

SPRAWOZDANIE Z PIERWSZEGO  
PROJEKTU Z PRZEDMIOTU  
„SZTUCZNA INTELIGENCJA W  
AUTOMATYCE”

Numer zadania: 10

Wykonawcy:

Daniel Giełdowski

Piort Chachuła

# Spis treści

<b>1. Opis obiektu</b>	2
<b>2. Zadanie 1</b>	3
2.1. Implementacja i analiza modelu obiektu	3
2.2. Model liniowy	5
2.3. Regulator	7
<b>3. Zadanie 2</b>	11
3.1. Obiekt rozmyty	11
3.2. DMC rozmyty	17
<b>4. Numeryczny rozmyty regulator SL</b>	21
<b>5. Numeryczny rozmyty regulator NPL</b>	23
<b>6. Skrypty</b>	25
6.1. Zadanie 1	25
6.2. Zadanie 2	25
6.3. Zadanie 3	25
6.4. Zadanie 4	25

# 1. Opis obiektu

Zadanie polegało na zaimplementowaniu oraz przebadaniu określonych algorytmów dla podanego przez prowadzącego obiektu. Podany przez prowadzącego obiekt miał postać układu dwóch zbiorników wodnych. Do pierwszego zbiornika wpływa woda z dwóch dopływów: sterującego  $F_1$  oraz zakłócającego  $F_D$ . Woda wypływa ze spodu pierwszego zbiornika strumieniem  $F_2$  (który zależy od wysokości wody w pierwszym zbiorniku  $h_1$ ) i wpada do drugiego zbiornika. Następnie woda w drugim zbiorniku wypływa z jego spodu strumieniem  $F_3$ , zależnym od poziomu wody w drugim zbiorniku  $h_2$ . Zmienną regulowaną w układzie jest poziom wody w drugim zbiorniku, natomiast zmienną sterującą jest strumień  $F_{1in}$ , którego przedłużeniem jest strumień  $F_1$ . Wszystko opisane jest poniżej w układzie równań 1.1.

$$\left\{ \begin{array}{l} \frac{dV_1}{dt} = F_1 + F_D - F_2 \\ \frac{dV_2}{dt} = F_2 - F_3 \\ F_2(h_1) = \alpha_1 \sqrt{h_1}, \quad F_3(h_2) = \alpha_2 \sqrt{h_2}, \quad V_1(h_1) = C_1 * h_1^2, \quad V_2(h_2) = C_2 * h_2^2 \\ F_1(t) = F_{1in}(t - \tau) \end{array} \right. \quad (1.1)$$

W równaniach tych występują stałe zmienne o wartościach:

- $C_1 = C_2 = 0.95$
- $\alpha_1 = \alpha_2 = 16$
- $\tau = 50s$

Dla układu podany został stały punkt pracy o wartościach:

- $F_1 = 54cm^3/s$
- $F_D = 10cm^3/s$
- $h_2 = 16cm$

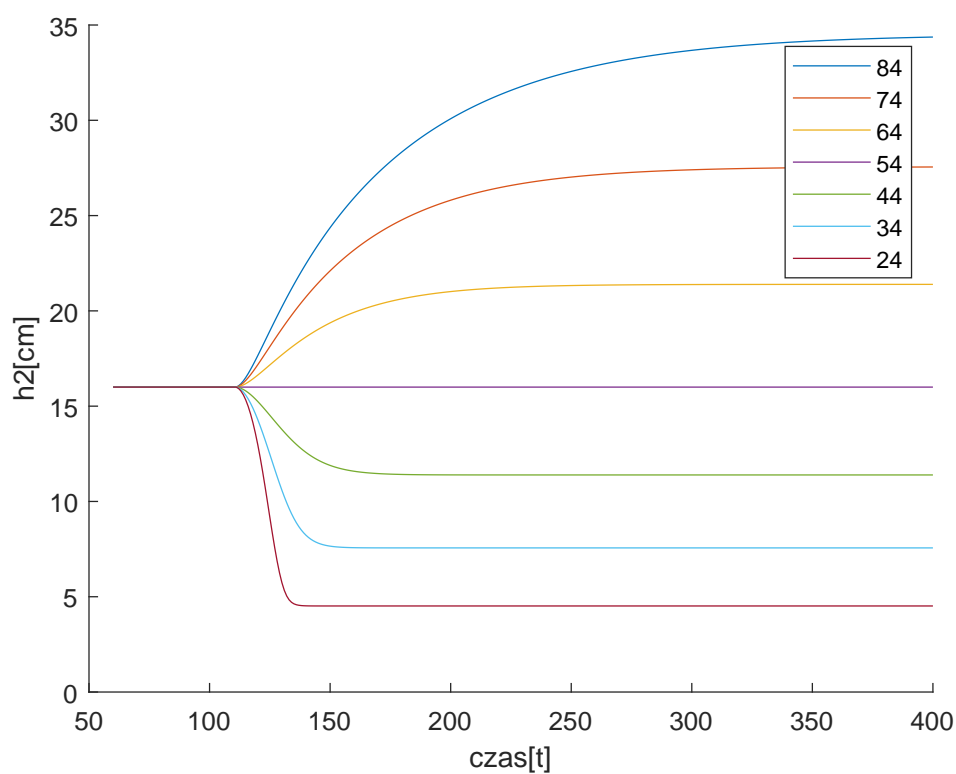
## 2. Zadanie 1

### 2.1. Implementacja i analiza modelu obiektu

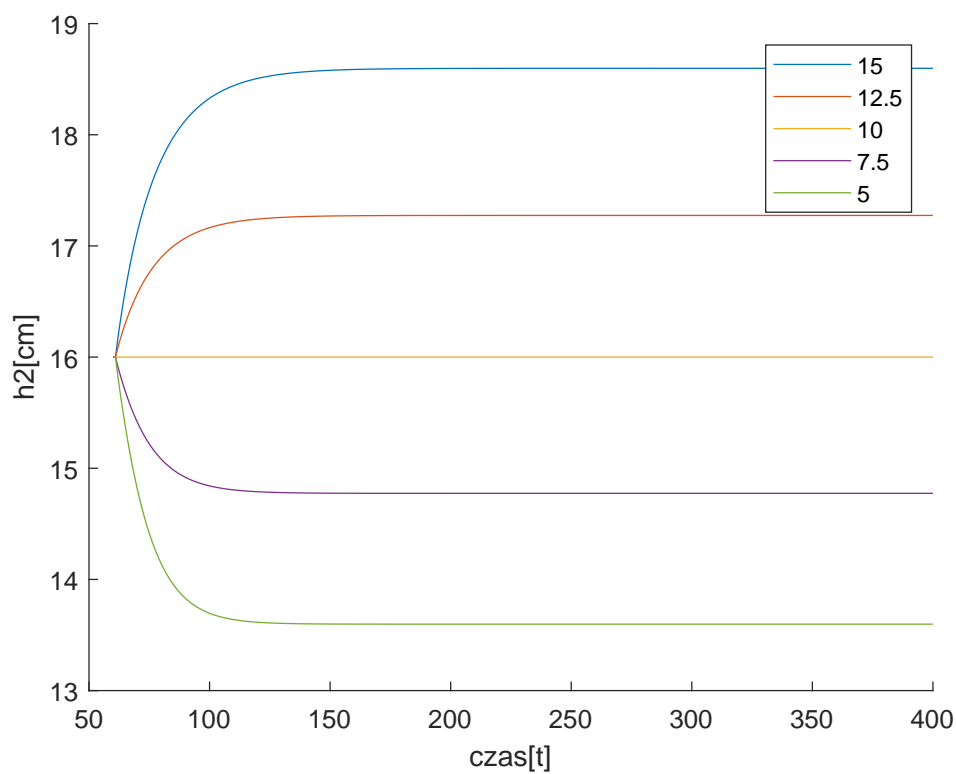
W celu zaimplementowania obiektu w programie Matlab potrzebna była postać dyskretna opisujących go równań. Dla wyznaczenia wzoru dyskretnego pochodnej zdecydowaliśmy się na okres próbkowania równy 1s. Otrzymane po dyskretyzacji i przekształceniach wzory wyglądają następująco:

$$\left\{ \begin{array}{l} V_1(k) = V_1(k-1) + F_1(k-1) + F_D(k-1) - \alpha_1 \sqrt{h_1(k-1)} \\ V_2(k) = V_2(k-1) + \alpha_1 \sqrt{h_1(k-1)} - \alpha_2 \sqrt{h_2(k-1)} \\ h_1(k) = \sqrt{\frac{V_1(k)}{C_1}} \\ h_2(k) = \sqrt{\frac{V_2(k)}{C_2}} \\ F_1(k) = F_{1in}(k-50) \end{array} \right. \quad (2.1)$$

W celu zanalizowania modelu obiektu przeprowadzone zostało badanie reakcji wyjścia  $h_2$  na skok sterowania  $F_{1in}$  oraz zakłócenia  $F_D$ . Skoki zostały przeprowadzone w chwili  $t = 60s$ . Skoki wykonywane były startując z podanego punktu pracy dla wartości od 24 do 84 co 10 dla sterowania oraz od 5 do 15 co 2.5 dla zakłócenia. Otrzymane wyniki przedstawione zostały na wykresach 2.1 oraz 2.2. Jak widać z wykresu dla zmiennego sterowania obiekt z całą pewnością nie jest liniowy. Już przy zmianie sterowania z 54 na 64 różnica wartości wyjścia jest widocznie większa niż przy zmianie z 54 na 44. Porównując przebiegi dla różnych wartości skoków możemy stwierdzić, że przy zwiększaniu sterowania obiekt staje się coraz łagodniejszy, ale różnica w wartości wyjścia zwiększa się. Dla mniejszych sterowań wynik jest odwrotny. Dla zmian zakłócenia obiekt także jest nieliniowy, jednakże różnica między przebiegami dla tak małych zmian jest niemalże niezauważalna. Na wykresach widać także wpływ opóźnienia sterowania  $F_{1in}$  względem wejścia  $F_1$ . Zmiany wartości sterowania spotykają się z opóźnioną reakcją obiektu, w przeciwieństwie do zmian zakłócenia.



Rys. 2.1. Przebiegi wyjścia dla skoku wartości sterowania w chwili 60s



Rys. 2.2. Przebiegi wyjścia dla skoku wartości sterowania w chwili 60s

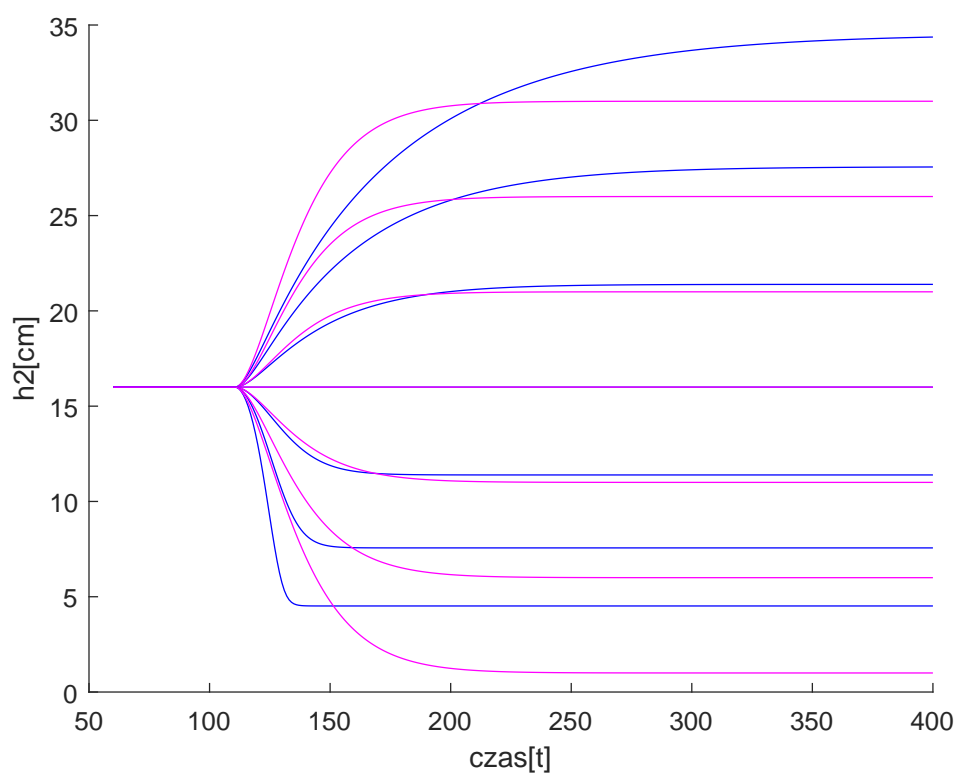
## 2.2. Model liniowy

W celu uzyskania modelu liniowego należy zlinearyzować równania obiektu w podanym punkcie pracy. Zlinearyzowane równania wyglądają następująco:

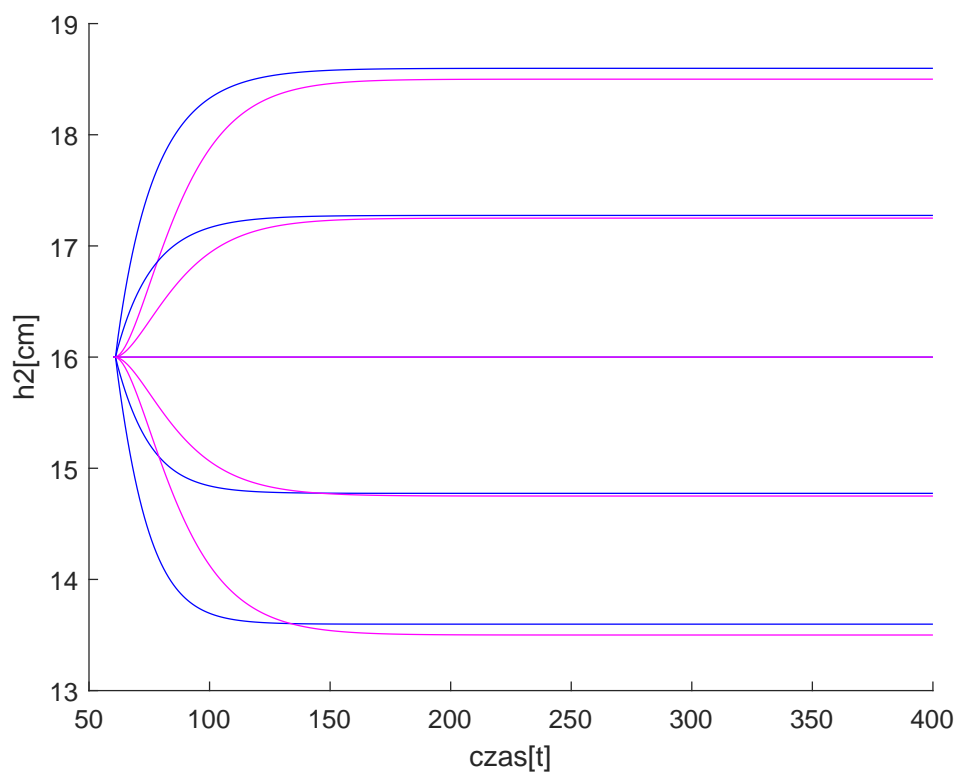
$$\left\{ \begin{array}{l} V_1(k) = V_1(k-1) + (F_1(k-1) - F_{10}) + (F_D(k-1) - F_{D0}) - \frac{\alpha_1}{2\sqrt{h_{10}}} * (h_1(k-1) - h_{10}) \\ V_2(k) = V_2(k-1) + \frac{\alpha_1}{2\sqrt{h_{10}}} * (h_1(k-1) - h_{10}) - \frac{\alpha_2}{2\sqrt{h_{20}}} * (h_2(k-1) - h_{20}) \\ h_1(k) = h_{10} + \frac{1}{2\sqrt{C_1 * V_{10}}} * (V_1(k) - V_{10}) \\ h_2(k) = h_{20} + \frac{1}{2\sqrt{C_2 * V_{20}}} * (V_2(k) - V_{20}) \\ F_1(k) = F_{1in}(k - 50) \end{array} \right. \quad (2.2)$$

O ile  $F_{10}$ ,  $F_{D0}$  oraz  $h_{10}$  mamy podane, tak  $V_{10}$ ,  $V_{20}$  oraz  $h_{10}$  musimy sobie określić. Nie jest to złożone zadanie. W punkcie pracy pochodne obydwu objętości musi być zerowa, z czego wynika, że  $F_2 = F_3 \Rightarrow \alpha_1 * \sqrt{h_1} = \alpha_2 * \sqrt{h_2}$ . Ponieważ  $\alpha_1 = \alpha_2$  to wysokość wody w punkcie pracy musi być taka sama dla obydwu zbiorników. Mając wysokości wody w obu zbiornikach można obliczyć jej objętość.

Dla uzyskanego w ten sposób modelu liniowego przeprowadzono takie same testy jak dla oryginalnego modelu. Wynik testów przedstawiony jest na wykresach 2.3 oraz 2.4, na których niebieskim kolorem widnieją przebiegi oryginalne, a różowym te dla obiektu zlinearyzowanego. Dla skoków sterowania do 44 oraz 64 przebiegi dla obydwu modeli są dosyć podobne zarówno kształtem jak i wartościami. Dla wysokich oraz niskich wartości nie jest już niestety tak dobrze. Obiekt liniowy, w przeciwieństwie do oryginału, działa z taką samą dynamiką dla całego zakresu, co oznacza, że dla wysokich wartości sterowania jest on od niego szybszy, a dla niskich wolniejszy. Co więcej im dalej od punktu pracy tym większa różnica w wartości końcowej wyjścia. Dla skoków zakłócenia osiągnęte wartości są zbliżone dla skoków do 7.5 i 12.5. Niższe oraz wyższe spadki powodują coraz większe odchylenie końcowej wartości wyjścia, choć i tak wciąż nie jest ono bardzo duże. Jeśli chodzi o dynamikę obiekt zlinearyzowany reaguje na zmianę zakłócenia nieco wolniej niż obiekt oryginalny.



Rys. 2.3. Przebiegi wyjścia dla skoku wartości sterowania w chwili 60s, dla modelu nieliniowego i liniowego



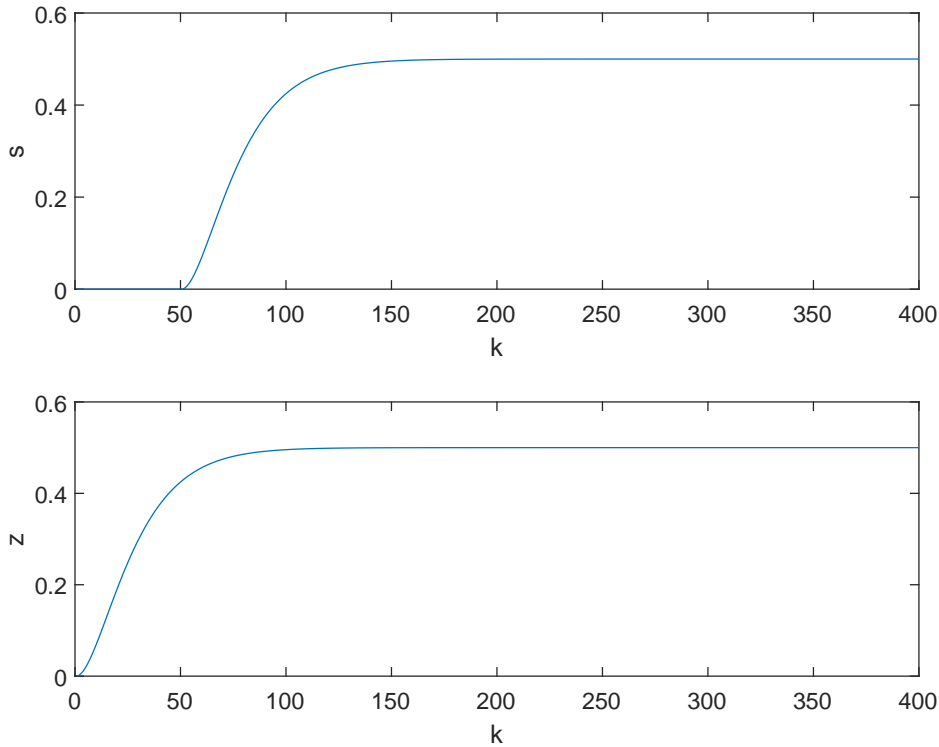
Rys. 2.4. Przebiegi wyjścia dla skoku wartości sterowania w chwili 60s, dla modelu nieliniowego i liniowego

### 2.3. Regulator

Tworząc regulator dla posiadanego obiektu zdecydowaliśmy się na regulator predykcyjny DMC. Nasza decyzja wynikała z faktu możliwości uzyskania przez nas niezaszumionej odpowiedzi skokowej obiektu na skok sterowania oraz skok zakłócenia oraz z faktu, że stosowanie regulatora DMC z uwzględnieniem zakłócenia mierzalnego jesteśmy w stanie uwzględnić zakłócenie w regulatorze.

W celu uzyskania odpowiedzi skokowych wykorzystaliśmy obiekt zlinearyzowany w punkcie pracy. Wykonując skok sterowania z punktu pracy otrzymaliśmy przebieg wyjścia obiektu. Następnie odjeliśmy od przebiegu wartość wyjścia w punkcie pracy i podzieliliśmy przez długość skoku sterowania, zgodnie z wzorem 2.3, otrzymując w ten sposób znormalizowaną odpowiedź skokową. Dla odpowiedzi skokowej zakłócenia postąpiliśmy analogicznie wykonując skok zakłócenia i dzieląc na końcu przez długość tegoż skoku. Wygląd obydwu odpowiedzi przedstawiony został na wykresie 2.5.

$$s = \frac{y - y_0}{\Delta u} \quad (2.3)$$



Rys. 2.5. Odpowiedzi skokowe dla skoku sterowania(s) i zakłócenia(z)

Do obliczania regulacji DMC używaliśmy regulatora w formie analitycznej, określonego wzorami od 2.4 do 2.13:

$$\mathbf{Y}^{\text{zad}}(k) = \begin{bmatrix} Y^{\text{zad}}(k) \\ \vdots \\ Y^{\text{zad}}(k) \end{bmatrix}_{N \times 1} \quad (2.4)$$



$$\mathbf{Y}(k) = \begin{bmatrix} y(k) \\ \vdots \\ y(k) \end{bmatrix}_{N \times 1} \quad (2.5)$$

$$\Delta \mathbf{U}(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k + N_u - 1|k) \end{bmatrix}_{N_u \times 1} \quad (2.6)$$

$$\Delta \mathbf{U}^P(k) = \begin{bmatrix} \Delta u(k-1) \\ \vdots \\ \Delta u(k - (D-1)) \end{bmatrix}_{(D-1) \times 1} \quad (2.7)$$

$$\mathbf{M} = \begin{bmatrix} s_1 & 0 & \dots & 0 \\ s_2 & s_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \dots & s_{N-N_u+1} \end{bmatrix}_{N \times N_u} \quad (2.8)$$

$$\mathbf{M}^P = \begin{bmatrix} s_2 - s_1 & s_3 - s_2 & \dots & s_D - s_{D-1} \\ s_3 - s_1 & s_4 - s_2 & \dots & s_{D+1} - s_{D-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & \dots & s_{N+D-1} - s_{D-1} \end{bmatrix}_{N \times D-1} \quad (2.9)$$

$$\mathbf{M}^{zP} = \begin{bmatrix} sz_1 & sz_2 - sz_1 & sz_3 - sz_2 & \dots & sz_{D_z} - sz_{D_z-1} \\ sz_2 & sz_3 - sz_1 & sz_4 - sz_2 & \dots & sz_{D_z+1} - sz_{D_z-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ sz_N & sz_{N+1} - sz_1 & sz_{N+2} - sz_2 & \dots & sz_{N+D_z-1} - sz_{D_z-1} \end{bmatrix}_{N \times D_z} \quad (2.10)$$

$$\mathbf{Y}^0(k) = \mathbf{Y}(k) + \mathbf{M}^P \Delta \mathbf{U}^P(k) + \mathbf{M}^{zP} \Delta \mathbf{Z}^P(k) \quad (2.11)$$

$$\mathbf{K} = (\mathbf{M}^T \mathbf{M} + \lambda * \mathbf{I})^{-1} \mathbf{M}^T \quad (2.12)$$

$$\Delta \mathbf{U}(k) = \mathbf{K}(\mathbf{Y}^{zad}(k) - \mathbf{Y}^0(k)) \quad (2.13)$$

W naszej regulacji potrzebujemy wyznaczyć tylko pierwszy element macierzy  $\Delta \mathbf{U}(k)$  czyli  $\Delta u(k|k)$ . W tym celu rozwijamy wzór do postaci:

$$\Delta u(k|k) = k_e e(k) - k_u \Delta \mathbf{U}^P - k_z \Delta \mathbf{Z}^P \quad (2.14)$$

gdzie:

$$e(k) = \mathbf{Y}^{zad}(k) - \mathbf{Y}(k) \quad (2.15)$$

$$k_e = \sum_{i=1}^N K(1, i) \quad (2.16)$$

$$k_u = k \mathbf{M}^P \quad (2.17)$$

$$k_z = k \mathbf{M}^{zP} \quad (2.18)$$

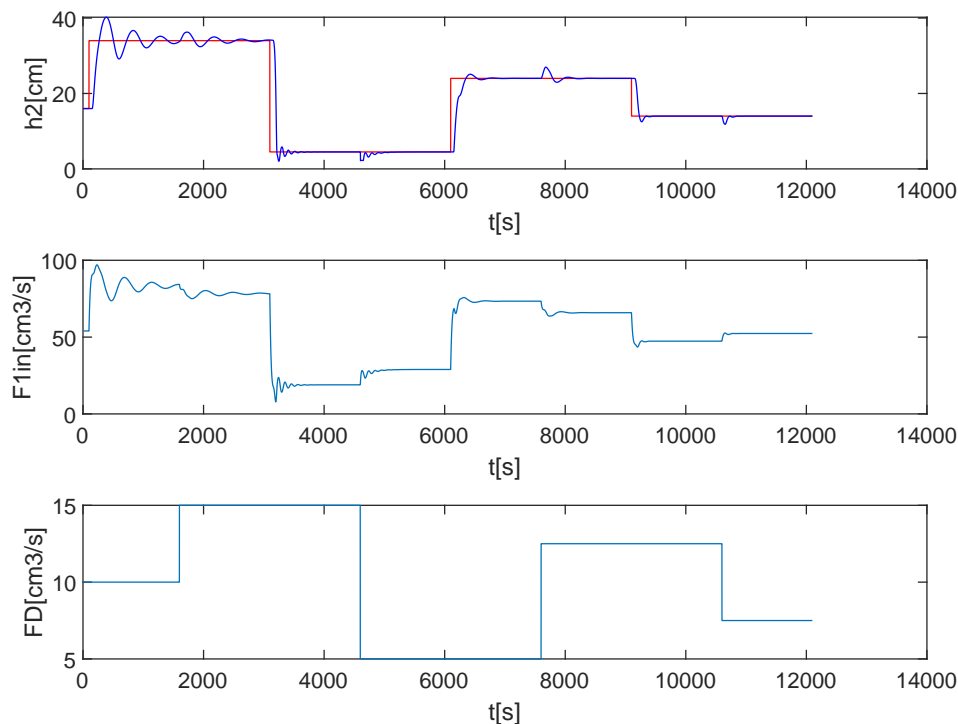
k to oznaczenie pierwszego wiersza macierzy K. Aktualne sterowanie otrzymujemy poprzez zsumowanie poprzedniego sterowania i aktualnie wyliczonego  $\Delta u(k|k)$ .

Po implementacji przystąpiliśmy do testowania regulatora. W tym celu zadaliśmy mu przebieg zakładający 4 zmiany wartości zadanej oraz 4 zmiany zakłócenia. Testowanie obiektu wymagało od nas także dobrania horyzontów oraz wartości lambda. Co do horyzontów, ponieważ zmniejszanie ich nie może poprawić regulacji pozostaliśmy przy stałej wartości równej odpowiedzi skokowej. Jeśli chodzi o lambda dla bardzo małych wartości przebiegi były nieakceptowalne. Poniżej przedstawiliśmy przebiegi regulacji dla trzech różnych lambda: 100, 2000 oraz 10000. Dla tych trzech przebiegów obliczony został średni błąd kwadratowy regulacji określony wzorem 2.19. Dla poszczególnych przebiegów błąd ten wynosił:

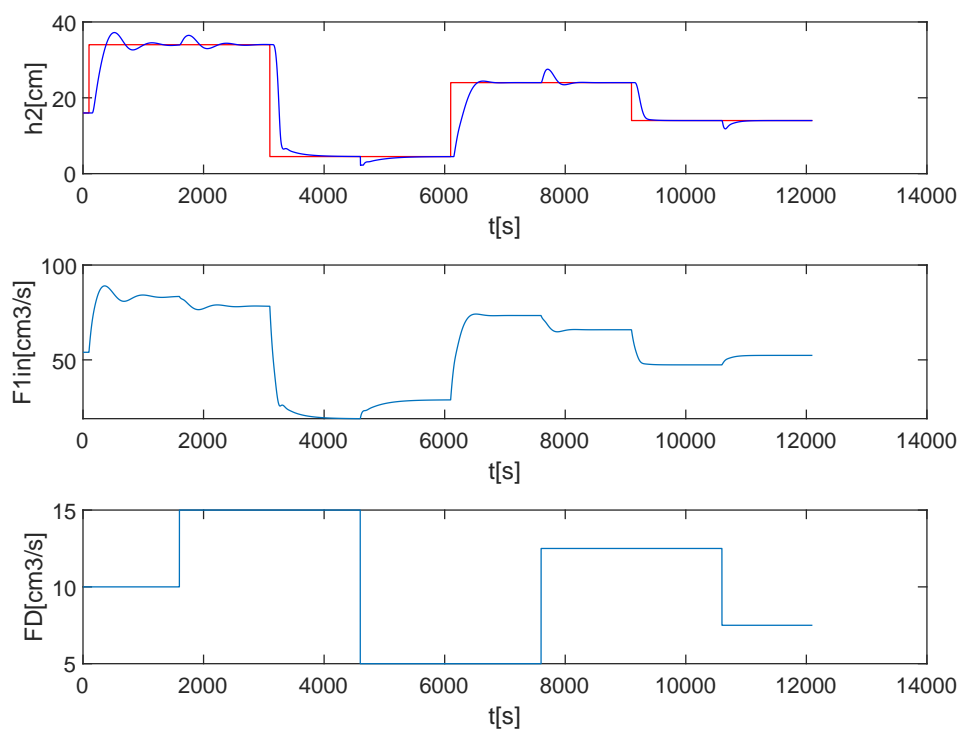
- lambda 100 - 13.2852
- lambda 2000 - 18.4502
- lambda 10000 - 27.1046

Dla lambda 100 błąd jest definitywnie najmniejszy, jednakże zarówno sterowanie jak i wyjście ulegają znacznym oscylacjom. Sytuacja dla lambda 10000 jest odwrotna, oscylacji brak, jednakże błąd wzrósł dwukrotnie. Lambda 2000 jest najlepszym rozwiązaniem z przetestowanych, oscylacje są obecne, jednakże błąd nie jest aż taki wysoki

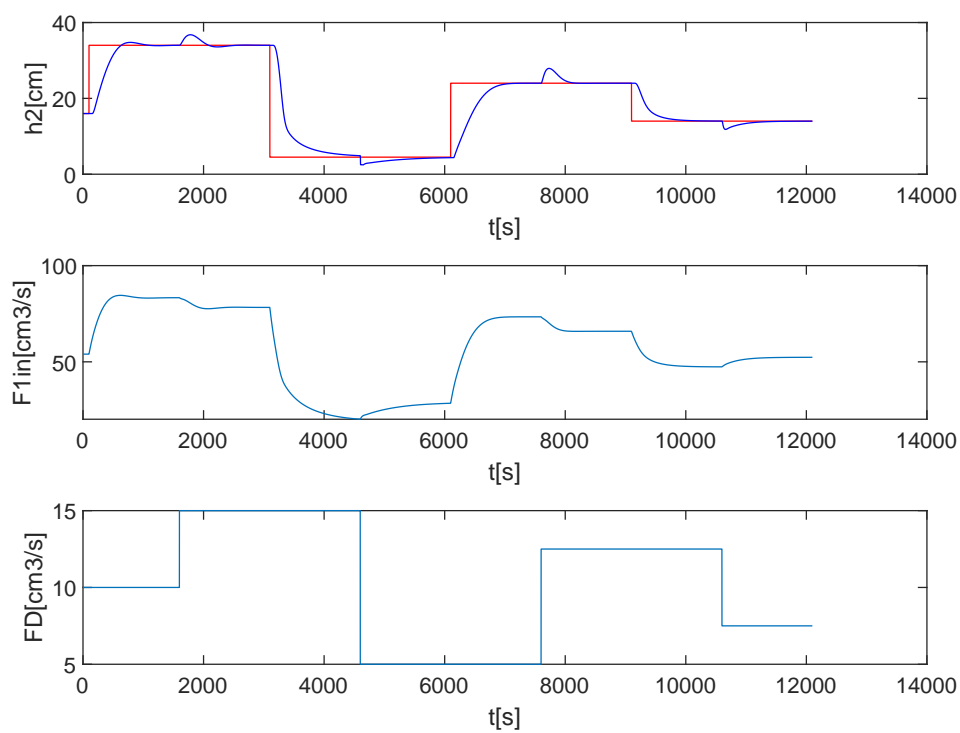
$$e = \frac{\sum_{t=1}^{t_{max}} (y_{zad}(t) - y(t))^2}{t_{max}} \quad (2.19)$$



Rys. 2.6. Przebieg regulacji dla lambda = 100



Rys. 2.7. Przebieg regulacji dla  $\lambda = 2000$



Rys. 2.8. Przebieg regulacji dla  $\lambda = 10000$

## 3. Zadanie 2

### 3.1. Obiekt rozmyty

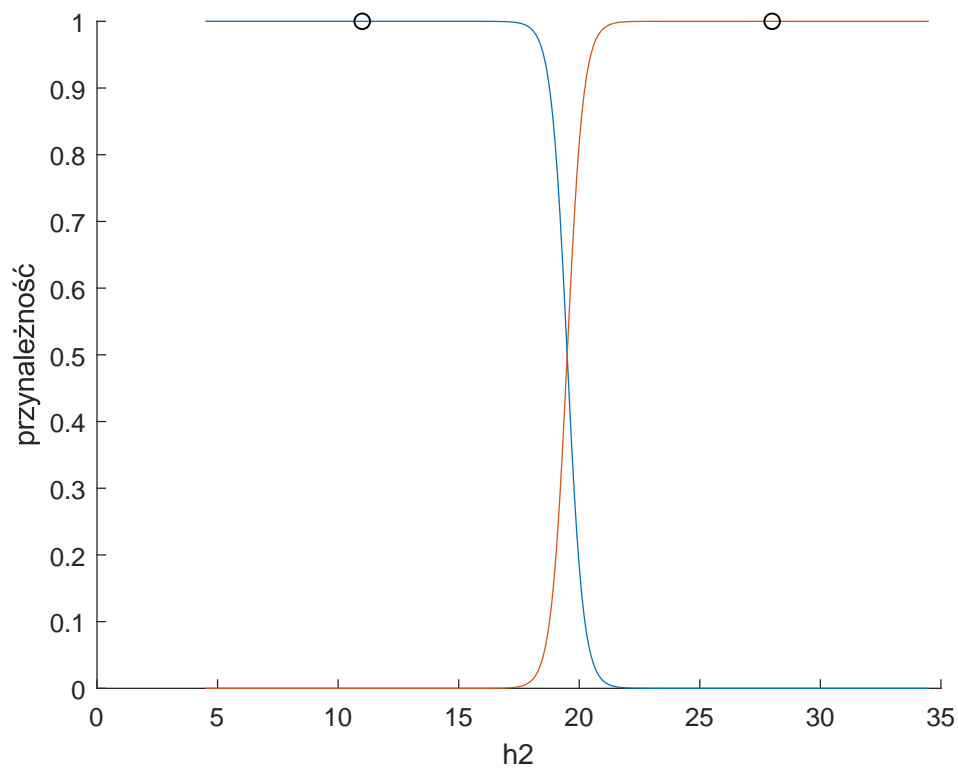
Utworzenie modelu rozmytego Takagi-Sugeno dla obiektu należy zacząć od wyznaczenia funkcji przynależności. Jako funkcji przynależności zdecydowaliśmy się użyć funkcji sigmoidalnych. Ponieważ projekt zakłada utworzenie modeli rozmytych z dwoma, trzema, czterema oraz pięcioma modelami lokalnymi, funkcje przynależności dla pierwszego,  $n$ -tego oraz ostatniego (przy  $r$  obecnych modelach lokalnych) modelu lokalnego wyglądają tak jak w równaniach od 3.1 do 3.3. W podanych wzorach zmienna  $a$  oznacza nachylenie funkcji i została przyjęta taka sama dla wszystkich modeli lokalnych. Wektor  $c$  natomiast zawiera punkty dla których funkcje przynależności sąsiadnych modeli lokalnych osiągają wartość 0.5. Wartość  $a$  została dobrana eksperymentalnie na poziomie  $a=3$ . Wartości  $c$  zostały wyliczone poprzez podzielenie zakresu wartości zmiennej na podstawie której liczony jest obiekt rozmyty przez ilość regulatorów lokalnych, co zagwarantowało w równomierny podział zbiorów rozmytych. Zdecydowaliśmy się na wykorzystanie wyjścia obiektu jako zmiennej determinującej zbiory rozmyte. Decyzja ta poparta była faktem, że dla sterowania w obiekcie występuje znaczne opóźnienie co prawdopodobnie uniemożliwiłoby późniejsze zaimplementowanie na utworzonych zbiorach rozmytych skutecznego działającego regulatora. Dla badanego w zadaniu pierwszym zakresu sterowań wyjście obiektu przyjmuje wartości w przybliżeniu od 4.5 do 34.5, dlatego właśnie w takim zakresie tworzone były zbiory rozmyte. Dodatkowo dla każdego zbioru rozmytego należało określić punkt pracy w którym zlinearyzowany był obiekt lokalny. Punkty pracy zostały wyznaczone na podstawie kształtu poszczególnych funkcji przynależności. Wartość wyjścia w punkcie pracy dla wybranego modelu równa jest w przybliżeniu wartości wyjścia znajdującej się w środku przedziału dla którego funkcja przynależności wynosi 1. Funkcje przynależności dla utworzonych obiektów rozmytych przedstawione zostały na wykresach od 3.1 do 3.4.

$$\mu(1) = 1 - \frac{1}{1 + e^{-a*(h2-c(1))}} \quad (3.1)$$

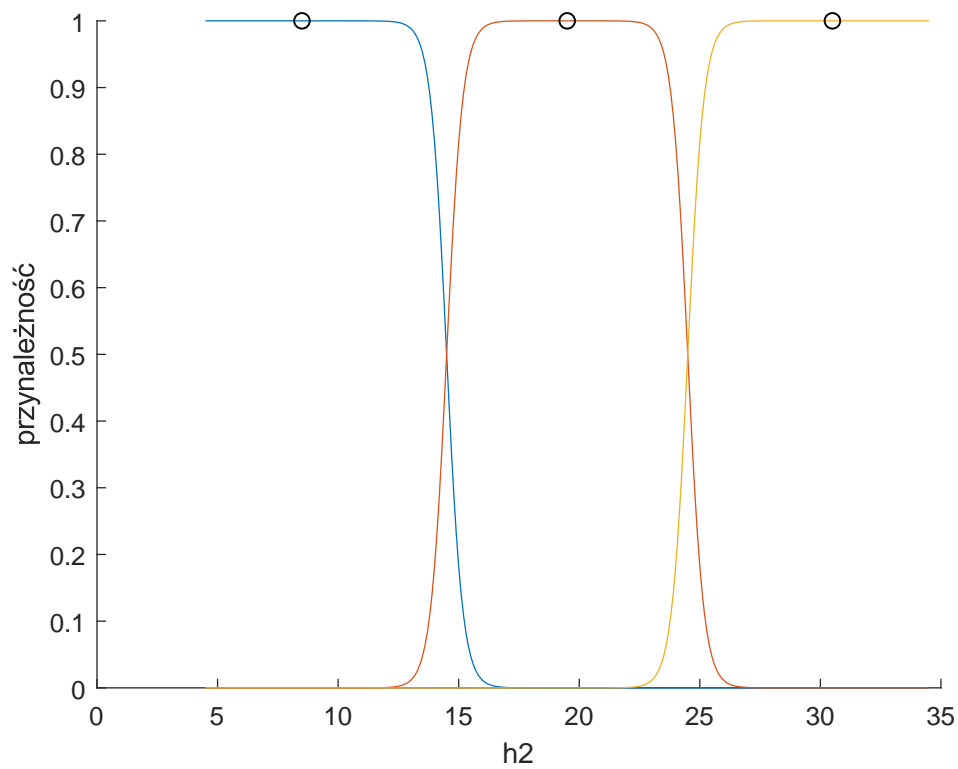
$$\mu(n) = \frac{1}{1 + e^{-a*(h2-c(n-1))}} - \frac{1}{1 + e^{-a*(h2-c(n))}} \quad (3.2)$$

$$\mu(r) = \frac{1}{1 + e^{-a*(h2-c(r-1))}} \quad (3.3)$$

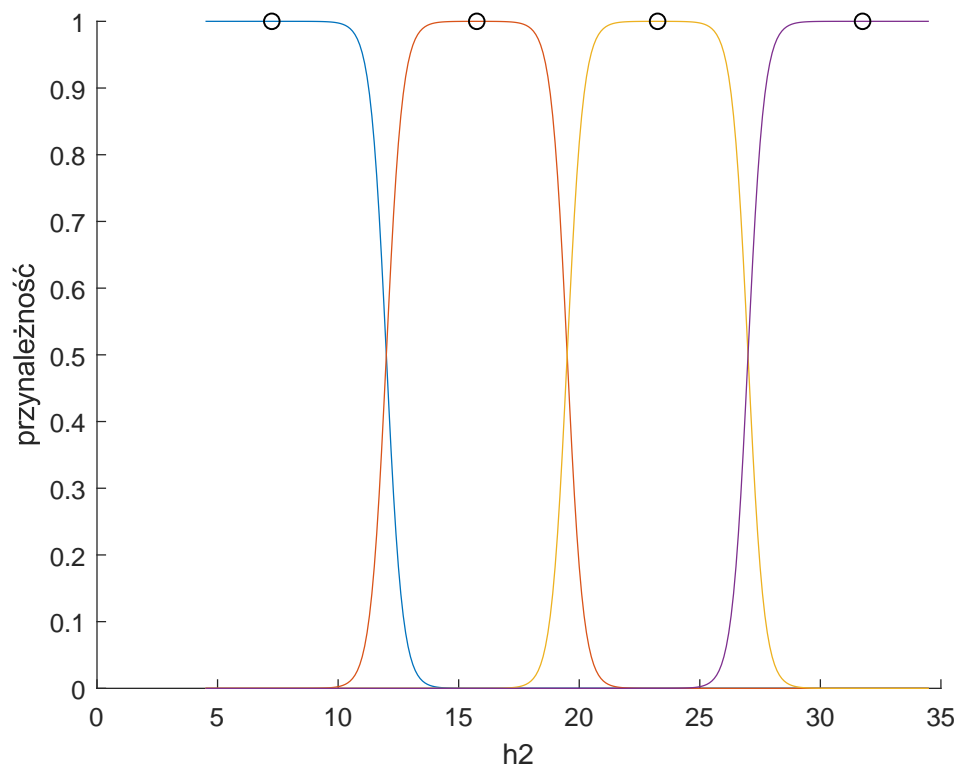
Utworzenie obiektu rozmytego na podstawie przedstawionych funkcji przynależności nie sprawiło już zbytniego problemu. Zakładając, że obiekt ma określoną ilość zbiorów rozmytych i każdy zbiór posiada swój punkt linearyzacji zaimplementowane została określona ilość obiektów liniowych. W każdej iteracji liczone są wyjścia wszystkich obiektów liniowych oraz wagi według funkcji przynależności. Następnie faktyczny stan obiektu otrzymywany jest poprzez przemnożenie wyników lokalnych przez ich wagi i podzielenie otrzymanej liczby przez sumę wag. Oczywiście dla każdego modelu liniowego potrzebny jest punkt linearyzacji, który jak już wspominałem został określony jedynie poprzez wartość wyjścia obiektu  $h_{20}$ . Wartości  $V_{10}$ ,  $V_{20}$  oraz  $h_{10}$  obliczane są analogicznie jak dla obiektu liniowego z pierwszego zadania. Co do wartości zakłócenia  $F_{D0}$  ustalona została ona na 10. Wartość sterowania w punkcie linearyzacji  $F_{10}$  można zatem wyliczyć podstawiając wartość 0 pod pochodną  $dV/dt$ , co daje nam działanie  $F_{10} = \alpha_1 * \sqrt{h_{20}} - F_{D0}$ .



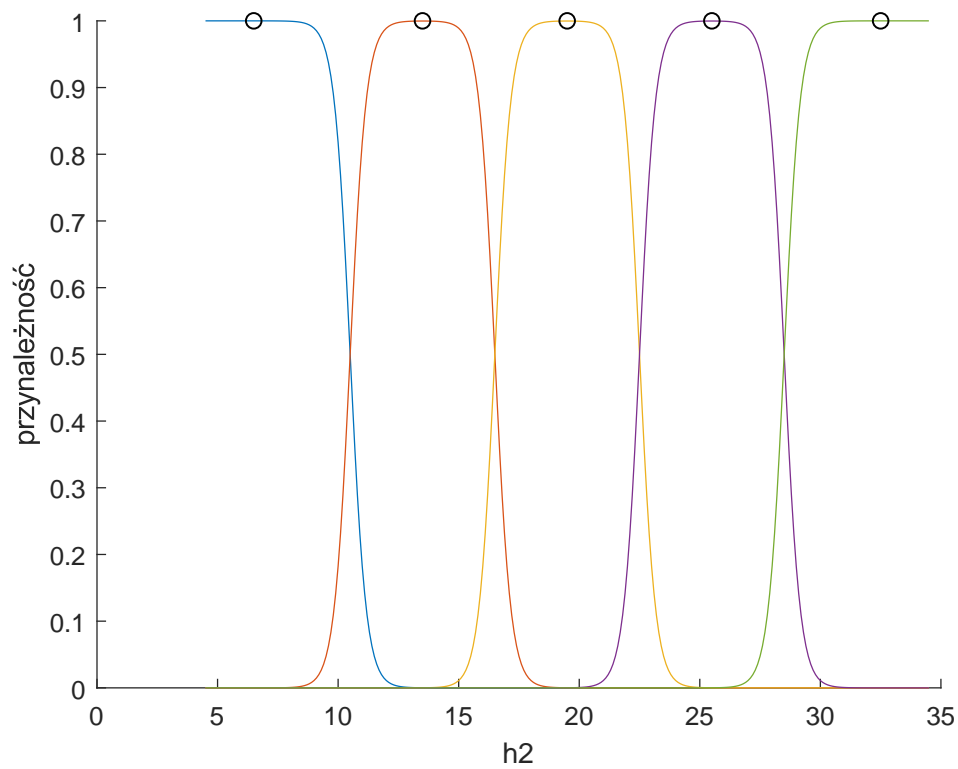
Rys. 3.1. Funkcje przynależności dla obiektu rozmytego z dwoma obiektami lokalnymi



Rys. 3.2. Funkcje przynależności dla obiektu rozmytego z trzema obiektami lokalnymi



Rys. 3.3. Funkcje przynależności dla obiektu rozmytego z czterema obiektami lokalnymi



Rys. 3.4. Funkcje przynależności dla obiektu rozmytego z pięcioma obiektami lokalnymi

Porównania działania poszczególnych modeli rozmytych modelem nieliniowym oraz liniowym obiektu przedstawione zostały na wykresach od 3.5 do 3.8. Dla każdego modelu wykonane zostały przebiegi ze skokami sterowania w kroku 100. Użyte skoki sterowania są takie same jak w zadaniu pierwszym. Na wykresach przebiegi dla oryginalnego obiektu przedstawione są kolorem niebieskim, dla zlinearyzowanego różowym, a dla rozmytego zielonym.

Dla modelu z dwoma obiektami lokalnymi działanie modelu rozmytego definitywnie nie jest zadowalające. Końcowa wartość wyjścia obiektu rozmytego w większości przypadków zauważalnie rozbiega się z końcową wartością dla oryginalnego modelu. Kształty przebiegów dla poszczególnych skoków nie są podobne do przebiegów nieliniowego ani liniowego obiektu. Przebiegiem, który najbardziej pokrywa się z oryginałem jest ten dla skoku sterowania do 34 (drugi od dołu).

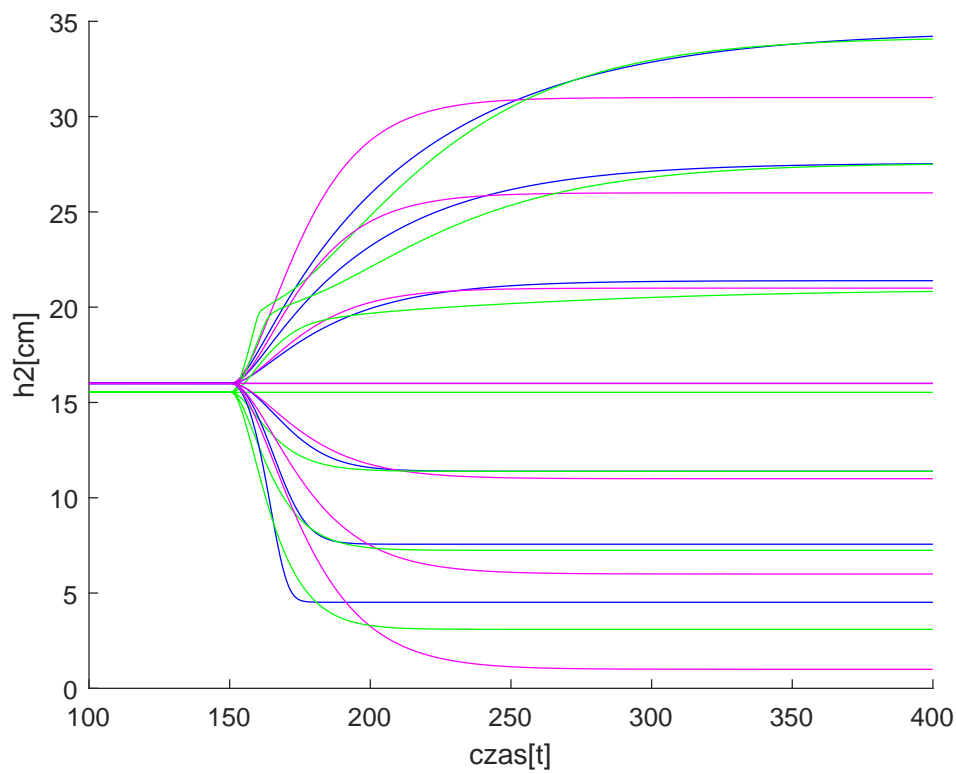
Dla modelu z trzema obiektami lokalnymi sytuacja wydaje się lepsza. Przebiegi dla skoków powyżej punktu pracy są zbliżone w wyglądzie do oryginalnego obiektu. Nie licząc przebiegu dla skoku sterowania do 24 (pierwszy od dołu) wartości końcowe wyjścia obiektu rozmytego w przybliżeniu pokrywają się z wartościami dla obiektu oryginalnego.

Dla modelu z czterema obiektami lokalnymi znowu nastąpiła nieznaczna poprawa. Przebiegi przez większość symulacji pokrywają się kształtem z przebiegami dla pierwotnego modelu, jednakże część wartości końcowych wyjścia nieznacznie odbiega od pożądanych.

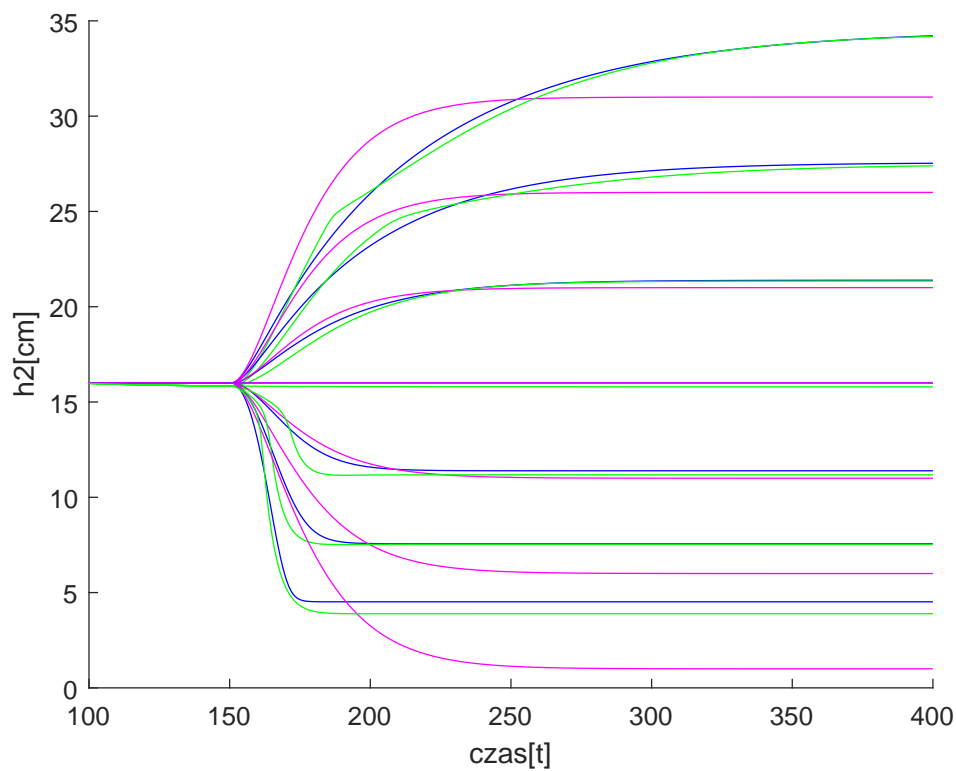
Modelem z pięcioma obiektami lokalnymi działa najlepiej z przedstawionych. Poza kilkoma miejscami przebiegi pokrywają się z przebiegami dla oryginalnego obiektu. Wartości końcowe także są w przybliżeniu równe wartościom końcowym dla obiektu oryginalnego. Obiekt ten jest zdecydowanie najlepszym z utworzonych i to właśnie on wykorzystywany będzie do dalszych badań.

Badanie to ukazuje jak przydatne mogą być modele rozmyte przy modelowaniu obiektów nieliniowych. Działanie modelu liniowego w dużej odległości od punktu pracy zbyt znacząco różniło się od działania modelu nieliniowego, by można było stosować je zamiennie. Dzięki wykorzystaniu kilku modeli liniowych sprzężonych w model rozmyty jesteśmy w stanie dosyć dokładnie odwzorować obiekt, co potencjalnie umożliwiałoby stosowanie modelu rozmytego zamiast oryginału.

Pomimo prób nie udało się znaleźć lepszego modelu rozmytego, czyli takiego, który niwelowałby znacząco wady modelu otrzymanego automatycznie (różnice w przebiegach funkcji dla skoków do niskich sterowań) nie pogarszając działania reszty modelu. Z tego powodu podtrzymaliśmy naszą decyzję o używaniu wygenerowanych dla 5 modeli lokalnych funkcji przynależności.

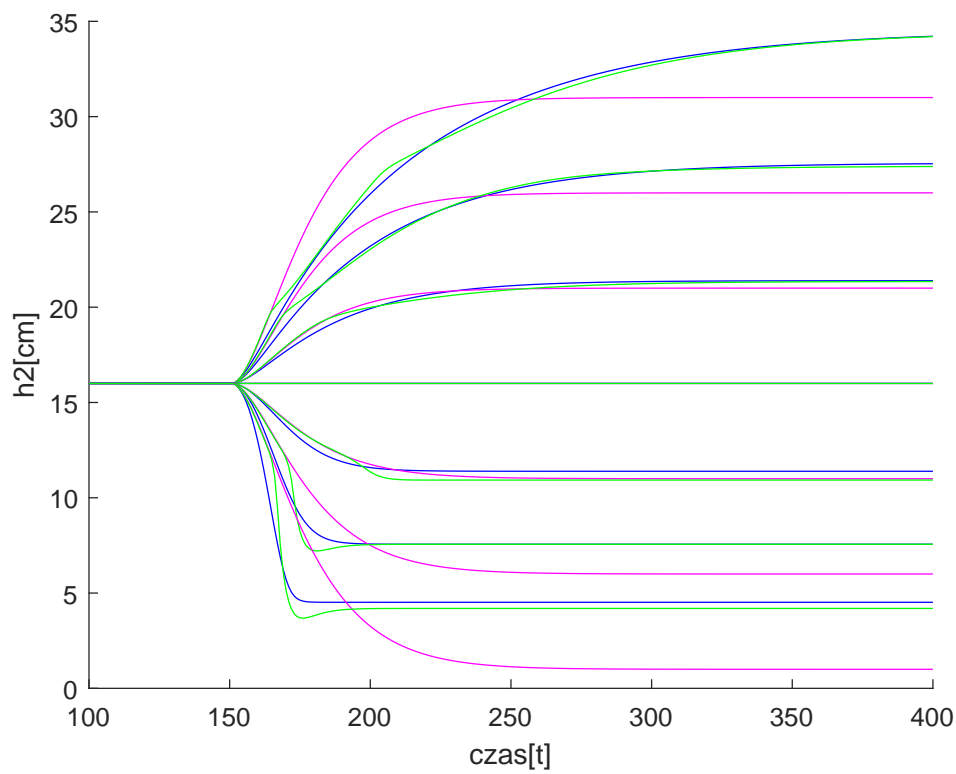


Rys. 3.5. Porównanie działania dla obiektu rozmytego z dwoma obiektami lokalnymi

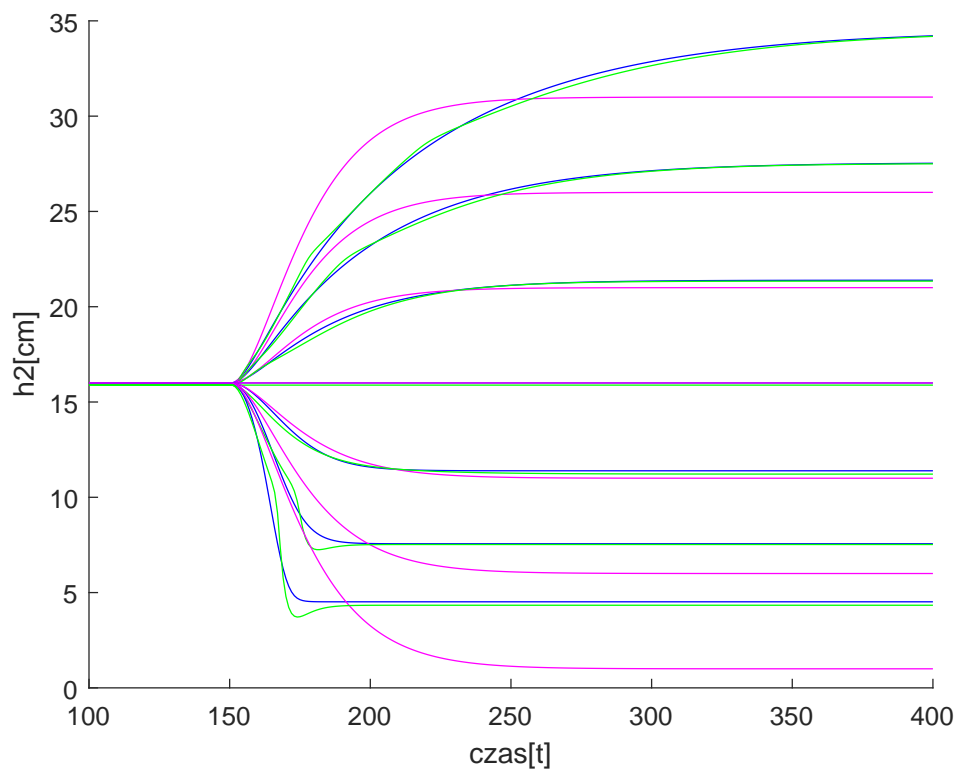


Rys. 3.6. Porównanie działania dla obiektu rozmytego z trzema obiektami lokalnymi





Rys. 3.7. Porównanie działania dla obiektu rozmytego z czterema obiektami lokalnymi



Rys. 3.8. Porównanie działania dla obiektu rozmytego z pięcioma obiektami lokalnymi

### 3.2. DMC rozmyty

Implementacja rozmytego regulatora DMC nie jest bardzo skomplikowana. W tym celu należy zaimplementować tyle regulatorów DMC ile jest zbiorów rozmytych, w naszym wypadku 5. Regulatory różnią się między sobą odpowiedziami skokowymi, które są zebrane w punktach pracy odpowiadających każdemu z modeli. W każdym kroku regulacji każdy z regulatorów lokalnych oblicza zmianę sterowania. Następnie obliczane są wagi poszczególnych zbiorów rozmytych i następuje rozmycie zmiany sterowania. Każde z wyliczonych sterowań przemnażane jest przez swoją wagę, wyniki sumuje się i dzieli przez sumę wag otrzymując w ten sposób końcową zmianę sterowania.

Dodatkowo w wymaganiach napisane było, że regulator rozmyty powinien posiadać ograniczenia. W tym celu przyjęte zostały przez nas: maksymalna wartość sterowania - 90, minimalna wartość sterowania - 20 oraz maksymalna zmiana sterowania w jednym kroku - 1. Ograniczenia te uwzględnia się dopiero po obliczeniu końcowej wartości zmiany sterowania.

Stworzony na podstawie wyznaczonych funkcji przynależności regulator rozmyty otrzymał takie same zadanie wyregulowania obiektu jak zwykły regulator w zadaniu pierwszym. Początkowo wartości wszystkich horyzontów zostały ustawione na najwyższą wartość, czyli długość odpowiedzi skokowej - 400, a lambda na wartość dla której przebieg zwykłego regulatora wydał nam się optymalny - 2000. Przebiegi regulacji obiektu dla tych parametrów zostały przedstawione na wykresie 3.9. Jak widać przebiegi są bardzo dobre, a w porównaniu z przebiegami pojedynczego DMC dla tej samej lambda wręcz genialne. Błąd regulacji także jest mniejszy i wynosi 18.2898. Obiekt dąży do zadanej wartości wyjścia bez oscylacji ani przesterowań i osiąga go dość szybko.

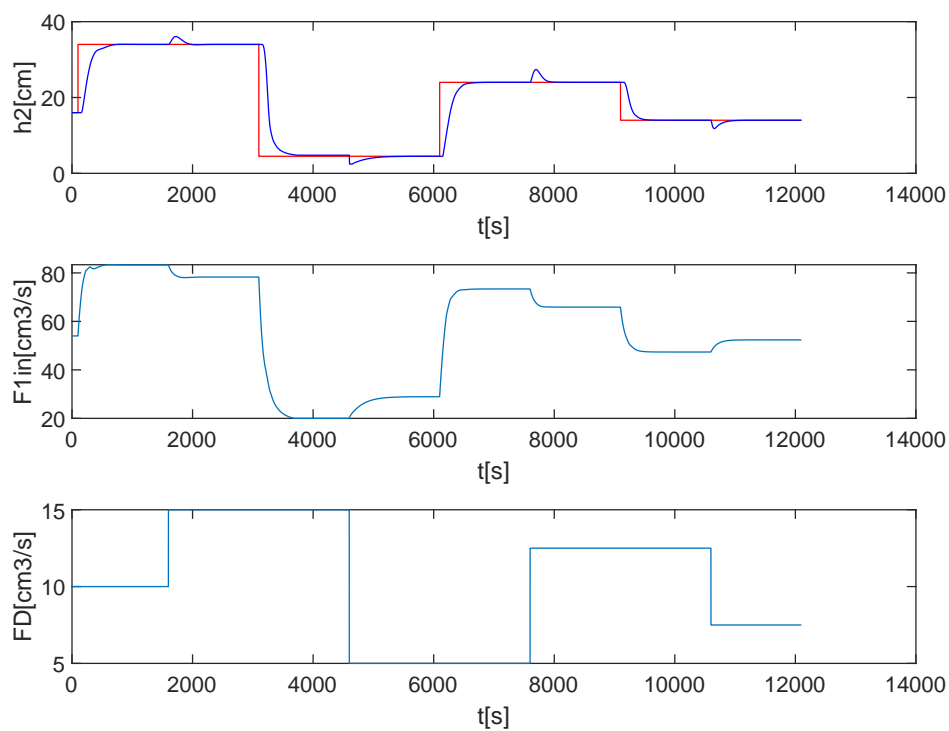
Po tak spektakularnym sukcesie oryginalnego kształtu funkcji rozpoczęliśmy eksperymenty nad jego zmianą w celu dalszej optymalizacji regulacji. Po wielu próbach przystaliśmy na kształt przedstawiony na rysunku 3.10. Przebieg regulacji dla tak skonstruowanych zbiorów rozmytych przedstawiony jest na wykresie 3.11. Jak widać nie uległ on znaczącej zmianie. Błąd regulacji dla zmienionej funkcji przynależności wyniósł jednak nieco mniej, a dokładnie 18.2504.

Po zmianie właściwości rozmycia zdecydowaliśmy się jeszcze sprawdzić działanie dla mniejszych wartości lambda, a dokładnie dla lambda równego 100, dla którego oryginalny DMC nie radził sobie dobrze z regulacją, powodując gasnące oscylacje dla wysokich skoków wartości zadanej. Przebieg dla zmniejszonej lambda przedstawiony został na wykresie 3.12. Jak widać uległ on nieco pogorszeniu, zaczęły występować niewielkie przesterowania oraz pojedyncze oscylacje. Z drugiej strony błąd sterowania zmalał prawie o jedną trzecią swojej dotychczasowej wartości, aż do 13.6096. Zdecydowaliśmy, że niewielkie pogorszenie przebiegów jest małą ceną za znacznie niższy błąd i pozostaliśmy przy zmienionej wartości lambda.

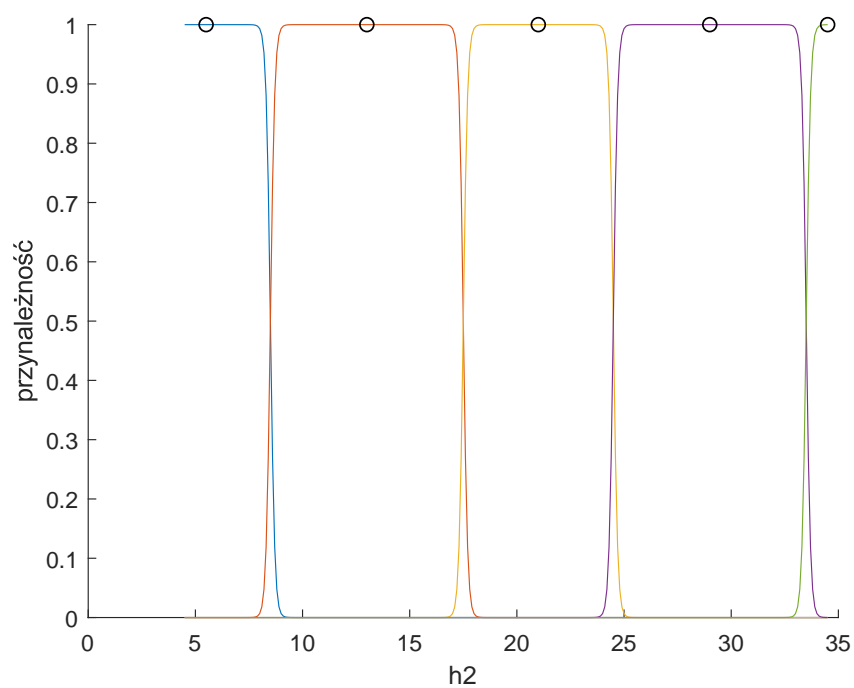
Ostatnią rzeczą jaką regulowaliśmy dla zaimplementowanego regulatora rozmytego jest zmiana długości horyzontów. Zdecydowaliśmy się na ten krok, choć zazwyczaj nie przynosi on poprawy regulacji, po części ze względu na kolejne zadania projektu, w których wymagane jest używanie algorytmów optymalizacji, znacznie spowalnianych przez dużą złożoność obliczeniową. Horyzonty skracane były przez nas do momentu, gdy błąd regulacji nie ulegał pogorszeniu. W ten sposób doszliśmy do następujących wartości horyzontów:  $D = 200$ ,  $N = 150$ ,  $N_u = 100$ ,  $D_z = 200$ . Przebieg regulacji dla zmniejszonych horyzontów znajduje się na wykresie 3.13. Jak widać nie ma w nim widocznych zmian. Co ciekawe błąd regulacji dla skróconych horyzontów uległ nieznacznej poprawie, do poziomu 13.5395. Nie widząc żadnych przeciwwskazań zdecydowaliśmy o dalszym używaniu skróconych horyzontów w kolejnych zadaniach.

W wykonanych w tym zadaniu przebiegach regulacji można zauważyć działanie ograniczenia wartości sterowania. Jeśli dokładnie przyjrzeć się wykresom zauważalne jest, że ustabilizowana wartość wyjścia obiektu dla czasu w okolicy 4000s jest nieznacznie większa od zadanej. Dzieje się tak, ponieważ sterowanie osiąga swoją wartość minimalną, co spowodowane jest równoczesną obecnością dużego zakłócenia. Przy wystąpieniu spadku zakłócenia w chwili 4600, regulatorowi po chwili udaje się bez problemu osiągnąć zadaną wartość. Ponieważ zachowanie to umożliwia

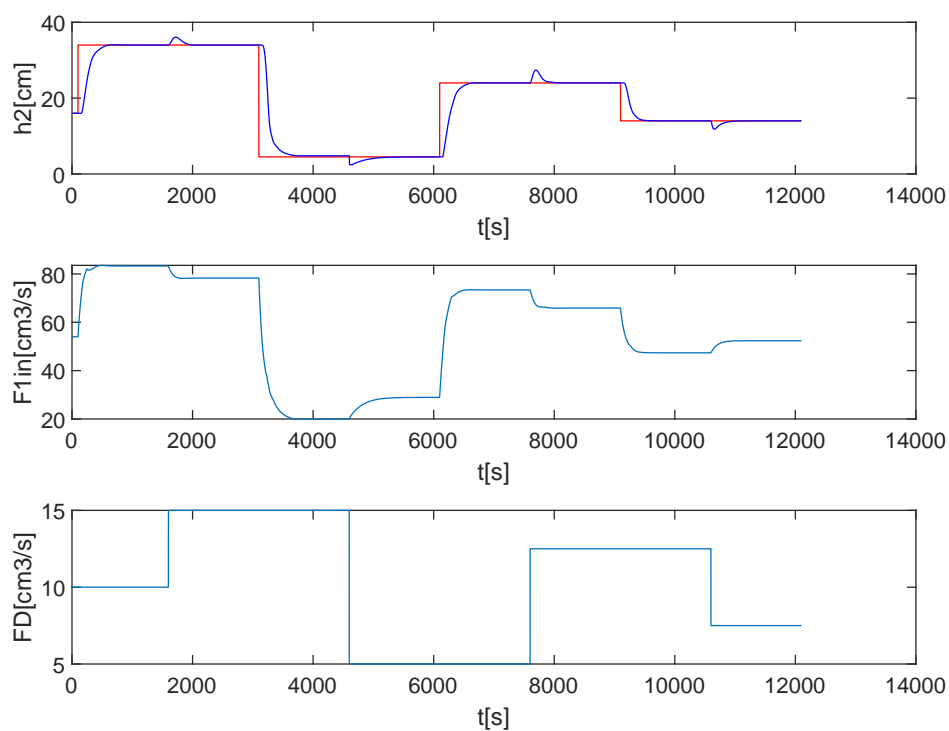
zaobserwowanie zaimplementowanego ograniczenia nie zostało ono usunięte w kolejnych zadaniach, co mogłoby być wykonane dzięki nieznacznemu zwiększeniu dolnej granicy sterowania.



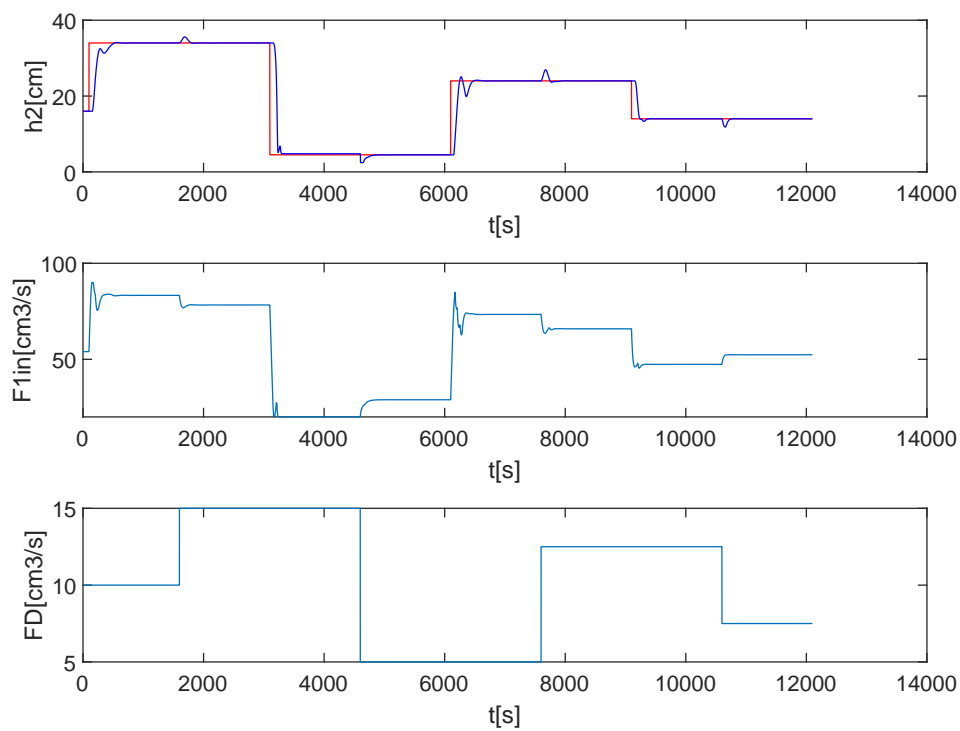
Rys. 3.9. Przebieg regulacji dla DMC rozmytego przy lambda 2000



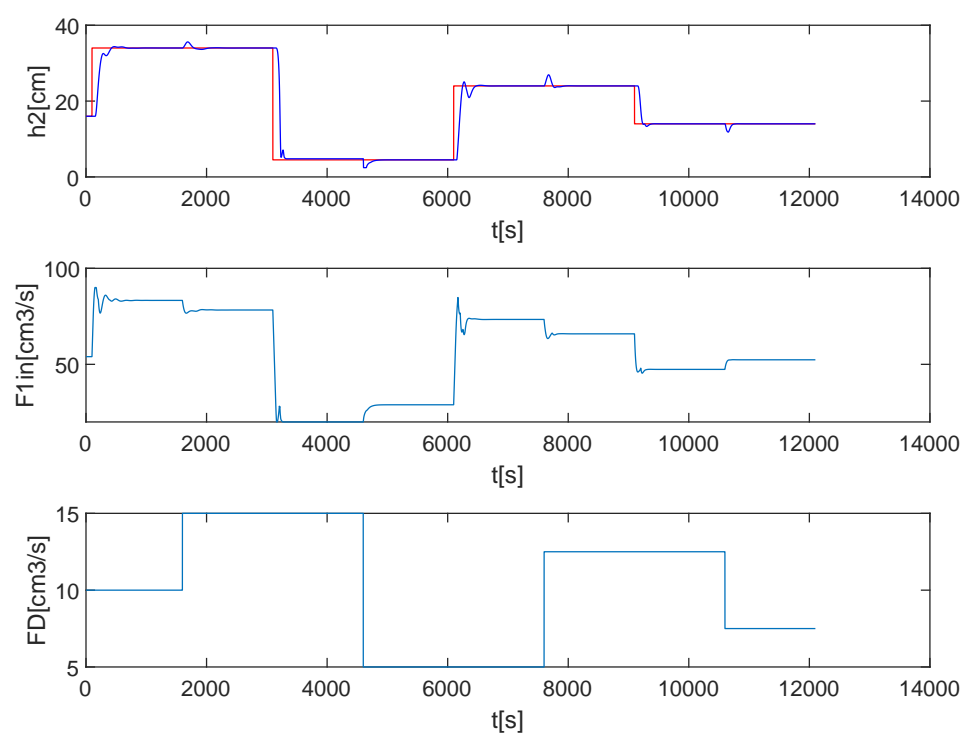
Rys. 3.10. Zmodyfikowane funkcje przynależności



Rys. 3.11. Przebieg regulacji dla DMC rozmytego przy lambda 2000 i zmodyfikowanych funkcjach przynależności



Rys. 3.12. Przebieg regulacji dla DMC rozmytego przy lambda 100



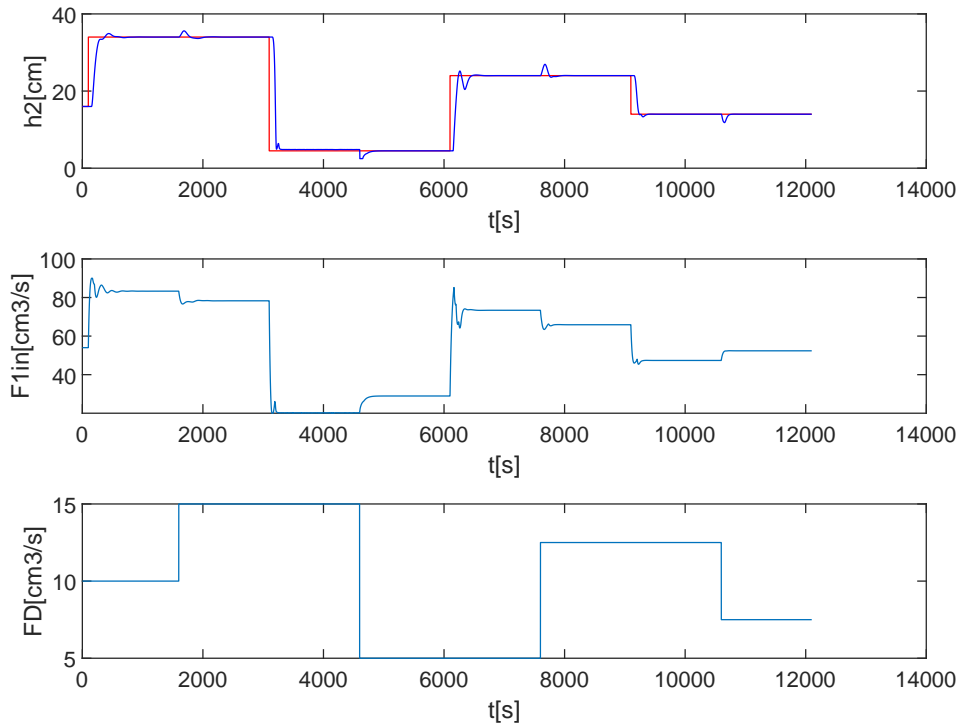
Rys. 3.13. Przebieg regulacji dla DMC rozmytego przy  $\lambda = 100$  i skróconych horyzontach

## 4. Numeryczny rozmyty regulator SL

Rozmyty regulator typu SL jest w pewnym sensie podobny do rozmytego regulatora DMC, z dokładnością do tego, kiedy następuje rozmycie. Otóż w regulatorze SL rozmywana jest nie końcowa zmiana sterowania, a odpowiedzi skokowe regulatorów. W każdym kroku regulacji na podstawie poszczególnych odpowiedzi skokowych oraz aktualnych wag obliczana jest wspólna odpowiedź skokowa, a następnie następuje obliczanie nowych macierzy  $M$ ,  $M^P$  i  $M^{zP}$  na podstawie których obliczana jest rozmyta odpowiedź swobodna obiektu, wykorzystywana do regulacji.

Ponieważ wspomniany rozmyty algorytm SL miał zostać zaimplementowany w formie numerycznej zmienił się także sposób obliczania zmiany sterowania. Jest ona obliczana poprzez rozwiązanie problemu optymalizacji liniowo-kwadratowej przedstawionego wzorem 4.1, w którym  $\tilde{y}^0$  oznacza rozmytą odpowiedź swobodną, a  $M_k$  rozmytą macierz  $M$ . Problem ten w naszej implementacji rozwiązujemy za pomocą wbudowanej w Matlab funkcji minimalizacji  $fmincon()$ . W funkcji tej bierzemy także pod uwagę ograniczenia co do wielkości maksymalnej oraz minimalnej sterowania. Ograniczenia odnośnie maksymalnej zmiany sterowania w kroku po konsultacji z prowadzącym nie zostały wzięte pod uwagę.

$$\min_{\Delta u} \{ (y^{zad} - \tilde{y}^0 - M_k * \Delta U)^T (y^{zad} - \tilde{y}^0 - M_k * \Delta U) - \lambda \Delta u^T * \Delta u \} \quad (4.1)$$



Rys. 4.1. Przebieg regulacji dla numerycznego SL rozmytego

Podczas sterowania regulatorem SL długości horyzontów,  $\lambda$  oraz kształt funkcji przynależności przyjęte zostały takie same jak dla ostatniego przebiegu rozmytego regulatora DMC w zadaniu 2, czyli  $\lambda$  równa 100, D równe 200, N równe 150,  $N_u$  równe 100,  $D_z$  równe 200 oraz funkcje przynależności tak jak na wykresie 3.10.

Przebieg regulacji umieszczony został na wykresie 4.1. Przebieg jest bardzo podobny do tego uzyskanego z rozmytego regulatora DMC, nie licząc okolic pierwszego skoku wartości zadanej wykresy praktycznie się nie różnią. Przy pierwszym skoku wartości zadanej algorytm SL działa jednakże nieco bardziej płynnie niż DMC. Przy regulacji DMC obiekt zwalnia przed dotarciem do wartości zadanej w formie pojedynczej oscylacji po czym dociera do niej praktycznie bez przeregulowania. Dla SL wykres jest płynniejszy, ale występuje wyraźne przekroczenie wartości zadanej. Błąd dla algorytmu SL wynosi jedynie 12.2571, więc jest zdecydowanie mniejszy niż błąd dla DMC. Oznacza to, że mimo wszystko wystąpiła poprawa regulacji.

## 5. Numeryczny rozmyty regulator NPL

Rozmyty regulator numeryczny NPL różni się od regulatora SL sposobem liczenia aktualnej odpowiedzi swobodnej. Wszystkie elementy odpowiedzi swobodnej liczy się rekurencyjnie za pomocą wzorów od 5.1 do 5.3, przy czym przyszłe odpowiedzi modelu  $y_{k+i}^M$  liczy się rekurencyjnie z modelu rozmytego obiektu, zakładając przyszłe przyrosty sterowania równe 0. Dalsza praca regulatora postępuje tak jak dla algorytmu SL, włącznie z obliczaniem zmiany sterowania za pomocą funkcji optymalizacji.

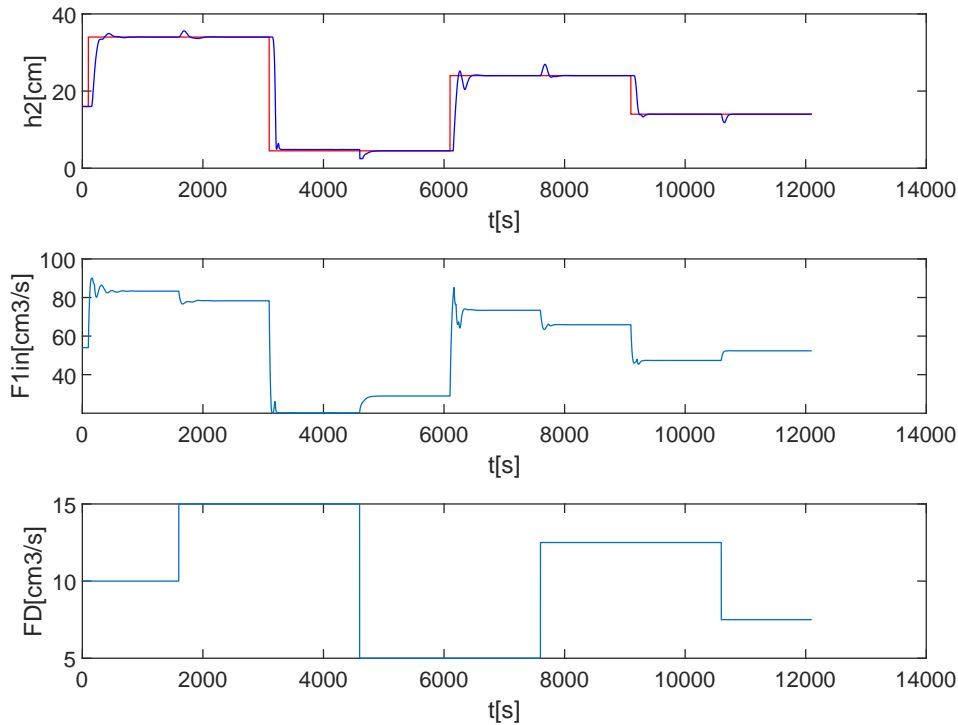
12.3832

$$y_{k+i|k}^0 = y_{k+i}^M + dk; \quad (5.1)$$

$$d_k = y_k - \sum_{i=1}^{D-1} \tilde{s}_i * \Delta u_{k-i} - \tilde{s}_D * u_{k-D} \quad (5.2)$$

$$y_k^M = \sum_{i=1}^{D-1} \tilde{s}_i * \Delta u_{k-i} + \tilde{s}_D * u_{k-D} \quad (5.3)$$

Podczas pracy regulatora wartości horyzontów, lambda oraz kształt funkcji przynależności przyjęte zostały takie jak dla regulatora SL.



Rys. 5.1. Przebieg regulacji dla numerycznego SL rozmytego



Przebieg regulacji dla rozmytego regulatora numerycznego NPL przedstawiony został na wykresie 5.1. Porównując tenże wykres, z wykresem dla regulatora SL można stwierdzić, że przebiegi dla obydwu regulatorów są praktycznie identyczne i nie sposób jest znaleźć jakąkolwiek różnicę. Jeśli chodzi o porównanie przebiegu NPL oraz DMC, to sprowadza się one do jednej różnicy w działaniu opisanej już w poprzednim zadaniu. Błąd obliczony przy stosowaniu regulatora NPL wynosił 12.3832, tak więc nieco więcej niż przy regulowaniu SL, ale znacznie mniej niż dla DMC. Na podstawie powyższych spostrzeżeń nie można niestety stwierdzić czy użycie regulatora NPL jest lepsze czy gorsze od użycia regulatora SL. Wykresy są identyczne pod kątem wizualnym, a różnica w błędach niewielka. Jeśli chodzi o porównanie NPL oraz DMC, NPL wydaje się być nieco lepsze.

## 6. Skrypty

W celu uzyskania wykresów zamieszczonych w sprawozdaniu należy wywołać podane skrypty:

### 6.1. Zadanie 1

- badanie modelu:  
`z1_model`
- porównanie modelu nieliniowego i liniowego:  
`z1_modellin`
- odpowiedź skokowa w punkcie pracy:  
`z1_step(16, true)`
- przebiegi regulacji dla DMC:  
`z1_dmc(400, 400, 400, 400, 100, true)`  
`z1_dmc(400, 400, 400, 400, 1000, true)`  
`z1_dmc(400, 400, 400, 400, 10000, true)`

### 6.2. Zadanie 2

- obiekty rozmyte:  
`z2_modelroz(2, true)`  
`z2_modelroz(3, true)`  
`z2_modelroz(4, true)`  
`z2_modelroz(5, true)`
- dmc rozmyty, lambda 2000, oryginalne zbiory rozmyte:  
`z2_dmcroz(400, 400, 400, 400, 2000, [], [], true)`
- dmc rozmyty, lambda 2000, nowe zbiory rozmyte:  
`z2_dmcroz(400, 400, 400, 400, 2000, 10, [8.5000 17.5000 24.5000 33.5000], true)`
- dmc rozmyty, lambda 100, nowe zbiory rozmyte:  
`z2_dmcroz(400, 400, 400, 400, 100, 10, [8.5000 17.5000 24.5000 33.5000], true)`
- dmc rozmyty, lambda 100, nowe zbiory rozmyte, skrócone horyzonty:  
`z2_dmcroz(200, 150, 100, 200, 100, 10, [8.5000 17.5000 24.5000 33.5000], true)`

### 6.3. Zadanie 3

- uruchomienie regulatora SL:  
`z3_sl(200, 150, 100, 200, 100, true)`

### 6.4. Zadanie 4

- uruchomienie regulatora NPL:  
`z4_npl(200, 150, 100, 200, 100, true)`