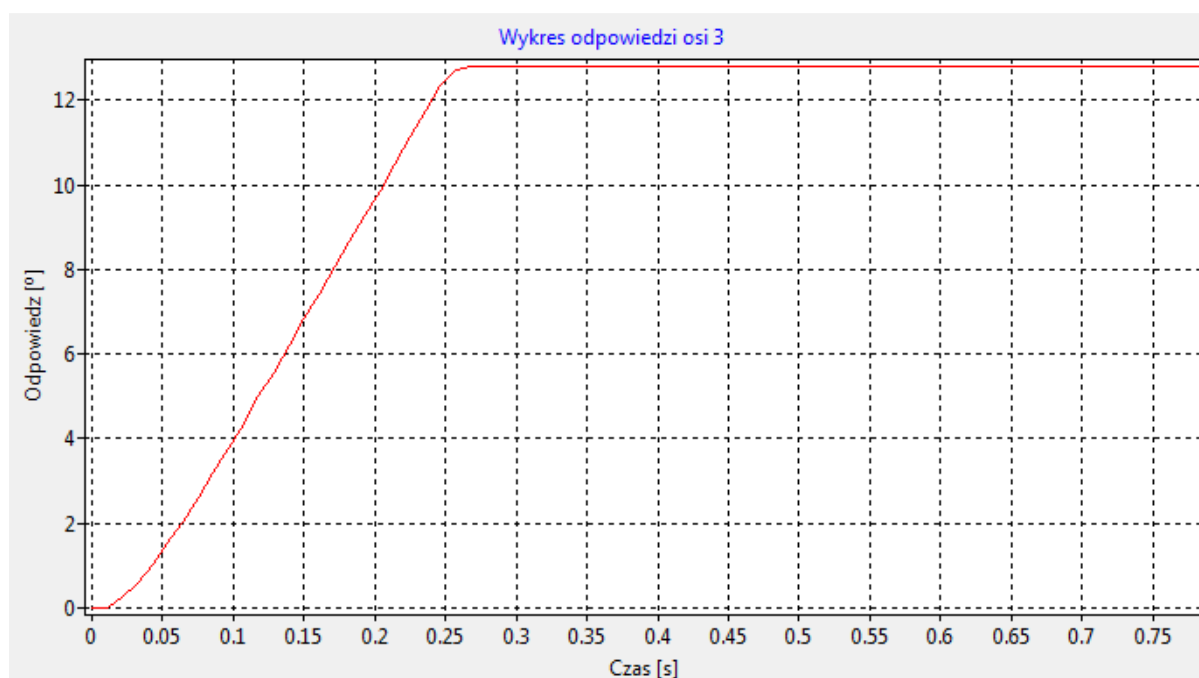
Rys. 42. Wykres odpowiedzi dla osi 5, sekwencja (a)



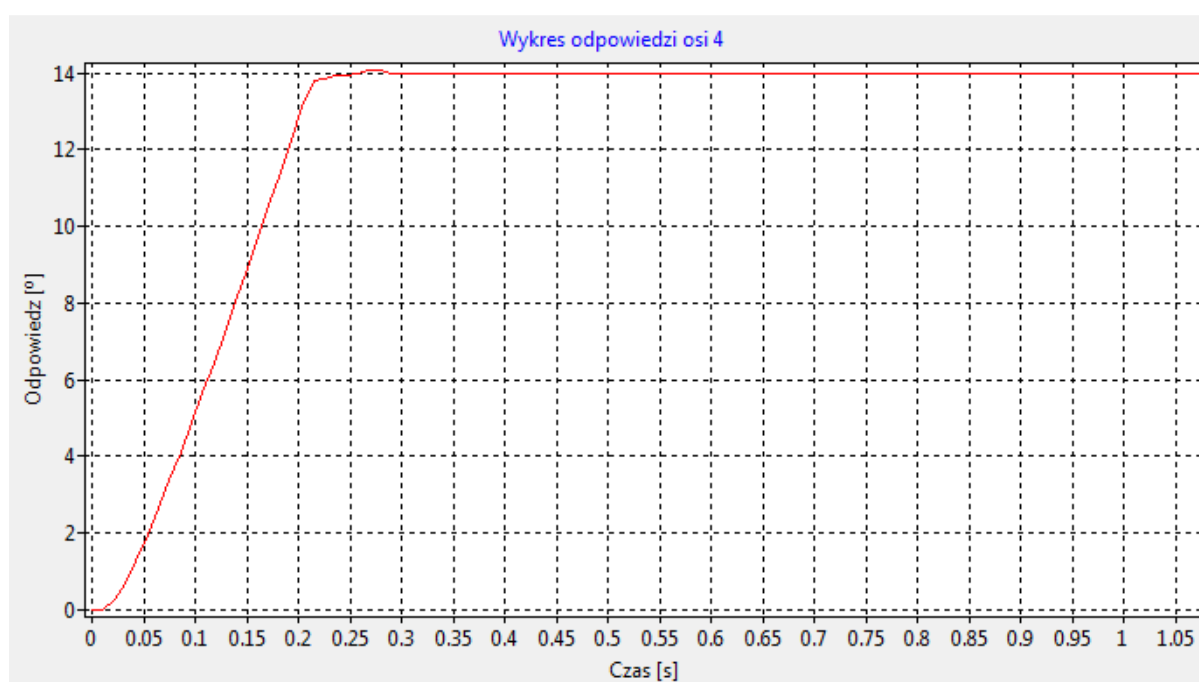
Rys. 43. Wykres odpowiedzi dla osi 1, sekwencja (b)



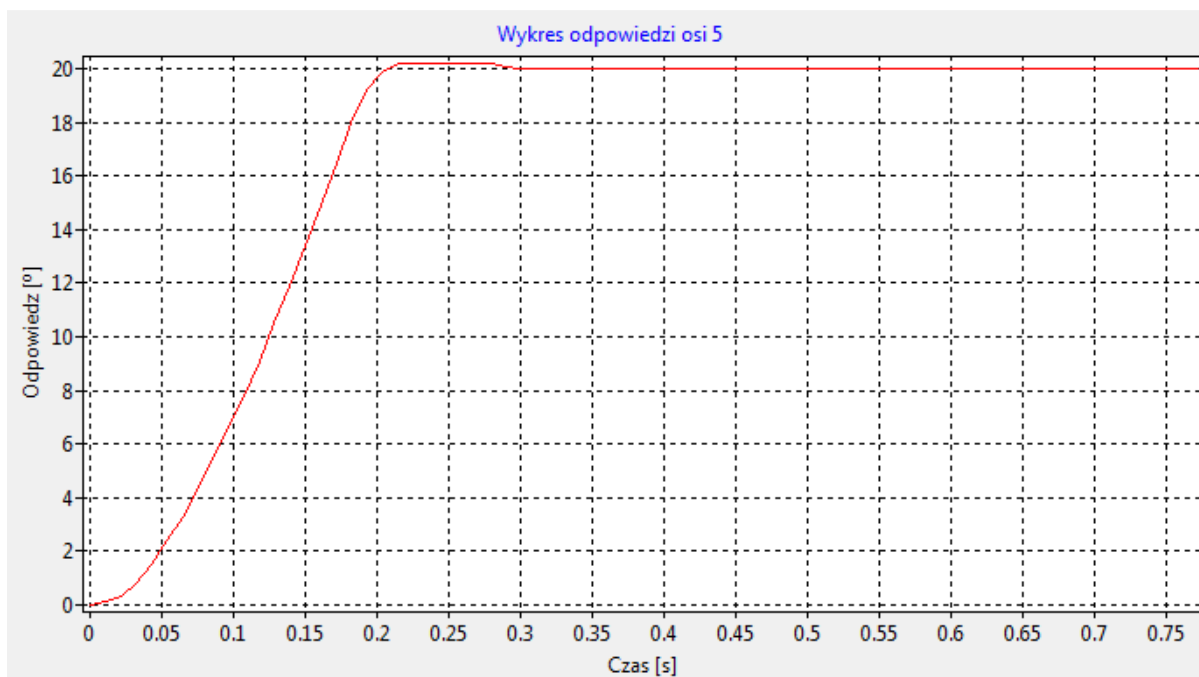
Rys. 44. Wykres odpowiedzi dla osi 2, sekwencja (b)



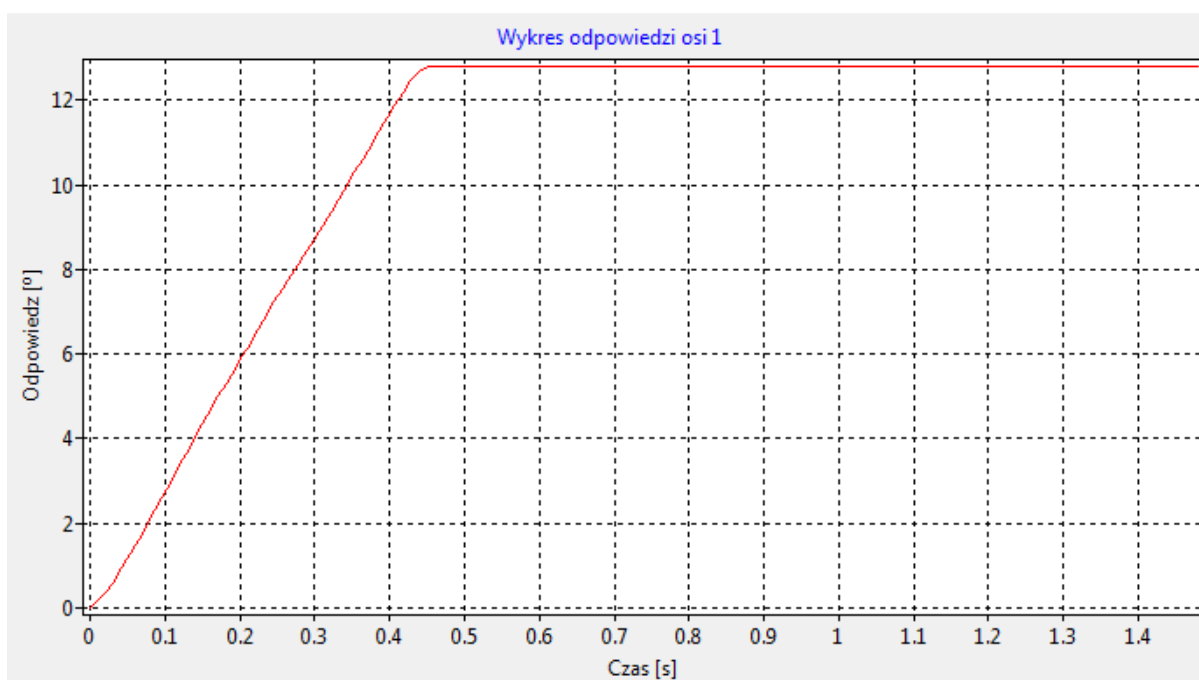
Rys. 45. Wykres odpowiedzi dla osi 3, sekwencja (b)



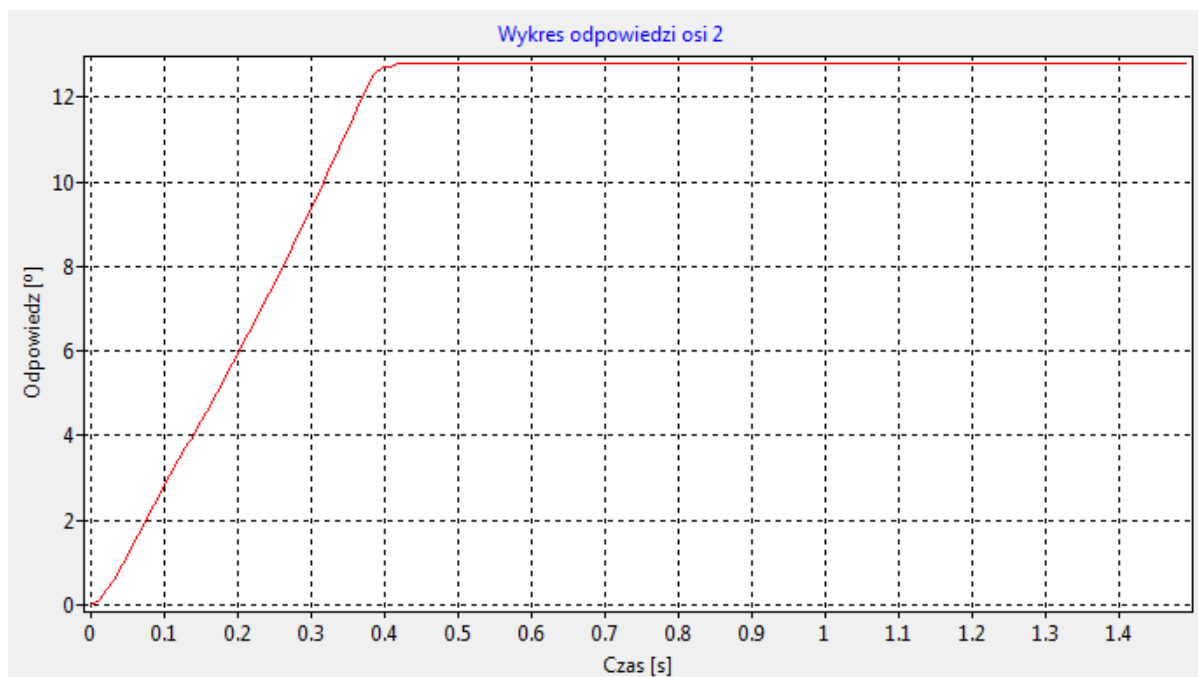
Rys. 46. Wykres odpowiedzi dla osi 4, sekwencja (b)



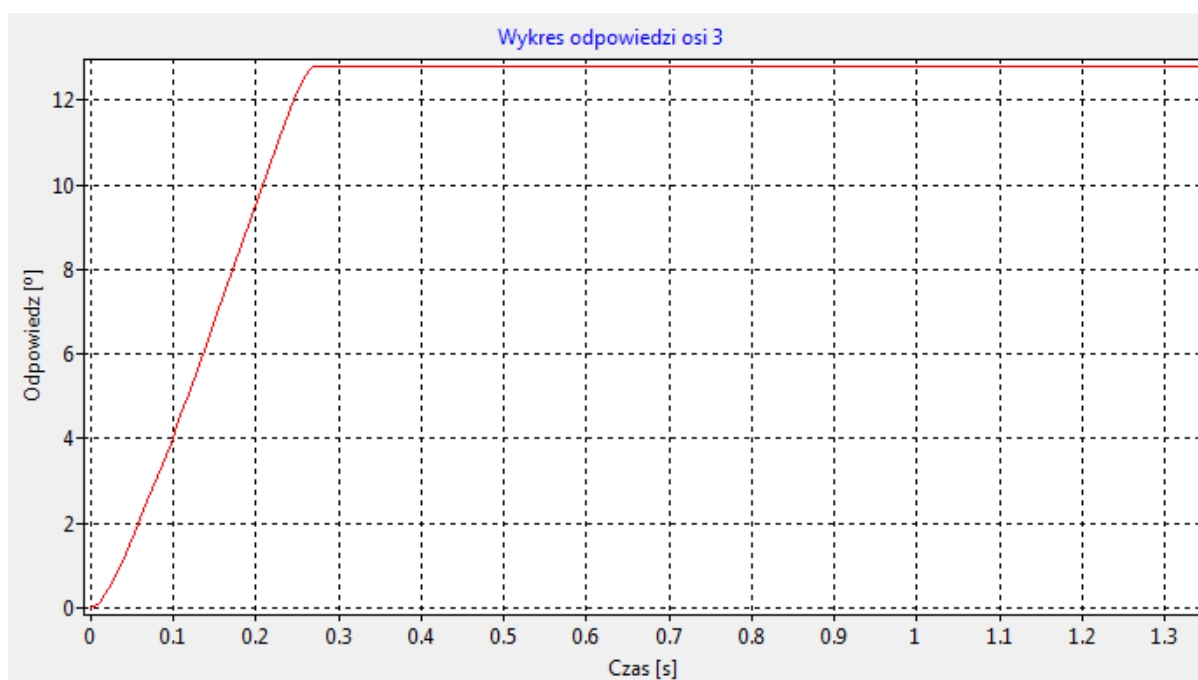
Rys. 47. Wykres odpowiedzi dla osi 5, sekwencja (b)



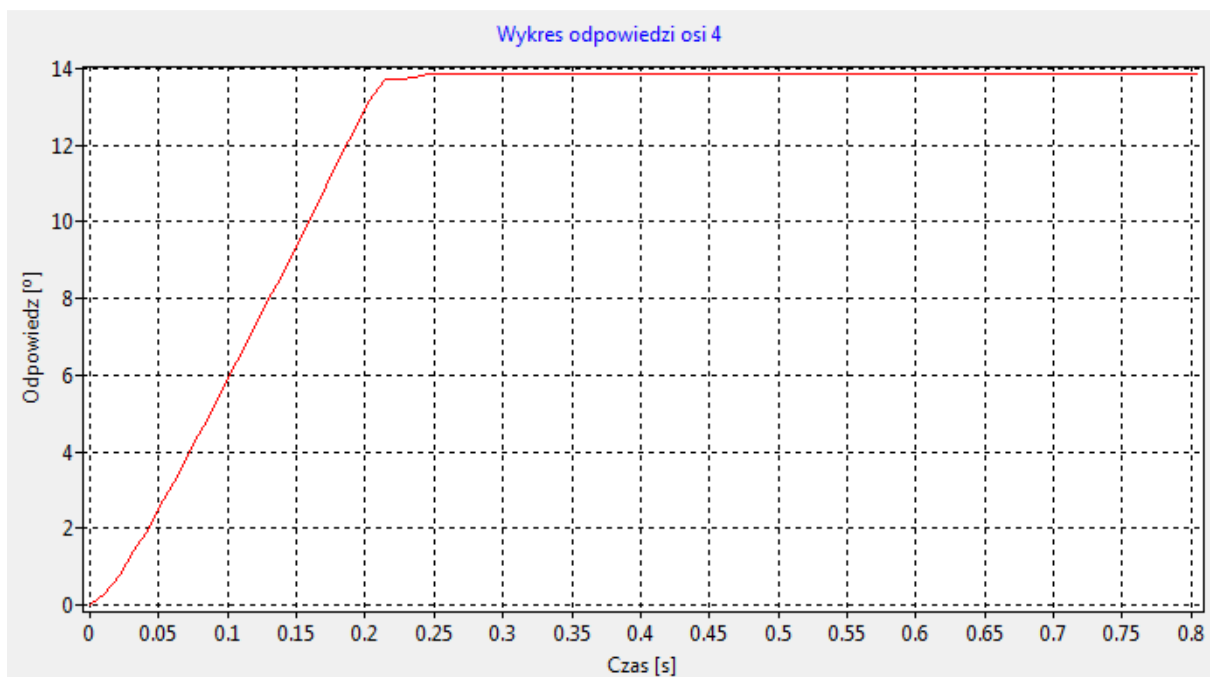
Rys. 48. Wykres odpowiedzi dla osi 1, sekwencja (c)



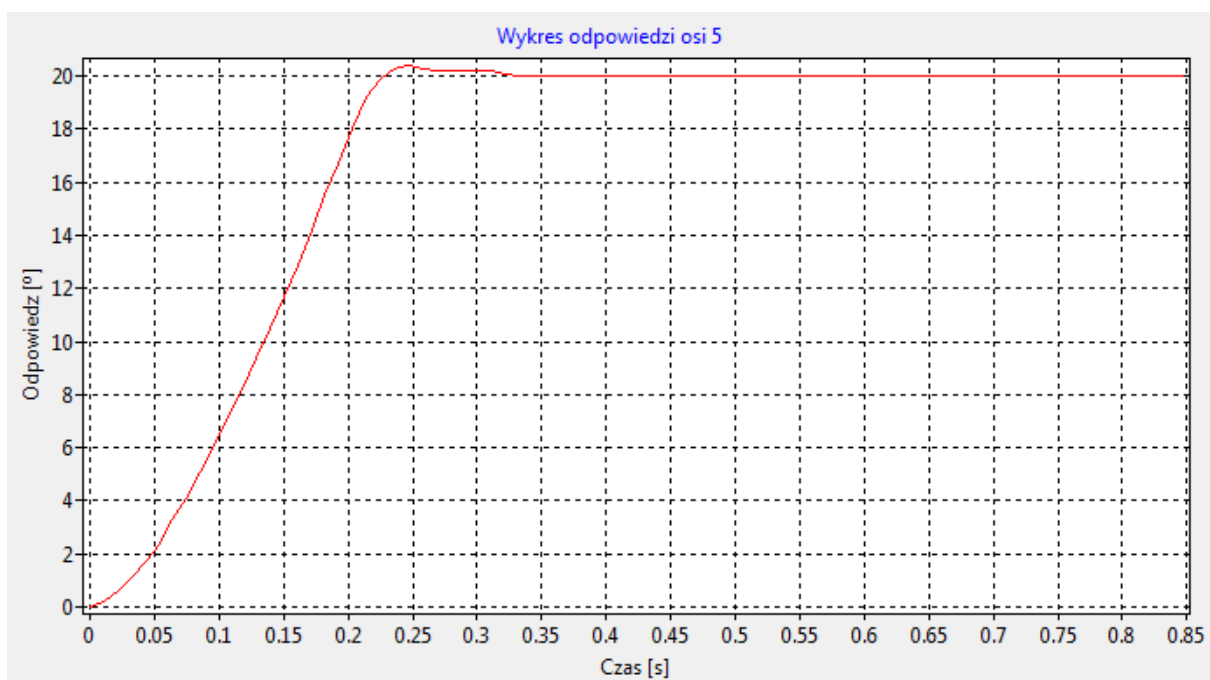
Rys. 49. Wykres odpowiedzi dla osi 2, sekwencja (c)



Rys. 50. Wykres odpowiedzi dla osi 3, sekwencja (c)

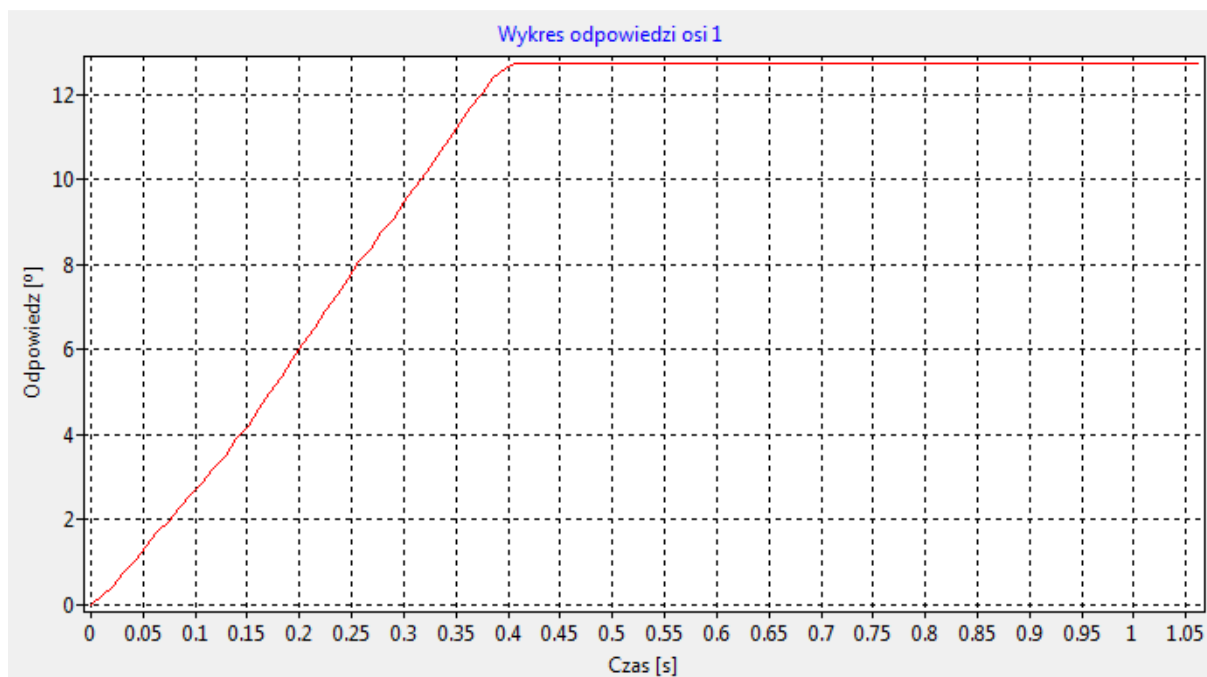


Rys. 51. Wykres odpowiedzi dla osi 4, sekwencja (c)

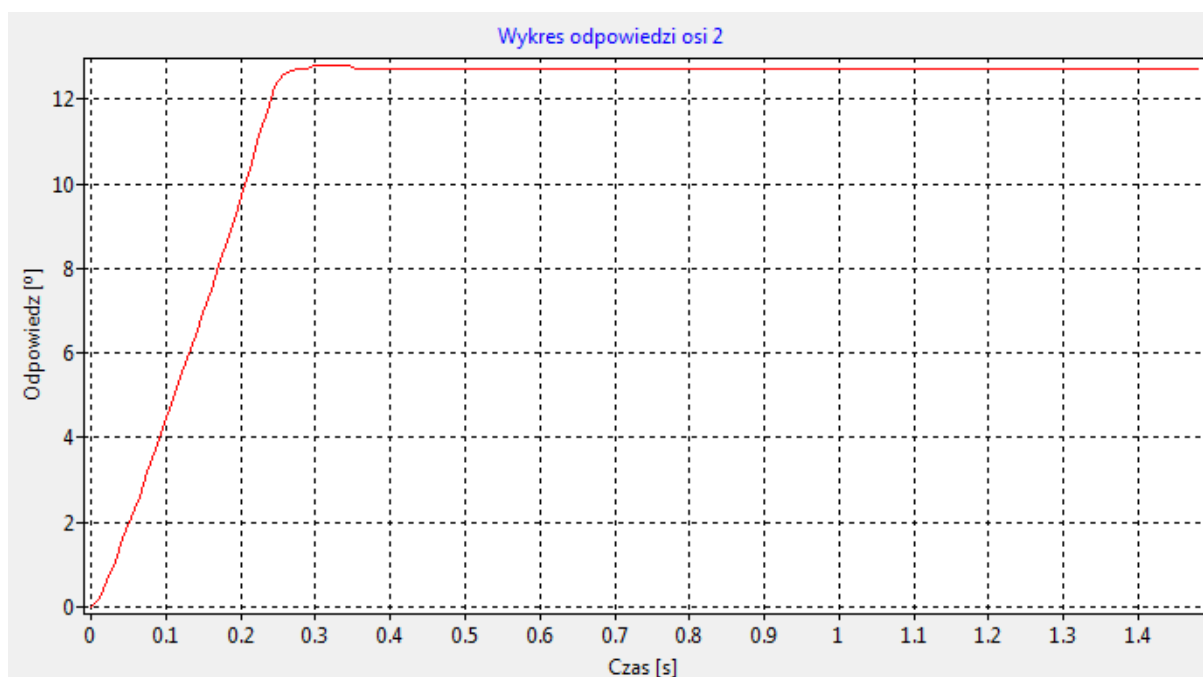


Rys. 52. Wykres odpowiedzi dla osi 5, sekwencja (c)

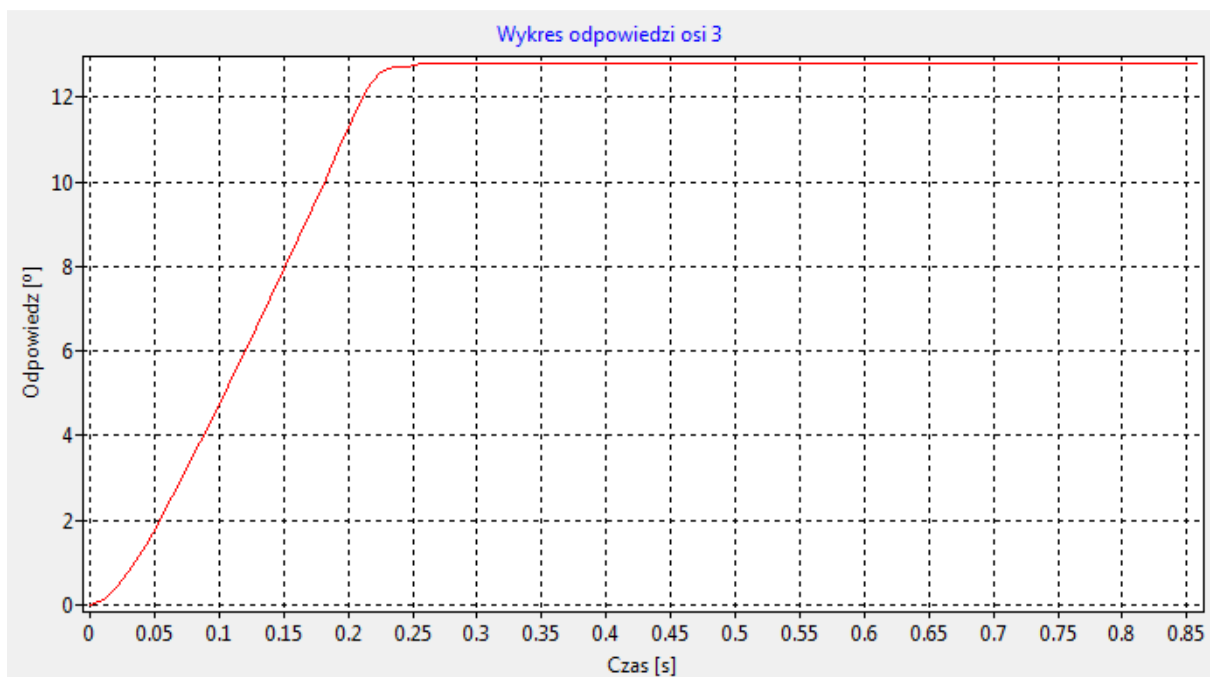




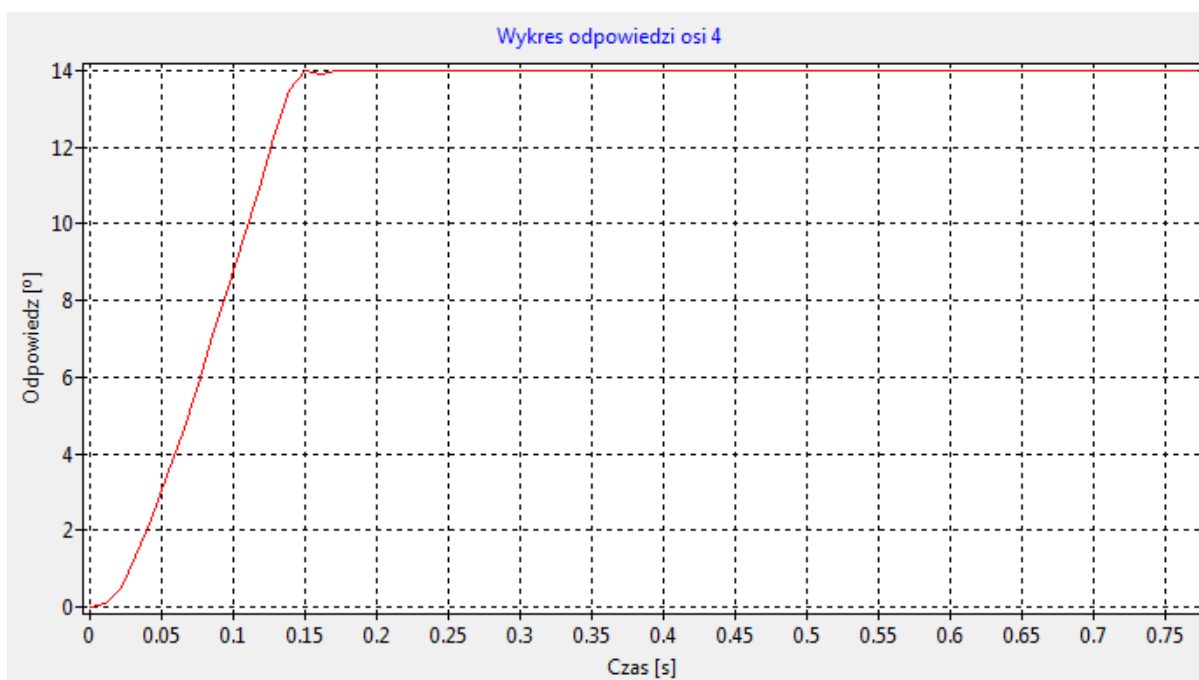
Rys. 53. Wykres odpowiedzi dla osi 1, sekwencja (e)



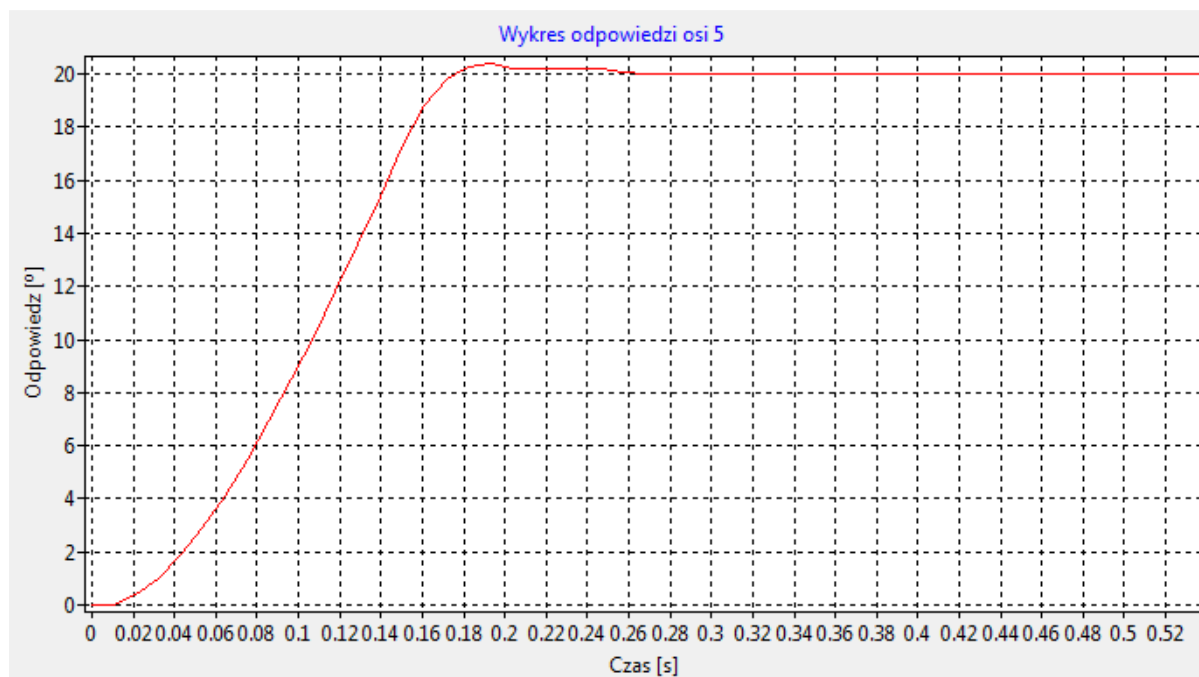
Rys. 54. Wykres odpowiedzi dla osi 2, sekwencja (e)



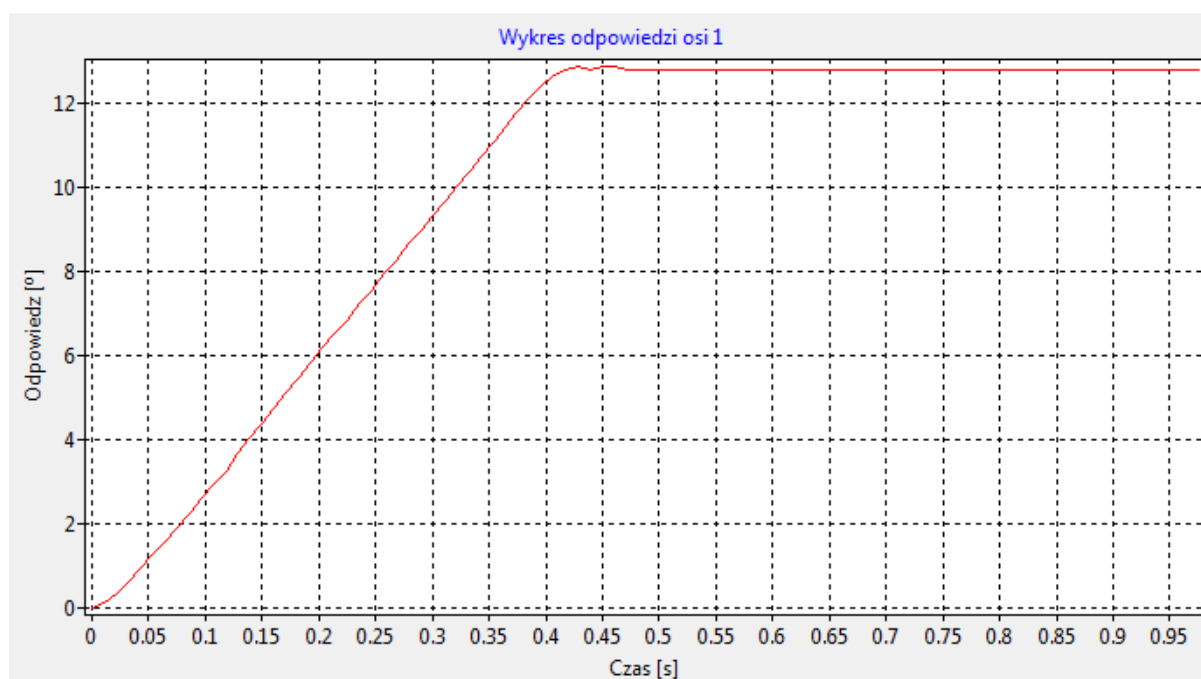
Rys. 55. Wykres odpowiedzi dla osi 3, sekwencja (e)



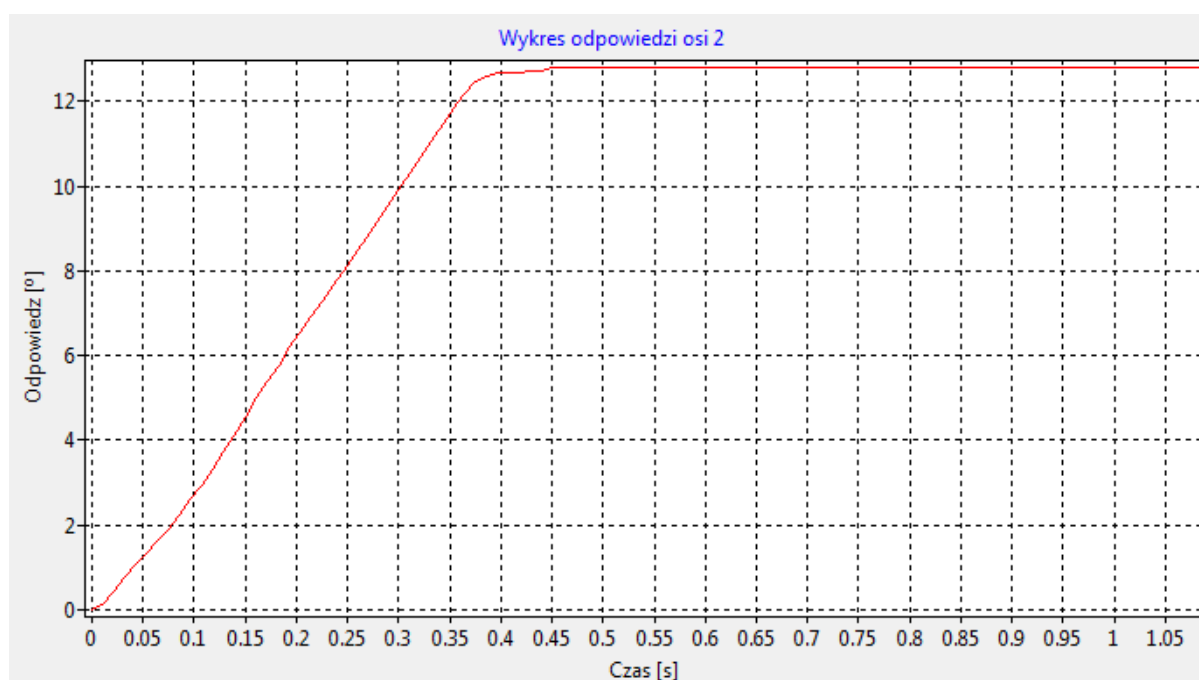
Rys. 56. Wykres odpowiedzi dla osi 4, sekwencja (e)



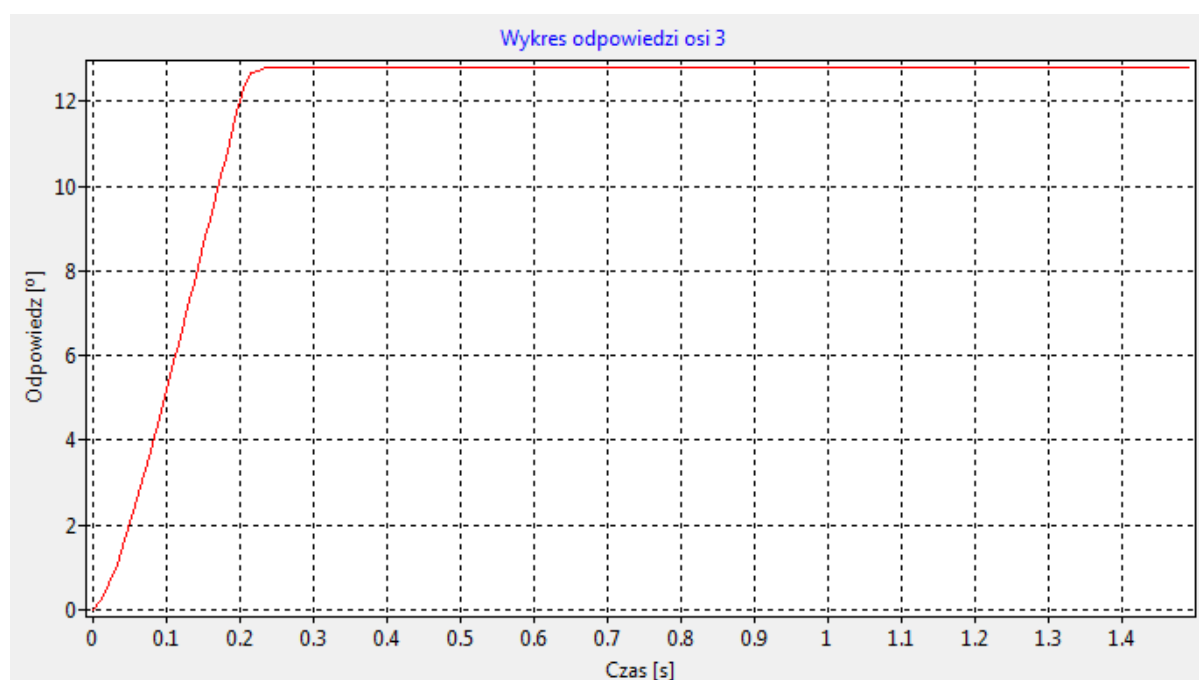
Rys. 57. Wykres odpowiedzi dla osi 5, sekwencja (e)



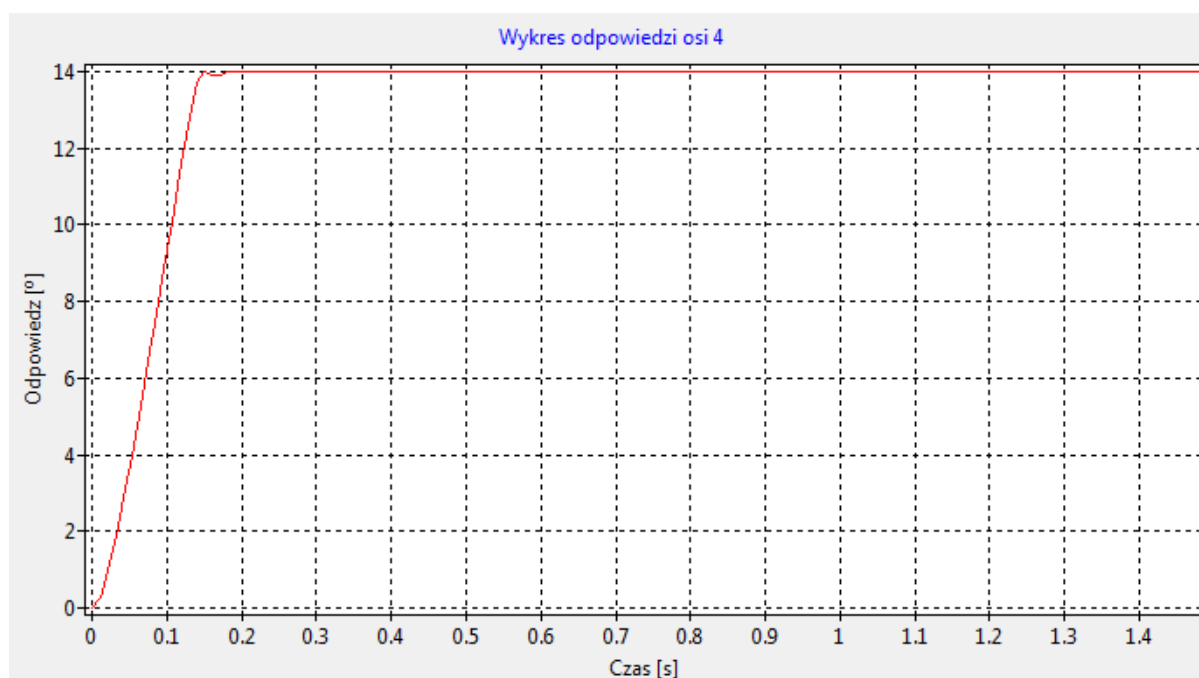
Rys. 58. Wykres odpowiedzi dla osi 1, sekwencja (d)



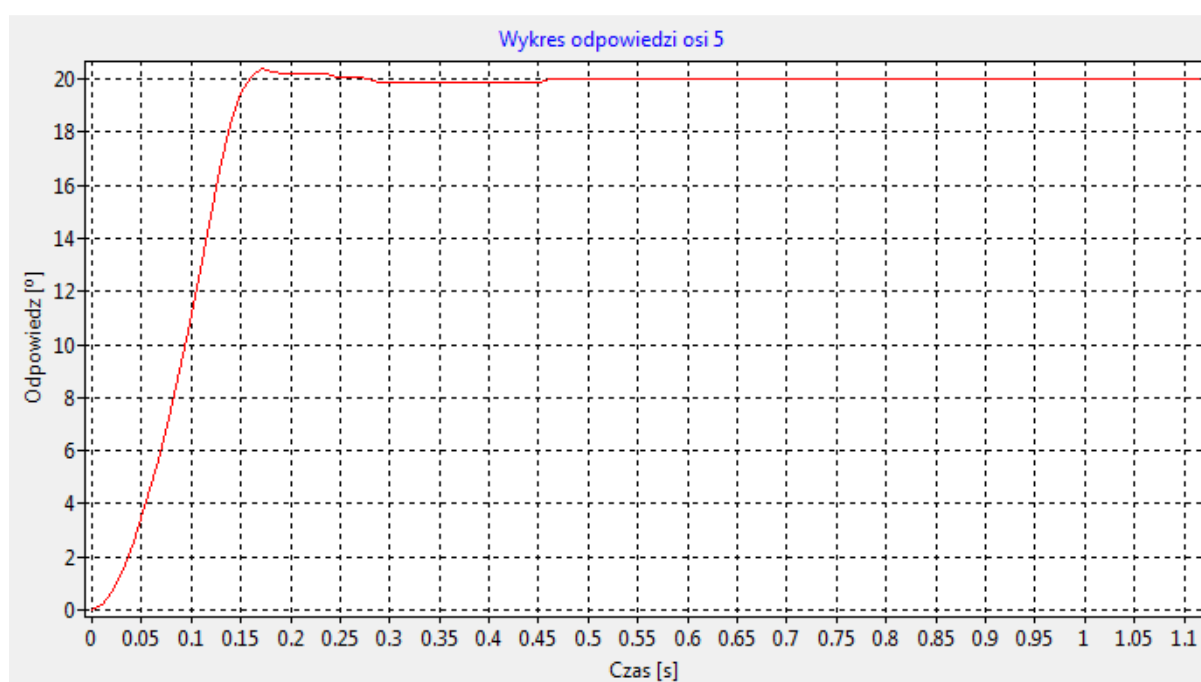
Rys. 59. Wykres odpowiedzi dla osi 2, sekwencja (d)



Rys. 60. Wykres odpowiedzi dla osi 3, sekwencja (d)



Rys. 61. Wykres odpowiedzi dla osi 4, sekwencja (d)



Rys. 62. Wykres odpowiedzi dla osi 5, sekwencja (d)

W tablicy 9 przedstawione zostały maksymalne przeregulowania dla regulatora FLC, oraz klasycznego sterowania PID.

Tablica 9. Zestawienie przeregulowań dla FLC oraz PID

		Oś 1 [°]		Oś 2 [°]		Oś 3 [°]		Oś 4 [°]		Oś 5 [°]	
		FLC	PID	FLC	PID	FLC	PID	FLC	PID	FLC	PID
Kolejka I	Sek. I	0	0,192	0,064	0,064	0	0,128	0	0	0,3	0,6
	Sek. II	0	0,192	0,064	0,064	0	0	0,07	0,07	0,2	0,4
	Sek. III	0	0,128	0	0	0	0	0	0	0,1	0,9
Kolejka II	Sek. I	0	0,32	0,064	0,064	0	0,128	0	0,07	0,4	0,5
	Sek. II	0,064	0	0	0	0	0,256	0	0	0,4	0,2

## 11. Podsumowanie

W pracy przedstawiono badania regulatora rozmytego na rzeczywistym obiekcie. Jego działanie zostało zweryfikowane poprzez badanie odpowiedzi skokowej poszczególnych napędów. Osiągnięte wyniki są zadowalające i w większości mieszczą się w ustalonej granicy błędu 1% poza osią 5.

W tablicach 4 – 8 zamieszczone zostały wygenerowane wzmocnienia regulatorów przez FLT. Można zauważyć, że zakres ich zmian jest stosunkowo mały. Duży wpływ na to ma konstrukcja manipulatora. W typowym manipulatorze przemysłowym stosowane są przekładnie o przełożeniach około 1:100 - 1:200, natomiast w prototypie, dla którego opracowano system przełożenia wynoszą 1:160 – 1:298 co bezpośrednio przekłada się na jego dynamikę. Dodatkowo napęd osi 2 posiada znaczny zapas mocy.

Analizując tabele 9 widoczne jest zmniejszenie wielkości przeregulowań w stosunku do klasycznego regulatora PID. Największe przeregulowania w stosunku do reszty osi występują w osi piątej i przekraczają założoną granicę błędu 1%. Przekładnia tej osi wykonana jest z tworzywa sztucznego oraz posiada największy z wszystkich osi luz wynoszący około 8°, jednakże zastosowanie FLC pozwoliło zmniejszyć łączną sumę przeregulowań prawie o połowę.

Do wad opracowanego systemu można zaliczyć jego pracochłonność. Opracowanie sprawnie działającego modelu zajmuje sporo czasu, dodatkowo wymagana jest spora wiedza ekspercka na temat zachowania się samego manipulatora i wpływu poszczególnych członów na siebie. Jakikolwiek błędy w bazie reguł są praktycznie niemożliwe do wykrycia z racji jej wielkości. W takim przypadku najlepszym rozwiązaniem jest usunięcie bazy i utworzenie nowej od podstaw. Dane wejściowe podzielone zostały na trzy zmienne lingwistyczne, co nie opisuje ich w najlepszy sposób. Jednakże zwiększenie ilości zmiennych znacznie komplikuje model, sprawiając że w tej sytuacji, opłacalne może być zastosowanie innej metody sterowania, która okazałaby się mniej pracochłonna.

## Bibliografia

1. Szkodny T., *"Dynamika robotów przemysłowych"*. Gliwice, Wyd. Pol. Śl. 2013.
2. Morecki A., Knapczyk J. *„Podstawy robotyki teoria i elementy manipulatorów i robotów”*, Wydanie II, WNT, Warszawa 1994.
3. Yager R.R., Filev D.P., *”Podstawy modelowania i sterowania rozmytego”*, wydanie I, WNT, Warszawa 1995.
4. Piegat A. *”Modelowanie i sterowanie rozmyte”*, EXIT, Warszawa 1999.
5. Francuz T. *AVR Praktyczne projekty*, Helion, Gliwice 2013.
6. Pamuła T. *„Aplikacje w Delphi”*, Helion, Gliwice 2011.
7. Butkiewicz B.S., *”Elektronika Praktyczna”*, *”Układy rozmyte, część 2”*, 2000, nr 4, s. 91-92.
8. Butkiewicz B.S., *”Elektronika Praktyczna”*, *”Układy rozmyte, część 1”*, 2000, nr 3, s. 87-88.
9. Tomera M., *„Automatyka-Elektryk-Zakłócenia”*, *„Regulator PID”*, 2010, nr 2, s. 89-93.
10. Tomera M., *”Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej”*, *„Rozmyty samonastrajalny regulator cyfrowy pid w układzie sterowania poziomem wody”*, 2013, nr 36, s. 165-170.
11. Akyuz I.H., Bingul Z., Kizir S., *”Turk J Elec Eng & Comp Sci”*, *„Cascade fuzzy logic control of a single – link flexibe-joint manipulator”*, 2012, nr 5.
12. Rezoug A., Boudoua S., Hamerlain F., *”Journal of electrical system”*, *”Fuzzy Logic Control for Manipulator Robot actuated by Pneumatic Artifical Muscles”*, 2009, nr 1, s. 1-6.
13. Niewiara Ł., Zawirski K., *”Poznan university of technology academic journals”*, *„Auto-strojenie regulatora typu pid z wykorzystaniem logiki rozmytej”*, 2013, nr 75, s. 79-84
14. [http://www.mathworks.com/help/pdf\\_doc/fuzzy/fuzzy.pdf](http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf)
15. [http://vega.am.gdynia.pl/~tomera/ts/pid\\_lab.pdf](http://vega.am.gdynia.pl/~tomera/ts/pid_lab.pdf) z dnia 31.08.2014.
16. [http://galaxy.uci.agh.edu.pl/~o\\_iwona/podstawy\\_aut/7.1.3.pdf](http://galaxy.uci.agh.edu.pl/~o_iwona/podstawy_aut/7.1.3.pdf) z dnia 31.08.2014.
17. <http://www.lazarus.freepascal.org/> z dnia 31.08.2014.
18. <http://www.fuzzylite.com/> z dnia 31.08.2014.