

POLITECHNIKA POZNAŃSKA  
WYDZIAŁ ELEKTRYCZNY

---

KOMPUTEROWO WSPOMAGANE PROJEKTOWANIE  
REGULATORÓW PRĘDKOŚCI I POŁOŻENIA  
W PRZEKSZTAŁTNIKOWYCH UKŁADACH NAPĘDOWYCH

Łukasz Matecki

Praca magisterska  
napisana pod kierunkiem  
dra inż. Jana Deskura

Koreferent: dr hab. inż.  
Roman Muszyński

Poznań, czerwiec 2005

Chciałbym podziękować dr. inż. Janowi Deskurowi za czas i zaangażowanie w pisanie tej pracy oraz za rzeczową krytykę, która w sposób znaczący wpłynęła na logiczną budowę i czytelność programów.

## SPIS TREŚCI

<b>1. WSTĘP I UZASADNIENIE TEMATU PRACY.....</b>	<b>6</b>
<b>2. PRZEGLĄD NARZĘDZI PROGRAMISTYCZNYCH DO PROJEKTOWANIA REGULATORÓW NA PRZYKŁADZIE PROGRAMÓW MATLAB/SIMULINK I VISSIM.....</b>	<b>8</b>
<b>3. PAKIET OPROGRAMOWANIA DO SYNTEZY I SYMULACJI REGULATORÓW.....</b>	<b>14</b>
3.1. ZAŁOŻENIA ZWIĄZANE Z BUDOWĄ I FUNKCJAMI PROGRAMÓW. ....	14
3.2. PREZENTACJA PROGRAMÓW WCHODZĄCYCH W SKŁAD PAKIETU. ....	15
<b>4. SZCZEGÓŁOWY OPIS FUNKCJI I BUDOWY WSZYSTKICH ELEMENTÓW OPROGRAMOWANIA.....</b>	<b>20</b>
4.1. STRUKTURA PLIKÓW I KATALOGÓW. ....	20
4.2. STRUKTURY DANYCH. ....	24
4.2.1. <i>Kreator symulacji.</i> ....	24
4.2.2. <i>Program Oscyloskop.</i> ....	27
<b>5. INSTRUKCJA UŻYTKOWNIKA.....</b>	<b>30</b>
5.1. PARAMETRY NAPĘDU. ....	30
5.2. PARAMETRY OBLICZONE. ....	31
5.3. SYMULACJE. ....	33
5.4. OBLICZONE WYNIKI.....	35
5.5. PROGRAM OSCYLOSKOP.....	37
5.6. BŁOK ZAPISZ DO PLIKU. ....	42
5.7. BŁOK PRZEŁĄCZANE WZMOCNIENIE ....	44
<b>6. ZASTOSOWANIE OPROGRAMOWANIA DO PROJEKTOWANIA ODPORNEGO REGULATORA SERWONAPĘDU.....</b>	<b>46</b>
6.1. KONCEPCJA ODPORNEGO REGULATORA PRĘDKOŚCI I POŁOŻENIA DLA SERWONAPĘDU. ....	46
6.2. ALGORYTM PROJEKTOWANIA REGULATORA. ....	48
6.3. PRZYGOTOWANIE SYMULACJI. ....	49
6.4. OBLICZENIA REGULATORA. ....	50
6.5. ZESTAWIENIE DOSTĘPNYCH EKSPERYMENTÓW. ....	55
6.6. PRZEPROWADZENIE PRZYKŁADOWYCH SYMULACJI. ....	56
6.6.1. <i>Pomiar czasu ustalania położenia z dokładnością 5 % i 0.1%;</i> ....	57
6.6.2. <i>Sprawdzenie spadku wzmocnienia w pętli prędkości.</i> .....	58
6.6.3. <i>Badanie stabilności pętli regulacji przyspieszenia.</i> ....	60
6.7. ZEBRANIE WYNIKÓW I PODSUMOWANIE. ....	62
<b>7. PODSUMOWANIE.....</b>	<b>65</b>
<b>SPIS ILUSTRACJI I TABEL .....</b>	<b>67</b>
<b>BIBLIOGRAFIA.....</b>	<b>68</b>

<b>BIBLIOGRAFIA</b> .....	<b>68</b>
<b>ZAŁĄCZNIK</b> .....	<b>69</b>
PLIK WPROWADZANIA DANYCH.....	69
PLIK OBLICZEŃ NASTAW REGULATORA. ....	69
PLIK OBLICZEŃ WYNIKÓW – WSKAŹNIKÓW JAKOŚCI. ....	72
<b>ZAWARTOŚĆ PŁYTY CD</b> .....	<b>75</b>

## **Streszczenie.**

Praca obejmuje zagadnienia zastosowania nowoczesnego oprogramowania obliczeniowo – symulacyjnego do wspomagania projektowania regulatorów dla serwonapędów. W pracy zaproponowano dyskretny model regulatora prędkości i położenia oraz prosty, również dyskretny model silnika. Oprócz modelu symulacyjnego zaprezentowana została metoda zautomatyzowanego obliczania nastaw regulatora odpornego na zmianę momentu bezwładności i stałej momentu silnika.

Treść pracy podzielono w następujący sposób: wstęp i uzasadnienie tematu pracy znajduje się w rozdziale 1. W rozdziale 2. znalazł się skrótowy przegląd dostępnych narzędzi projektowych dotyczących regulatorów. Rozdziały 3., 4. i 5. zawierają obszerny opis stworzonego przez autora oprogramowania do projektowania, symulowania i analizowania jakości działania regulatorów. Rozdział 6. to przykład praktycznego zastosowania oprogramowania do zaprojektowania odpornego regulatora prędkości i położenia dla serwonapędu z synchronicznym silnikiem o magnesach trwałych. Ostatni, 7. rozdział stanowi podsumowanie pracy.

## **Summary.**

This text covers usege of modern computational –simulation software in computer aided design of controllers for servo drives. Simple discrete motor and position/speed controller models are proposed. Besides simulation model, automated parameters calculation method for motor interia and torque constant change-robust controller is presented.

Contets of this paper splits as follows: chapter 1. – introduction. Chapter 2. – brief review of available controller design tools. Chapters 3., 4. and 5. contain extensive description of software created by author which includes software for designing, simulating and quality analysys of controllers. Chapter 6 consists of practical example of usage in designing robust speed and position regulator for servo drive witch synchronous, permanent magnet motor. Summary is presented in chapter 7.

# 1. Wstęp i uzasadnienie tematu pracy.

We współczesnych napędach elektrycznych w sposób wyraźny zaznacza się postęp w dziedzinie elektroniki (energoelektroniki) i techniki mikroprocesorowej. Do fizycznej realizacji regulatorów wykorzystywane są wydajne obliczeniowo procesory sygnałowe (ang. DSP – Digital Signal Processor), precyzyjne enkodery, dające rozdzielczości rzędu miliona impulsów na obrót, moduły mocy o dobrych osiągnięciach (zarówno pod względem dopuszczalnej mocy, dynamiki jak i sprawności energetycznej) i inne zaawansowane technologicznie elementy.

Powszechność wymienionych środków daje projektantowi regulatorów duże możliwości i swobodę wyboru algorytmów. Z drugiej strony istnieje potrzeba symulacyjnego sprawdzenia poprawności rozwiązań, struktur i parametrów regulatorów. Symulacje dają konstruktorowi możliwość pracy poza laboratorium i ograniczają koszty związane z wdrożeniem nowego rozwiązania. Oczywiście końcowym stadium projektowania musi być zawsze laboratoryjne zweryfikowanie rozwiązań uzyskanych na drodze symulacji.

Podczas projektowania regulatorów, projektant spotyka się z wieloma czynnikami, które muszą być wzięte pod uwagę podczas ustalania struktury układu i jego parametrów. Poruszane w pracy kwestie związane są przede wszystkim z budową cyfrowego regulatora prędkości i położenia i nie obejmują innych zagadnień związanych z budową kompletnego napędu: regulacją prądu, zakłóceń w torach pomiarowych, problemów związanych z niedoskonałością konstrukcyjną silnika i innych.

Dla uzyskania dobrych wyników, należy wziąć pod uwagę następujące czynniki, degradujące jakość działania układu regulacji prędkości/położenia w układach napędów elektrycznych:

- dokładność obliczania położenia (a za czym idzie również prędkości i przyspieszenia) jest ograniczona przez skończoną rozdzielczość enkodera umieszczonego na wale silnika;
- cyfrowe różniczkowanie sygnału położenia, które ma na celu uzyskanie sygnału prędkości (a po kolejnym różniczkowaniu również przyspieszenia) niesie ze sobą powstawanie szumów kwantyzacyjnych, których obecność w szczególnie przykry sposób objawia się w postaci tętnień składowej czynnej prądu silnika, co pociąga za sobą również tętnienia momentu;
- sprężystość układu napędowego związanego z silnikiem utrudnia, a czasami uniemożliwia (rezonans mechaniczny) dokładne pozycjonowanie;
- wahania wartości parametrów napędu (przede wszystkim momentu bezwładności oraz stałej momentu) wymagają aby regulator był odporny – niewrażliwy lub mało wrażliwy na zmiany tych parametrów.

Efektorem wysokich wymagań stawianych regulatorom oraz dostępności oprogramowania symulacyjnego i obliczeniowego jest powstanie narzędzi do komputerowego wspomaganie projektowania regulatorów. W niniejszej pracy autor spróbuje pokrótce przedstawić popularne

programy do projektowania i symulowania układów regulacji automatycznej. Pod uwagę wzięto dwa środowiska służące obliczeniom inżynierskim – Matlab 6.5 / Simulink 5.0 oraz Vissim 5.0b. W następnej kolejności podjęto próbę stworzenia oprogramowania dedykowanego dla projektowania odpornych regulatorów prędkości i położenia. Oprogramowanie to zostało stworzone w Matlabie na podobieństwo narzędzi przygotowanych w Vissimie [7]. Na powstałe narzędzia składają się w głównej mierze dwa moduły. Pierwszy moduł to elastyczny kreator służący kompleksowemu projektowaniu regulatorów (począwszy od wprowadzenia parametrów napędu, poprzez obliczenia nastaw regulatora aż do przeprowadzania symulacji i obliczenia wyników i wskaźników jakości działania). W trakcie pisania pracy okazało się, że program musi mieć możliwość dopasowania do zmieniających się algorytmów, nazewnictwa i innych zmian. Dlatego też, zastosowano szkieletową strukturę, którą można wypełniać – w zależności od potrzeb – różną zawartością. Drugim modulem jest wirtualny oscyloskop służący analizie czasowej uzyskanych wyników.

Ostatnim, być może najważniejszym elementem pracy będzie przykład zastosowania stworzonych narzędzi do implementacji konkretnej metody projektowania odpornych regulatorów dla serwonapędu. Podwaliny teoretyczne tej części pracy zaczerpnięto w głównej mierze z następujących prac dra inż. Jana Deskura: „Odporny regulator PID serwonapędu”, IX Sympozjum Energoelektronika w Nauce i Dydaktyce, Poznań 20 – 22 września 2004 ([1]), „Dobór nastaw regulatora PID odpornego na zakłócenia” ([2]); materiały niepublikowane autora, 2004 oraz „Praktyczna metoda projektowania regulatorów prędkości i położenia odpornych na zakłócenia i zmiany parametrów napędu” V Krajowa Konferencja Naukowa Sterowanie w Energoelektronice i Napędzie Elektrycznym SENE, Łódź-Arturówek, 14-16 listopada 2001r. ([3]). Obejmują one tematykę projektowania odpornych, cyfrowych regulatorów serwonapędów ze szczególnym uwzględnieniem zagadnień filtrowania sygnału pomiarowego prędkości oraz stosowania niekonwencjonalnej struktury PID z członami nieliniowymi w pętli regulacji położenia. Poruszany jest również problem doboru nastaw zapewniających poprawną pracę układu w całym zakresie zmian parametrów napędu. Szersze wprowadzenie do tematyki zawartej w w/w pracach znajduje się w rozdziale 6.

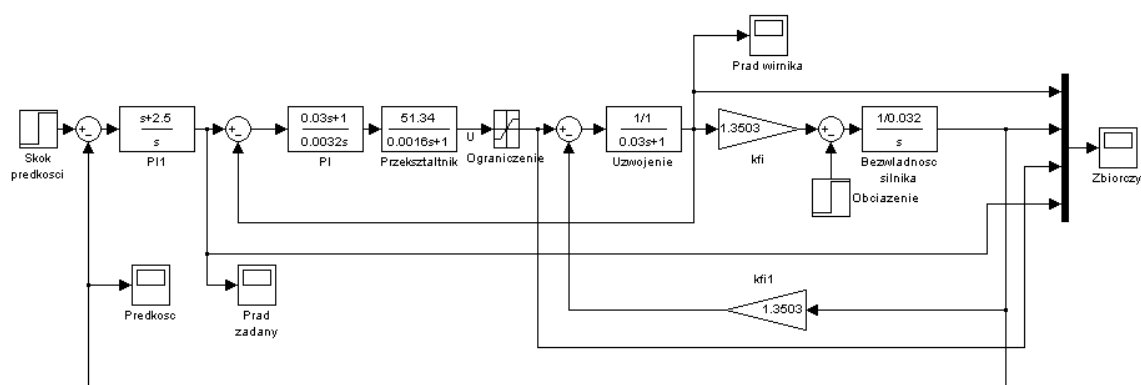
## 2. Przegląd narzędzi programistycznych do projektowania regulatorów na przykładzie programów Matlab/Simulink i Vissim.

Do syntezy i symulacji regulatorów można wykorzystywać różnorakie oprogramowanie obliczeniowo - symulacyjne. Jednym z najbardziej popularnych pakietów służących temu celowi jest Matlab/Simulink [8], [9] oraz Vissim [10] (należy także wspomnieć o bezpłatnym Scilabie [11], utworzonym nieco na wzór Matlaba). Wszystkie te narzędzia udostępniają środowiska do wizualnego tworzenia diagramów symulacyjnych: Simulink w przypadku Matlaba, Scicos w Scilabie i Vissim jako całość.

Wykorzystanie tych narzędzi ukierunkowane jest przede wszystkim na analizę w dziedzinie czasu. W przypadku analizy częstotliwościowej największe możliwości daje Matlab. Zawarte w nim (Control System Toolbox) funkcje umożliwiają badanie modeli LTI (Linear Time Invariant – liniowych stacjonarnych) pod względem ich charakterystyk częstotliwościowych, wykresów fazowych, ale również przebiegów czasowych. Jednak w przypadku modeli nieliniowych (a takimi będziemy zajmować się podczas projektowania regulatorów dla silników), analiza w dziedzinie czasu jest o wiele wygodniejsza w Simulinku lub Vissimie.

Poniżej zostanie przedstawiony krótki przegląd najczęściej używanych w tej pracy narzędzi zawartych w Matlabie/Simulinku. Opis Vissima będzie ograniczony do diagramów symulacyjnych wykorzystywanych przy modelowaniu regulatorów i napędów.

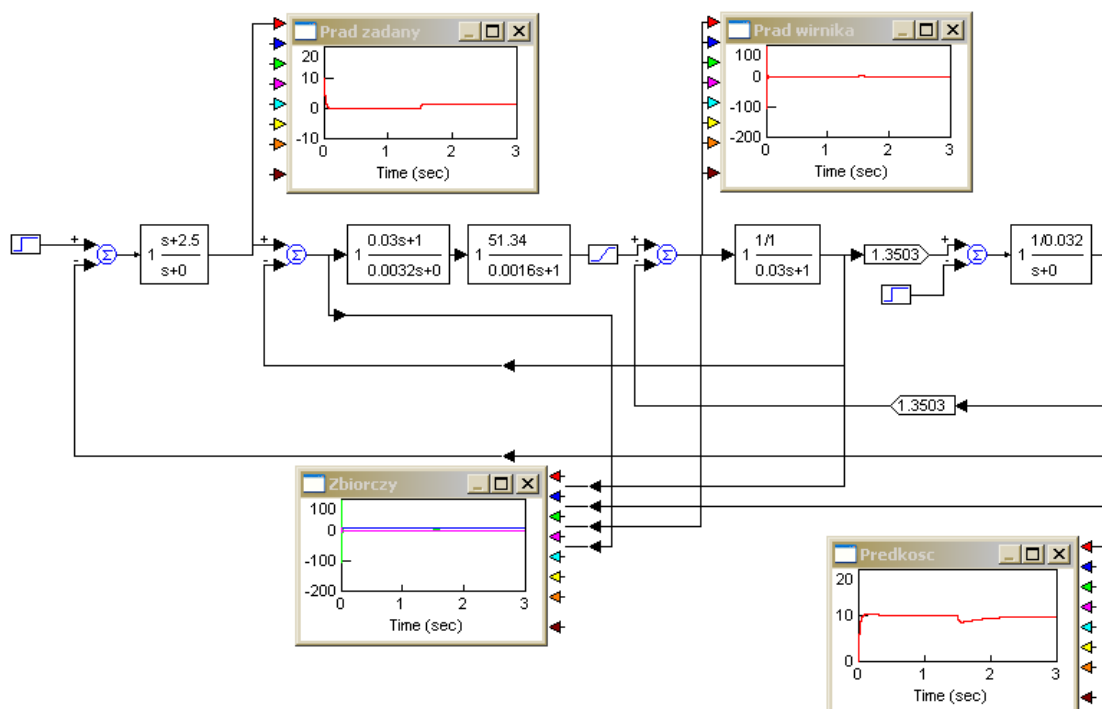
Na rysunkach 1 i 2 porównano postać prostego modelu<sup>1</sup> silnika prądu stałego z regulatorem stworzonego Simulinku i Vissimie.



Rysunek 1 - symulacja regulatora dla silnika prądu stałego stworzona w Simulinku

<sup>1</sup> zob. [5], rozdział p.t. „Modele matematyczne silników prądu stałego”





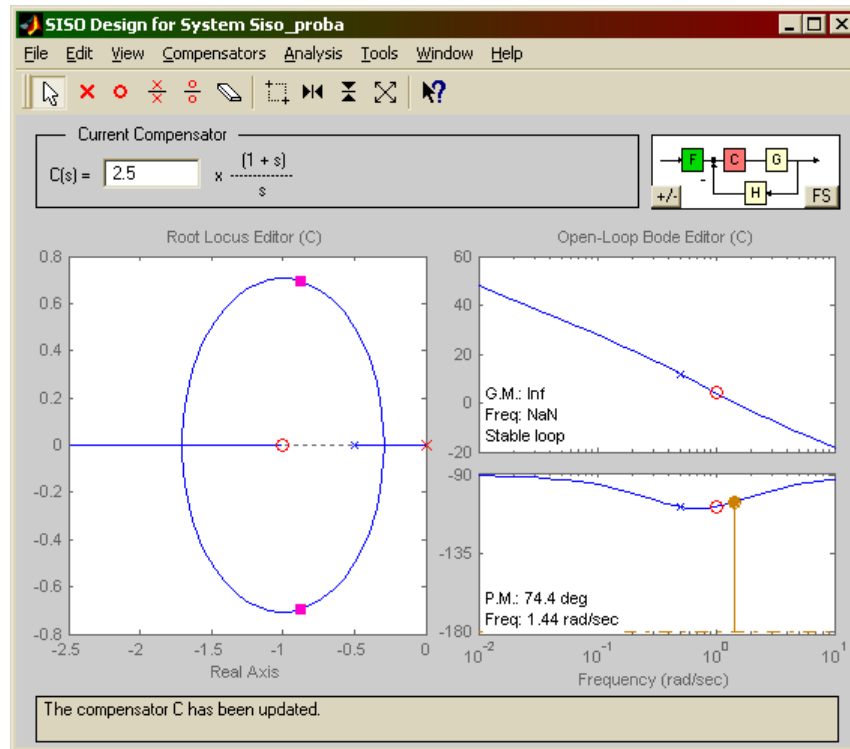
**Rysunek 2 - symulacja regulatora dla silnika prądu stałego stworzona w Vissimie**

Na powyższych diagramach widać standardowe bloki: wzmacnienia, integratory, bloki różniczkowania, nasycenia, sumatory i służące wizualizacji wirtualne oscyloskopy. Taki zestaw bloków jest już wystarczający do stworzenia nawet bardzo rozbudowanego systemu. Budowanie diagramów w obydwu programach jest bardzo zbliżone.

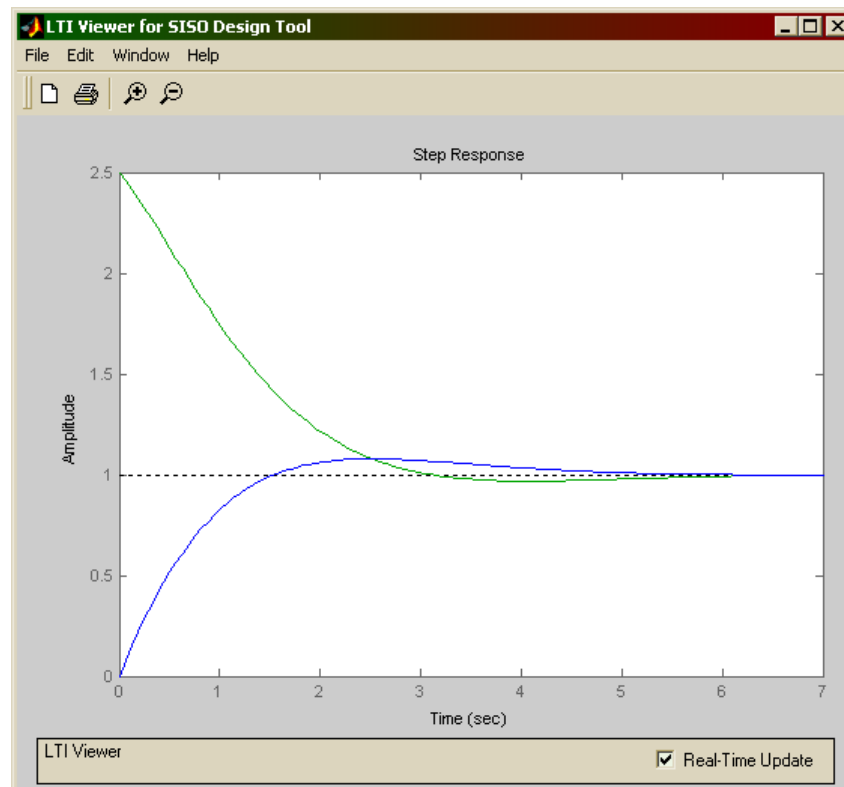
Należy wspomnieć o gotowych narzędziach do tworzenia regulatorów dostępnych w Matlabie.

Należą do nich:

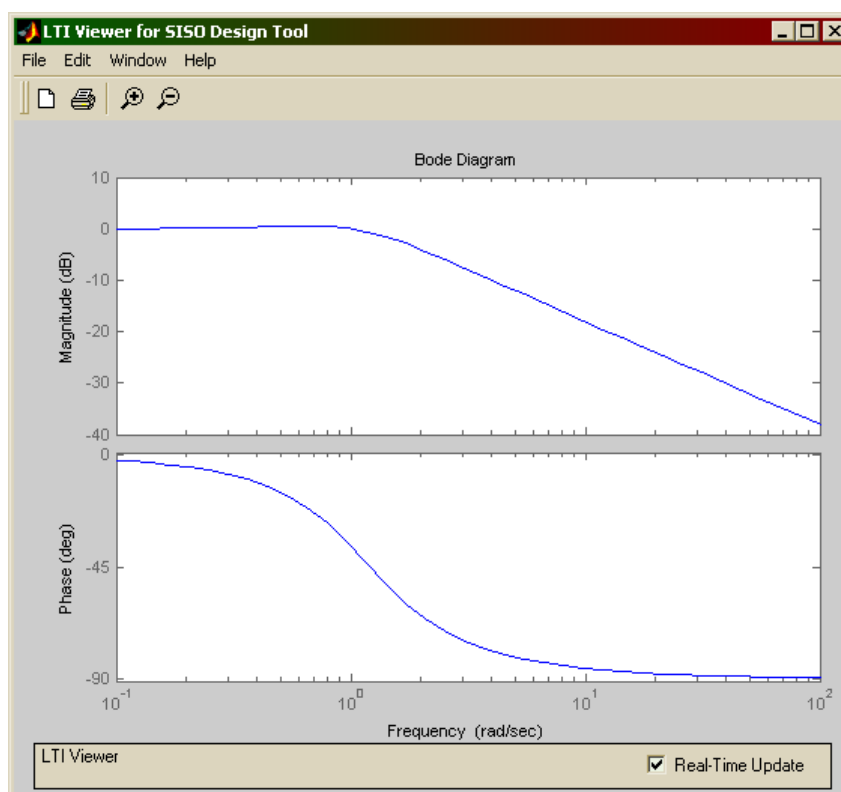
- I. SISO Design Tool [12] – narzędzie do tworzenia regulatorów i filtrów wejściowych dla układów o jednym wejściu i jednym wyjściu. Program posiada wygodny interfejs umożliwiający podgląd na bieżąco charakterystyk czasowych i częstotliwościowych zarówno układu otwartego (dla regulatora) oraz zamkniętego (obiekt + regulator). Na rysunkach 3, 4 i 5 zamieszczono widok wybranych okien programu w którym utworzono prosty regulator PI dla obiektu inercyjnego:



Rysunek 3 - okno główne narzędzia SISO Design Tool



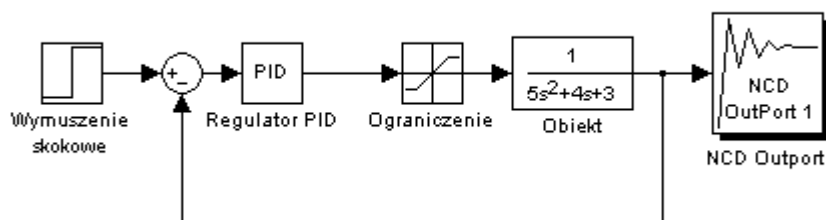
Rysunek 4 - odpowiedź skokowa zaprojektowanego układu



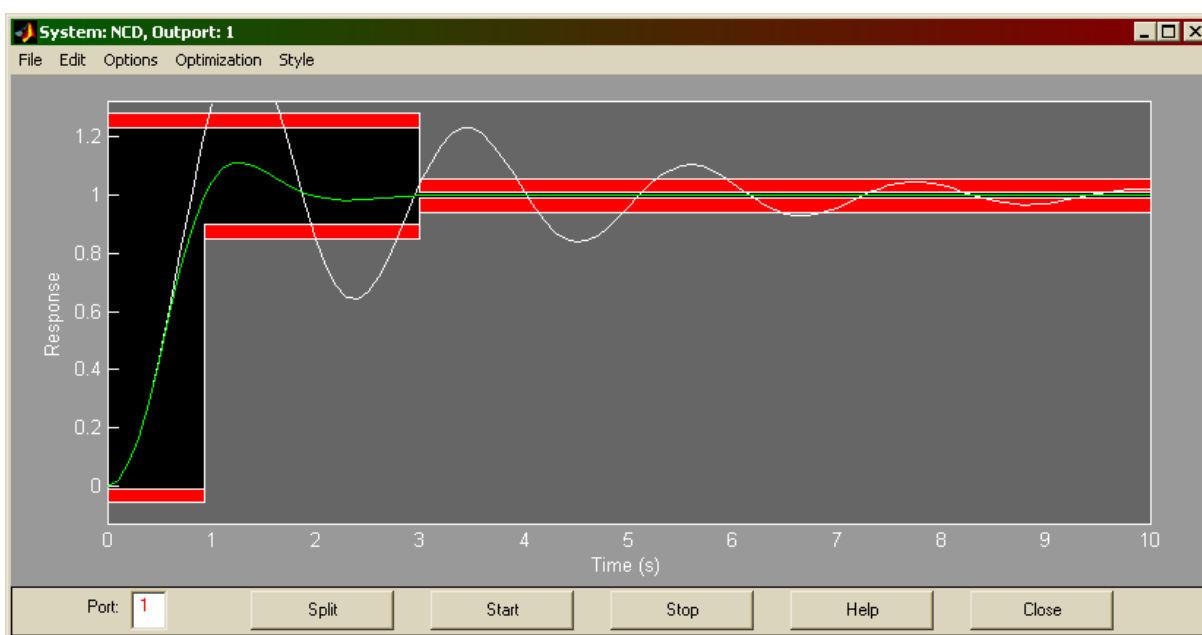
**Rysunek 5 - charakterystyka amplitudowo - fazowa układu zamkniętego**

Narzędzie to ma bardzo duże możliwości, jednak ogranicza się do obiektów i regulatorów liniowych.

- II. Następnym narzędziem użytecznym do projektowania regulatorów jest Simulinkowe NCD (Nonlinear Control Design) [13]. Dokonuje ono optymalizacji wybranych parametrów układu tak, aby uzyskać założone przebiegi czasowe (nie koniecznie na wyjściu, mogą to być przebiegi czasowe w dowolnym miejscu układu). To narzędzie można stosować z obiektami nieliniowymi, oraz z wielowymiarowymi. Jego ograniczenie polega na tym, że specyfikacji podlega jedynie przebieg czasowy na wyjściu i to tylko przy jednym, wybranym wymuszeniu. Jednak dzięki prostocie obsługi i małemu nakładowi czasowemu związanego z automatyczną pracą, narzędzie jest wygodne i szybko pomaga dobrać parametry regulatorów, ograniczeń, filtrów i innych elementów URA. Rysunki 6 i 7 przedstawiają idealny regulator PID sterujący obiektem oscylacyjnym, oraz okno programu NCD.



Rysunek 6 - model prostego układu regulacji automatycznej



Rysunek 7 - okno optymalizacji parametrów

Na rysunku widać odpowiedź skokową układu przy początkowych parametrach regulatora (linia szara) oraz po dokonaniu optymalizacji (linia zielona).

III. Dodatkowo należy wspomnieć o innych przydatnych narzędziach (choć nie związanych ściśle z regulatorami) [14]:

- *ltview* – przeglądarka obiektów LTI
- *tf* – tworzenie transmitancji obiektów linowych (LTI) ciągłych i dyskretnych
- *bode* – charakterystyka amplitudowo – fazowa obiektu LTI
- *nyquist* – charakterystyka amplitudowo – fazowa na płaszczyźnie zespolonej
- *step* – odpowiedź skokowa układu
- *impulse* – odpowiedź impulsowa układu
- *lsim* – symulacja układu dla zadanego wymuszenia
- *pade* – liniowe przybliżenie członu opóźniającego

Niestety, specyficzna budowa i wymagania stawiane regulatorom serwonapędów powodują, że narzędzia dobrze nadające się do projektowania prostych regulatorów nie sprawdzają się.

Powodem tego jest przede wszystkim występowanie nieliniowości (dokładność reprezentowania sygnałów - enkoder, ograniczenia, specjalne bloki nieliniowe), konieczność spełnienia wielu kryteriów jednocześnie (np. warunek zapasu stabilności w pętli regulacji przyspieszenia razem z warunkiem wartości przeregulowania w pętli regulacji prędkości), a także wymagania dotyczące oczekiwanych parametrów napędu (dokładność pozycjonowania, czasy regulacji i inne).

## 3. Pakiet oprogramowania do syntezy i symulacji regulatorów.

### 3.1. Założenia związane z budową i funkcjami programów.

Stworzone oprogramowanie miało na celu ułatwienie projektowania regulatorów położenia i prędkości dla serwonapędów z silnikami elektrycznymi. Na oprogramowanie mają się składać moduły pozwalające realizować następujące zadania:

- odczytywanie i zapisywanie w plikach danych początkowych (takich jak parametry silnika, oczekiwane wartości wskaźników, ograniczenia itp.);
- prezentacja użytych parametrów początkowych z możliwością ręcznego edytowania z poziomu interfejsu graficznego;
- obliczanie parametrów regulatora za pomocą programu zawartego w wybranym przez użytkownika pliku;
- prezentacja wyników działania w/w programu;
- umożliwienie użytkownikowi wyboru zestawu symulacji (wraz z dodatkowymi parametrami), które będą przeprowadzone przy użyciu obliczonych wcześniej parametrów zarówno maszyny, jak i regulatora;
- automatyczne przeprowadzenie wybranych symulacji;
- obliczenie wyników i wskaźników jakości na podstawie wygenerowanych przez symulację danych (program do obliczania wyników i wskaźników również ma być definiowany przez użytkownika);
- prezentacja wyników w postaci liczbowej;
- prezentacja wyników w postaci przebiegów czasowych;

Wszystkie - za wyjątkiem ostatniego - wymienione zadania miały być realizowane przez jedno spójne środowisko. Z uwagi na usystematyzowany tok postępowania w czasie projektowania regulatorów (i nie tylko regulatorów) przyjęto, że program zostanie zrealizowany w postaci kreatora, który krok po kroku przeprowadzi użytkownika przez proces projektowania.

Ostatnie zadanie – prezentacja wyników w postaci graficznej - ma zostać powierzone dodatkowemu programowi, który ma mieć postać wirtualnego oscyloskopu, dzięki któremu można będzie oglądać zapisane przebiegi, edytować je i zapisywać w ten sposób spreparowane dane.

Początkowo program miał być dość sztywno związany z zagadnieniem projektowania regulatorów. Jednak po kilku zmianach w nazewnictwie parametrów, plików, liczbie parametrów i symulacji okazało się, że program musi zostać stworzony jako surowy szkielet, którego zawartość będzie definiowana w kilku plikach tekstowych. Dzięki temu program zyskał na elastyczności.

Ostatnim wymogiem stawianym oprogramowaniu ma być skalowalność. Oznacza to, że można go łatwo przystosować do różnej ilości danych zarówno wejściowych jak i wyjściowych.

### 3.2. Prezentacja programów wchodzących w skład pakietu.

Poniżej zostaną przedstawione wszystkie (za wyjątkiem okien dialogowych, ostrzeżeń i komunikatów o błędach) okna wchodzące w skład programu. Prezentacja ta zawierać będzie jedynie opis elementów okien i ich przeznaczenia. Dokładny opis wszystkich funkcji znajduje się w rozdziale 5. Instrukcja użytkownika.

Wartość	Opis	Wskazanie	Wartość	Opis	Wskazanie
0	Max. wzgl. przer. położenia [s]	?	0	Max. wzgl. przer. położenia [s]	?
4	Max. moment oporowy [Nm]	?			?
1.2272e	Rozdzielczość enkodera [rad]	?			?
17.5	Max. wartość stałej silnika [Nm/A]	?			?
17.5	Min. wartość stałej silnika [Nm/A]	?			?
0.75	Min. moment bezwładności [kgm <sup>2</sup> ]	?			?
5.8	Max. moment bezwładności [kgm <sup>2</sup> ]	?			?
0.0001	Opóźnienie w pętli reg. przysp. [s]	?			?
3.1416	Margines wzmoc. reg. przysp. [ ]	?			?
0.2	Max. tetnienia prądu zadanego iq [A]	?			?
6	Poziom ograniczenia prądu [A]	?			?
0.5	Prędkość znamionowa [rad/s]	?			?
0.05	Max. abs. przer. prędkości [rad/s]	?			?
0.1	Max. wzgl. przer. prędkości [ ]	?			?
0.7	Początk. wsp. tłum. reg. położ. [ ]	?			?

**DALEJ**

Rysunek 8 - pierwsze okno kreatora - wprowadzanie parametrów

Jak widać, większość elementów tego okna stanowią pola edycyjne. Po załadowaniu pliku parametrów wypełniają się one wartościami. Kolor tła można definiować tak, aby użytkownik mógł w sposób graficzny grupować parametry. Oprócz pól edycyjnych, na formularzu widać opisy poszczególnych parametrów. Dodatkowo, dla każdego pola można zdefiniować nazwę pliku html w którym znajdować się będą dodatkowe informacje (wyświetlenie pliku następuje poprzez naciśnięcie przycisku ze znakiem ?).

Zapisywanie i odczytywanie plików z parametrami jest możliwe tylko w tym oknie. Dzieje się tak ze względu na potrzebę zapewnienia spójności danych w czasie działania programu.

Wciśnięcie przycisku Dalej powoduje obliczenie parametrów regulatora i przejście do okna 2. Parametry obliczone (rysunek 9):

Wartość	Opis parametru	?		?
0.0009	Opóźnienie zastępcze [s]	?		?
0.0008	Opóźnienie filtru [s]	?		?
0.00767	Rozdzielczość obl. prędkości [rad/s]	?		?
7.7333	Zmienność parametrów obiektu [ ]	?		?
23.8095	Wzmocnienie reg. przysp. [A/rad/s]	?		?
555.5543	Max. pulsacja odc. reg. przysp. [rad/s]	?		?
71.8389	Min. pulsacja odc. reg. przysp. [rad/s]	?		?
51.3922	Wzmocnienie reg. prędkości [1/s]	?		?
0.59116	Tłumienie w pętli prędkości [ ]	?		?
16	Rząd filtru różniczkującego [ ]	?		?
15.346	Wzmocnienie reg. położenia [1/s]	?		?
0.915	Tłumienie w pętli położenia [ ]	?		?
0.56737	Obniżenie ch-ki pierwiastkowej [rad/s]	?		?
0.56737	Min. prędkość na ch-ce nielin. [rad/s]	?		?
0.5	Ogr. prędkości w reg. poł. [rad/s]	?		?
17.4138	Ogr. narastania prędkości [rad/s <sup>2</sup> ]	?		?

Rysunek 9 - wyniki obliczenia parametrów regulatora

To okno ma budowę niemal identyczną jak okno wprowadzania parametrów. Jediną istotną różnicą jest fakt, że pól z wartościami w tym oknie nie można edytować.

Następnym oknem jest wybór zestawu symulacji i ich parametrów (rysunek 10):



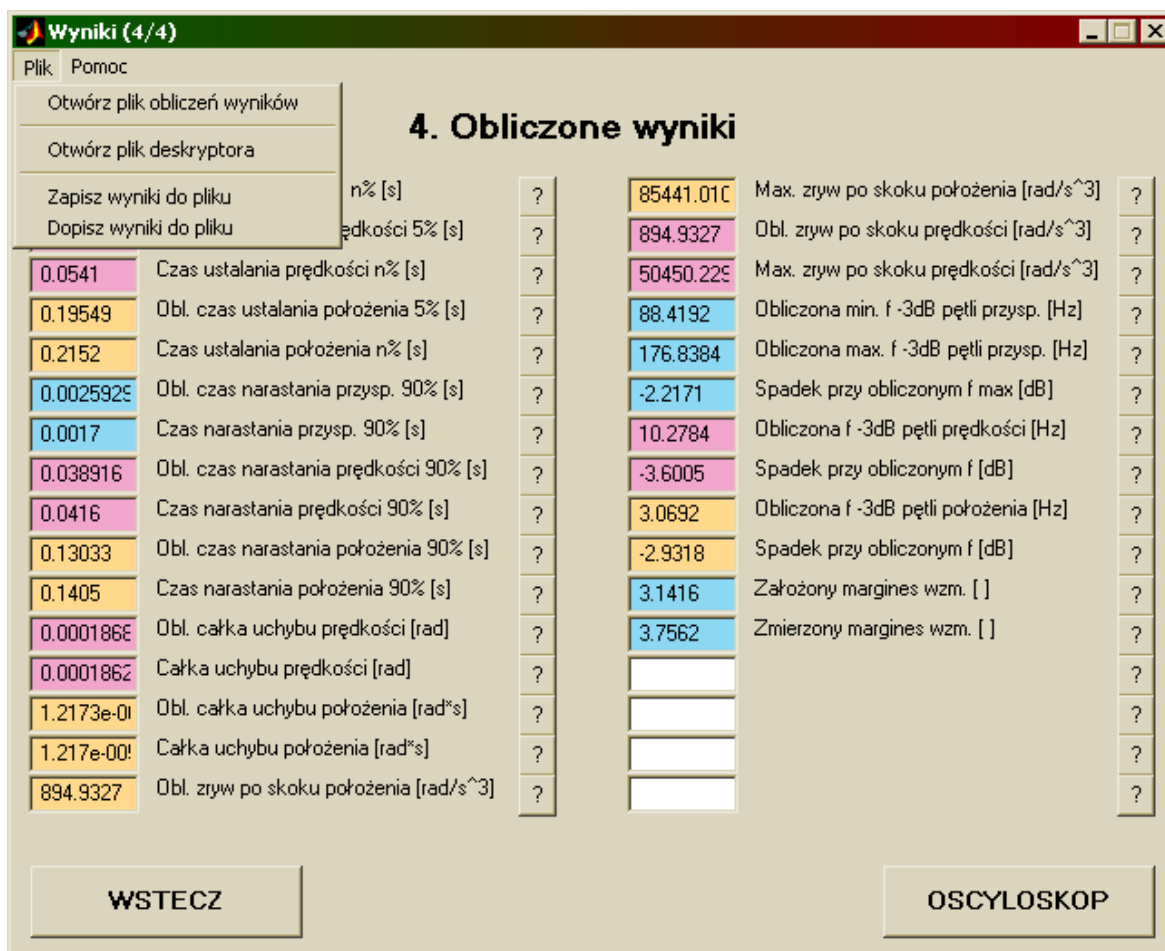
Parametr	Wartość	Parametr	Wartość	Parametr	Wartość
<input type="checkbox"/> Czas ustalania przyspieszenia n%	17.4138	skok_przyspieszenia_n	5	prec_przyspieszenia	?
<input type="checkbox"/> Czas ustalania prędkości n%	0.5	skok_predkosci_n	5	prec_predkosci	?
<input checked="" type="checkbox"/> Czas ustalania położenia n%	0.0369721	skok_polozenia_n	1	prec_polozenia	?
<input type="checkbox"/> Czas narastania przyspieszenia do 90%	17.4138	skok_przyspieszenia_0			?
<input type="checkbox"/> Czas narastania prędkości do 90%	0.5	skok_predkosci_09			?
<input type="checkbox"/> Czas narastania położenia do 90%	0.0369721	skok_polozenia_09			?
<input type="checkbox"/> IE prędkości po skoku obciążenia	4	skok_obc_predkosc			?
<input type="checkbox"/> IE położenia po skoku obciążenia	4	skok_obc_polozenie			?
<input checked="" type="checkbox"/> Max. zryw po skoku prędkości	0.5	skok_predkosci_zryw			?
<input checked="" type="checkbox"/> Max. zryw po skoku położenia	0.0369721	skok_polozenia_zryw			?
<input type="checkbox"/> Spadek 3dB w pętli przyspieszenia	176.838	fe3dBmax0	17.4138	amplituda_fe3dB	?
<input type="checkbox"/> Spadek 3dB w pętli prędkości	10.2784	fw3dB0	0.5	amplituda_fw3dB	?
<input type="checkbox"/> Spadek 3dB w pętli położenia	3.0692	ft3dB0	0.0369721	amplituda_ft3dB	?
<input type="checkbox"/> Badanie stabilności pętli przysp.					?
<input type="checkbox"/> <Pusty>					?
<input type="checkbox"/> <Pusty>					?

Rysunek 10 - okno wyboru symulacji

W tym oknie ma miejsce wybór symulacji do przeprowadzenia. Oprócz tego zostają wyświetlone parametry dodatkowe symulacji (dwa dla każdej symulacji). Parametry te, podobnie jak w oknie 1., można edytować. Następnym elementem jest lista rozwijana *Wariant symulacji*. Umożliwia ona wybranie wariantu symulacji dla wszystkich symulacji. Wariant symulacji jest liczbą o wartości takiej samej, jak pozycja wybrana na w/w liście. Sposób wykorzystania tego parametru zależy od użytkownika – nie ma tu bezpośredniego powiązania z żadnym elementem programu.

Na koniec warto zwrócić uwagę na polecenie *Wyczyść wyniki poprzednich symulacji* w menu *Plik*. Służy ono do usuwania wszystkich plików \*.mat, z wynikami wygenerowanymi we wcześniejszych symulacjach.

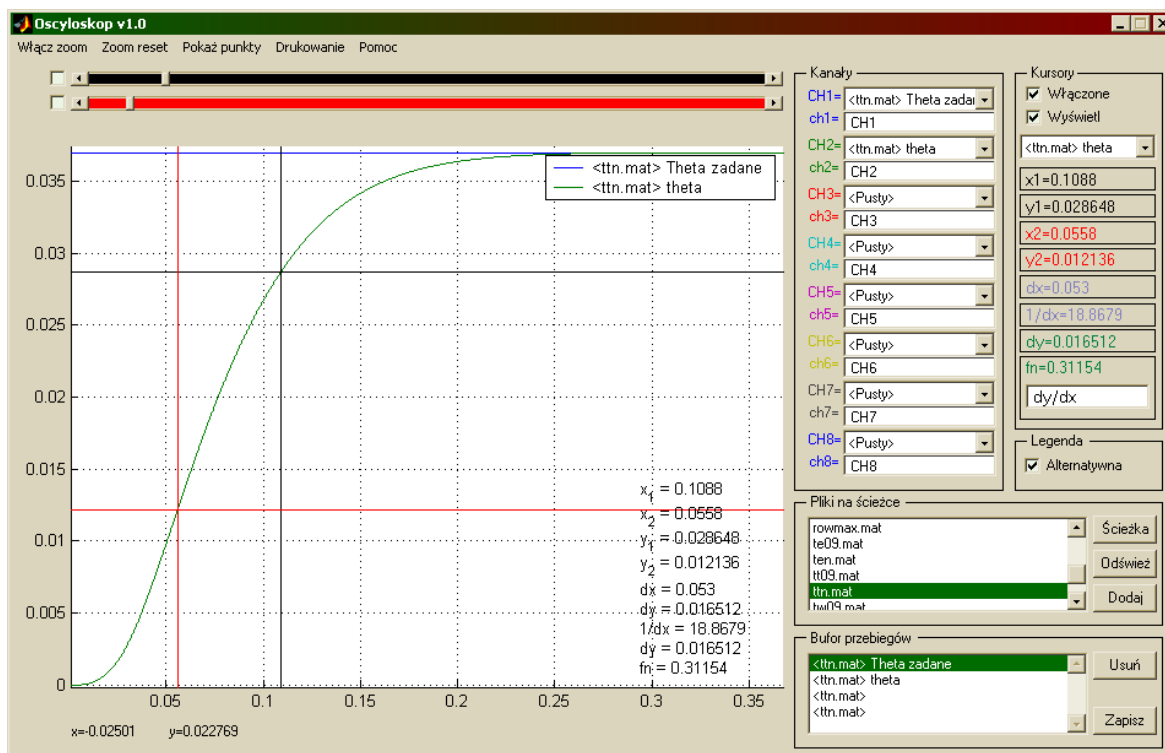
Po wciśnięciu przycisku Dalej przechodzimy do obliczenia wskaźników regulacyjnych i innych parametrów wyznaczonych na podstawie wyników symulacji, po czym jest wyświetlane ostatnie okno (rysunek 11):



Rysunek 11 - ostatnie okno kreatora - wyniki symulacji

W tym oknie oprócz pól z wyświetlanymi danymi znajdują się: przycisk *Oscyloskop* uruchamiający wirtualny oscyloskop, oraz polecenia *Plik->Zapisz wyniki do pliku* i *Plik->Dopisz wyniki do pliku*. Służą one eksportowaniu wszystkich istotnych danych utworzonych w programie do arkusza kalkulacyjnego Excel.

Ostatnim elementem interfejsu graficznego, przedstawionym na rysunku 12, jest program *Oscyloskop*.



Rysunek 12 - okno programu Oscyloskop

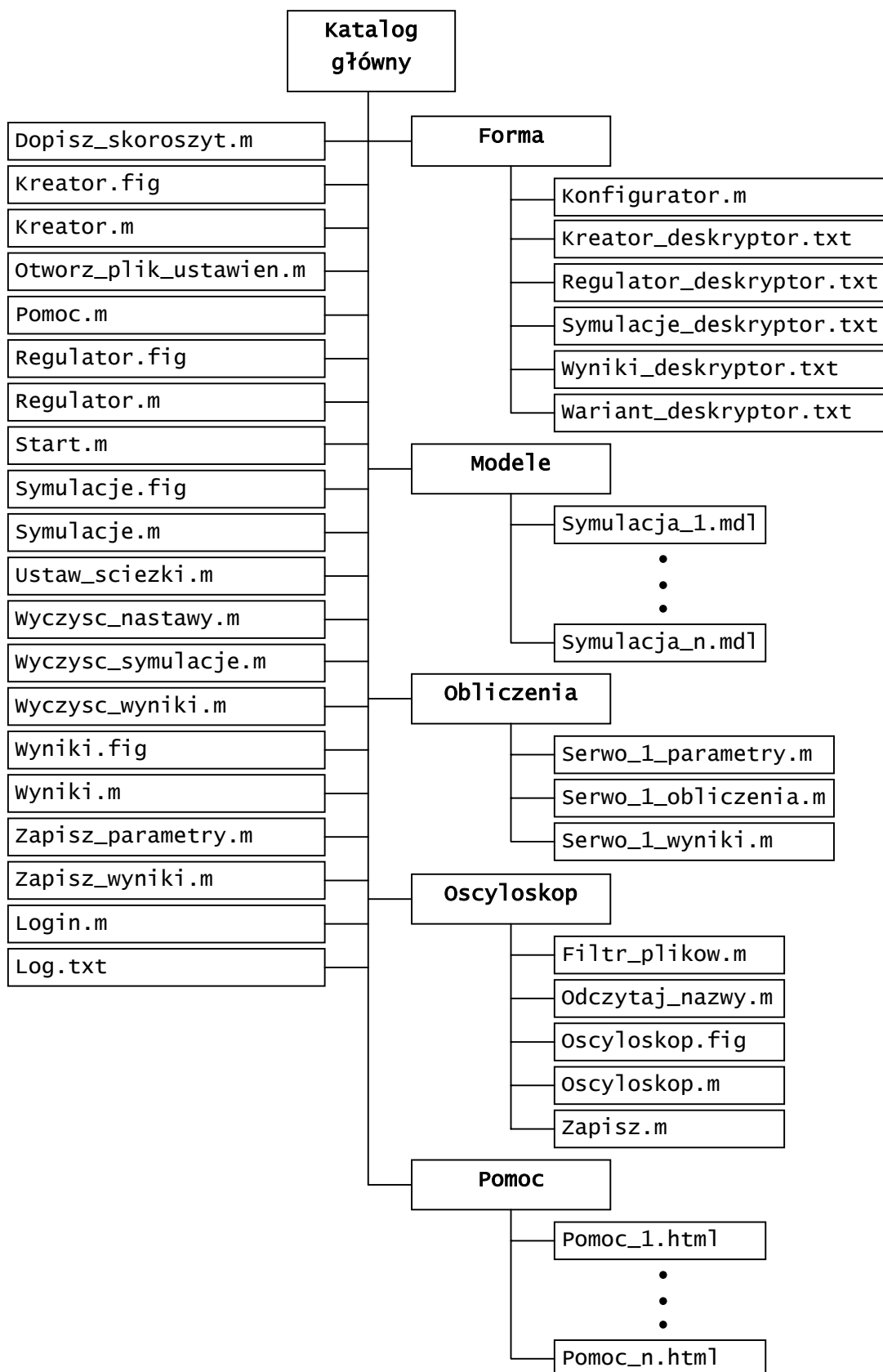
Program ten służy do analizowania wyników otrzymanych z symulacji; specyficzny sposób zapisywania danych umożliwia zachowanie nazwy zmiennej (odpowiadającej sygnałowi w Simulinku). Dodatkowo istnieje możliwość zapisania zawartości wszystkich kanałów (także tych utworzonych przez użytkownika) do nowego pliku oraz dodanie do takiego zapisu komentarza.

## **4. Szczegółowy opis funkcji i budowy wszystkich elementów oprogramowania.**

W tym rozdziale zostanie przedstawiona strona techniczna budowy oprogramowania. Ze względu na fakt, że ilość zmiennych zarówno w programie symulacyjnym, jak i oscyloskopie zmienia się dynamicznie, będzie im poświęcona szczególna uwaga.

### **4.1. Struktura plików i katalogów.**

Pliki wchodzące w skład programów rozmieszczone w sposób przedstawiony na rysunku 13. Dla określenia znaczenia poszczególnych plików, w tabeli 1 przedstawione jest zestawienie zawierające nazwy plików, ich położenie, przeznaczenie, możliwość modyfikowania i miejsca wywoływania. Zestawienie to ma na celu ułatwienie modyfikacji programu lub naprawienia ewentualnych błędów.



Rysunek 13- struktura katalogów i plików

Tabela 1 - pliki wchodzące w skład programów

Nazwa pliku	Położenie	Przeznaczenie	Możliwość modyfikacji	Miejsce wywołania
Dopisz_skoroszyt.m	Katalog główny	Dopisywanie kolumny z danymi do wcześniej zapisanego skoroszytu Excela	Możliwość dodania pozycji, która będzie zapisywana w plikach wyników (uwaga na zachowanie zgodności z Zapisz_wyniki.m)	Wyniki.m
Kreator.fig	Katalog główny	Plik figury dla pierwszego okna kreatora, zawiera dane definiujące elementy menu	-	Kreator.m
Kreator.m	Katalog główny	Plik funkcyjny pierwszego okna, zawiera funkcje niezbędne do działania okna	-	Start.m
Otworz_plik_ustawien.m	Katalog główny	Służy do otwierania początkowego pliku z parametrami	-	Kreator.m
Pomoc.m	Katalog główny	Funkcja wykorzystywana przez elementy menu do otwierania stron pomocy	-	Kreator.m Regulator.m Symulacje.m Wyniki.m Oscyloskop.m Otworz_plik_ustawien.m Zapisz_parametry.m
Regulator.fig	Katalog główny	Plik figury dla drugiego okna kreatora, zawiera dane definiujące elementy menu	-	Regulator.m
Regulator.m	Katalog główny	Plik funkcyjny drugiego okna, zawiera funkcje niezbędne do działania okna	-	Kreator.m Symulacje.m
Start.m	Katalog główny	Plik startowy, służy do uruchamiania programu i zawiera stałe konfiguracyjne	Zmiana ścieżek i nazw plików opisujących zawartość okien - deskryptorów	Wywoływany przez użytkownika
Symulacje.fig	Katalog główny	Plik figury dla trzeciego okna kreatora, zawiera dane definiujące elementy menu	-	Symulacje.m
Symulacje.m	Katalog główny	Plik funkcyjny trzeciego okna, zawiera funkcje niezbędne do działania okna	-	Regulator.m Wyniki.m
Ustaw_sciezki.m	Katalog główny	Dodawanie do bieżącej ścieżki Matlaba odpowiednich katalogów programu	Jeśli program ma wywoływać zewnętrzne programy użytkownika, w tym pliku można dodać odpowiednie ścieżki	Start.m
Wyczysc_nastawy.m	Katalog główny	Wykonywany podczas cofania z okna 2. Regulator do 1. Kreator; czyści obliczone nastawy regulatora	-	Regulator.m
Wyczysc_symulacje.m	Katalog główny	Usuwanie plików wynikowych (*.mat) symulacji; usuwanie odbywa się w katalogu głównym i podrzędnym Modele	-	Symulacje.m
Wyczysc_wyniki.m	Katalog główny	Wykonywany podczas cofania z okna 4. Wyniki do 3. symulacje; czyści obliczone wyniki i wskaźniki symulacji	-	Wyniki.m
Wyniki.fig	Katalog główny	Plik figury czwartego okna kreatora, zawiera dane definiujące elementy menu	-	Wyniki.m
Wyniki.m	Katalog główny	Plik funkcyjny czwartego okna, zawiera funkcje niezbędne do działania okna	-	Symulacje.m
Zapisz_parametry.m	Katalog główny	Zapisywanie parametrów początkowych w tym samym lub nowym pliku	-	Kreator.m

Zapisz_wyniki.m	Katalog główny	Zapisywanie wszystkich znaczących danych do skoroszytu Excela	Możliwość dodania pozycji, która będzie zapisywana we wszystkich plikach wyników	Wyniki.m
Konfigurator.m	Katalog główny\Forma (ścieżka opcjonalna)	Interpreter deskryptorów, służy do konfigurowania wyglądu poszczególnych okien	Jeśli struktura wizualna programu ulegnie zmianie (np. dodane zostaną dodatkowe pola edycyjne) to będzie potrzebna modyfikacja tego pliku	Kreator.m Regulator.m Symulacje.m Wyniki.m
Kreator_deskryptor.txt (nazwa opcjonalna)	Katalog główny\Forma (ścieżka opcjonalna)	Plik zawierający definicje wyglądu okna 1. Kreator	Modyfikowany w zależności od potrzeb, opis formatu zapisu tego pliku w rozdziale 5.1	Konfigurator.m
Regulator_deskryptor.txt (nazwa opcjonalna)	Katalog główny\Forma (ścieżka opcjonalna)	Plik zawierający definicje wyglądu okna 2. Regulator	Modyfikowany w zależności od potrzeb, opis formatu zapisu tego pliku w rozdziale 5.2	Konfigurator.m Wyczyszc_nastawy.m
Symulacje_deskryptor.txt (nazwa opcjonalna)	Katalog główny\Forma (ścieżka opcjonalna)	Plik zawierający definicje wyglądu okna 3. Symulacje; w pliku tym mogą być również zawarte wartości liczbowe dwóch opcjonalnych parametrów dla każdej symulacji	Modyfikowany w zależności od potrzeb, opis formatu zapisu tego pliku w rozdziale 5.3	Konfigurator.m
Wyniki_deskryptor.txt (nazwa opcjonalna)	Katalog główny\Forma (ścieżka opcjonalna)	Plik zawierający definicje wyglądu okna 4. Wyniki	Modyfikowany w zależności od potrzeb, opis formatu zapisu tego pliku w rozdziale 5.4	Konfigurator.m Wyniki.m
Wariant_deskryptor.txt (nazwa opcjonalna)	Katalog główny\Forma (ścieżka opcjonalna)	Plik zawiera pozycje menu rozwijanego (wyboru wariantu symulacji) w oknie 3. Symulacje	Modyfikowany w zależności od potrzeb, opis formatu zapisu tego pliku w rozdziale 5.3	Konfigurator.m
Symulacja_1.mdl ... Symulacja_n.mdl (nazwy opcjonalne)	Katalog główny\Modele (ścieżka opcjonalna)	Pliki symulacyjne zdefiniowane przez użytkownika; definicje nazw i położenia tych plików znajdują się odpowiednio w pliku deskryptora symulacji (np. Symulacje_deskryptor.txt) oraz pliku Start.m	Dowolne symulacje stworzone w Simulinku. Jedynymi wymaganiami co do ich budowy jest, aby przekazywanie parametrów do symulacji odbywało się przez strukturę danych parametry (lub p)	Symulacje.m
Serwo_1_parametry.m (nazwa musi kończyć się na _parametry.m)	Katalog główny\Obliczenia (ścieżka opcjonalna)	Plik definiujący parametry początkowe; ten plik może zawierać również inne parametry (liczbowe) które mają być zapisywane – zapisywanie parametrów uwzględnia tylko zmienne zdefiniowane w tym pliku	Dowolne modyfikacje z następującymi ograniczeniami: plik musi zawierać deklarację global parametry, wpisy do struktury parametry muszą być liczbami	Kreator.m
Serwo_1_obliczenia.m (nazwa musi kończyć się na _obliczenia.m)	Katalog główny\Obliczenia (ścieżka opcjonalna)	Plik obliczający parametry regulatora; przyjmuje dane w strukturze parametry a wyniki są do tej struktury dopisywane	Dowolny program Matlaba, musi zawierać deklarację global parametry	Kreator.m
Serwo_1_wyniki.m (nazwa musi kończyć się na _wyniki.m)	Katalog główny\Obliczenia (ścieżka opcjonalna)	Obliczanie wskaźników i innych wyników symulacji, efekty obliczeń zapisywane w strukturze parametry	Dowolny program Matlaba, aby było możliwe wyświetlenie wyników musi zawierać deklarację global parametry i wpisy do struktury parametry	Symulacje.m
Filtr_plikow.m	Katalog główny\Oscyloskop	Odnajduje pliki o rozszerzeniach .mat i .dat i zwraca ich nazwy i rozszerzenia	Możliwość dodania innych potrzebnych rozszerzeń	Oscyloskop.m
Odczytaj_nazwy.m	Katalog główny\Oscyloskop	Odczytuje z etykiety Simulinka nazwy sygnałów i tworzy na tej podstawie nazwy zawierające nazwę pliku oraz nazwę sygnału	-	Oscyloskop.m
Oscyloskop.fig	Katalog główny\Oscyloskop	Plik figury programu Oscyloskop, zawiera dane definiujące elementy menu	-	Oscyloskop.m

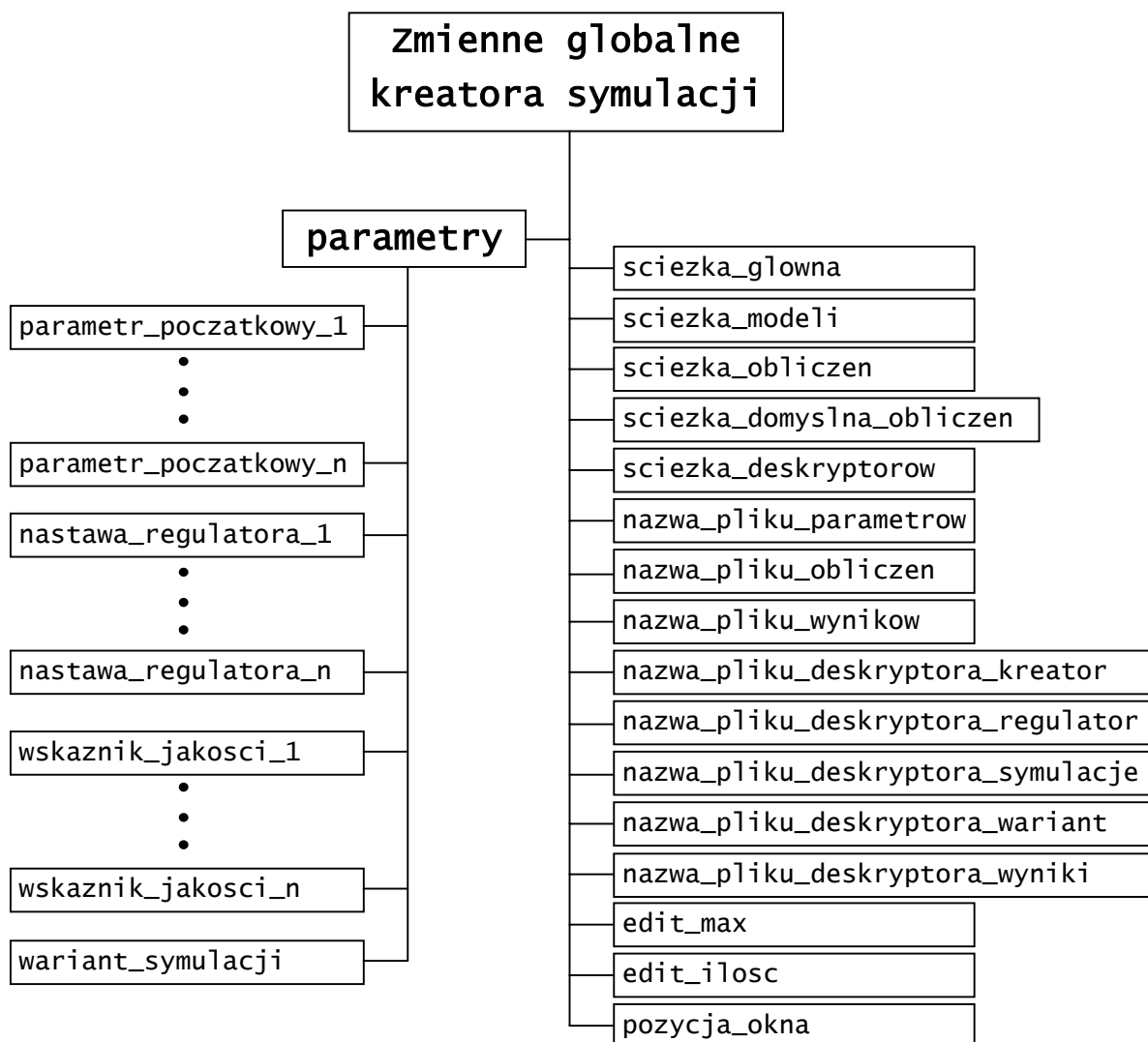
Oscyloskop.m	Katalog główny\Oscyloskop	Plik funkcyjny programu Oscyloskop, zawiera funkcje niezbędne do działania okna	-	Wyniki.m Wywoływany przez użytkownika
Pomoc_1.html ... Pomoc_n.html (nazwy opcjonalne)	Katalog główny\Pomoc (ścieżka opcjonalna)	Pliki *.html zawierające pomoc na temat różnych elementów programów (na temat używania poszczególnych funkcji, na temat parametrów, symulacji itp.)	Dowolne pliki *.html użytkownika	Kreator.m Regulator.m Symulacje.m Wyniki.m Oscyloskop.m oraz niektóre okna dialogowe
Login.m	Katalog główny	Funkcja do zapisywania zdarzeń programu	Modyfikacja umożliwia zmianę nazwy pliku logu	Kreator.m Regulator.m Symulacje.m Wyniki.m Oscyloskop.m oraz niektóre okna dialogowe
Log.txt	Katalog główny	Domyślny plik tekstowy z wykazem zdarzeń programu	-	Login.m

## 4.2. Struktury danych.

### 4.2.1. Kreator symulacji.

Najważniejsze zmienne, zarówno w kreatorze symulacji, jak i programie Oscyloskop umieszczone są w strukturze. Dla kreatora są to przede wszystkim parametry związane z napędem, regulatorem i wynikami symulacji. Oprócz struktury *parametry* w kreatorze symulacji występują również zmienne globalne (konfiguracyjne). Struktura ta została przedstawiona na rysunku 14.





Rysunek 14 - zmienne związane z kreatorem symulacji

Znaczenie poszczególnych zmiennych kreatora symulacji jest przedstawione w tabeli 2.

Tabela 2 - zmienne kreatora symulacji

Nazwa zmiennej	Typ	Przeznaczenie	Możliwość modyfikacji	Miejsce utworzenia
parametry. parametr_początkowy_1 ... parametry. parametr_początkowy_n (nazwy opcjonalne)	liczba	stałe wartości potrzebne do obliczenia parametrów regulatora; również inne zmienne użytkownika (dowolne dane <u>liczbowe</u> ) które mają być pamiętane po zamknięciu programu	modyfikowanie z poziomu interfejsu (pól edycyjnych); zmiany mogą być zapisane do tego samego lub nowego pliku parametrów	parametry ładowane z pliku z ustawieniami początkowymi
parametry. nastawa_regulatora_1 ... parametry. nastawa_regulatora_n (nazwy opcjonalne)	dowolny <sup>1</sup>	parametry w modelach symulacyjnych	zmienne niedostępne bezpośrednio (można je edytować z linii poleceń)	obliczone w pliku obliczeń regulatora

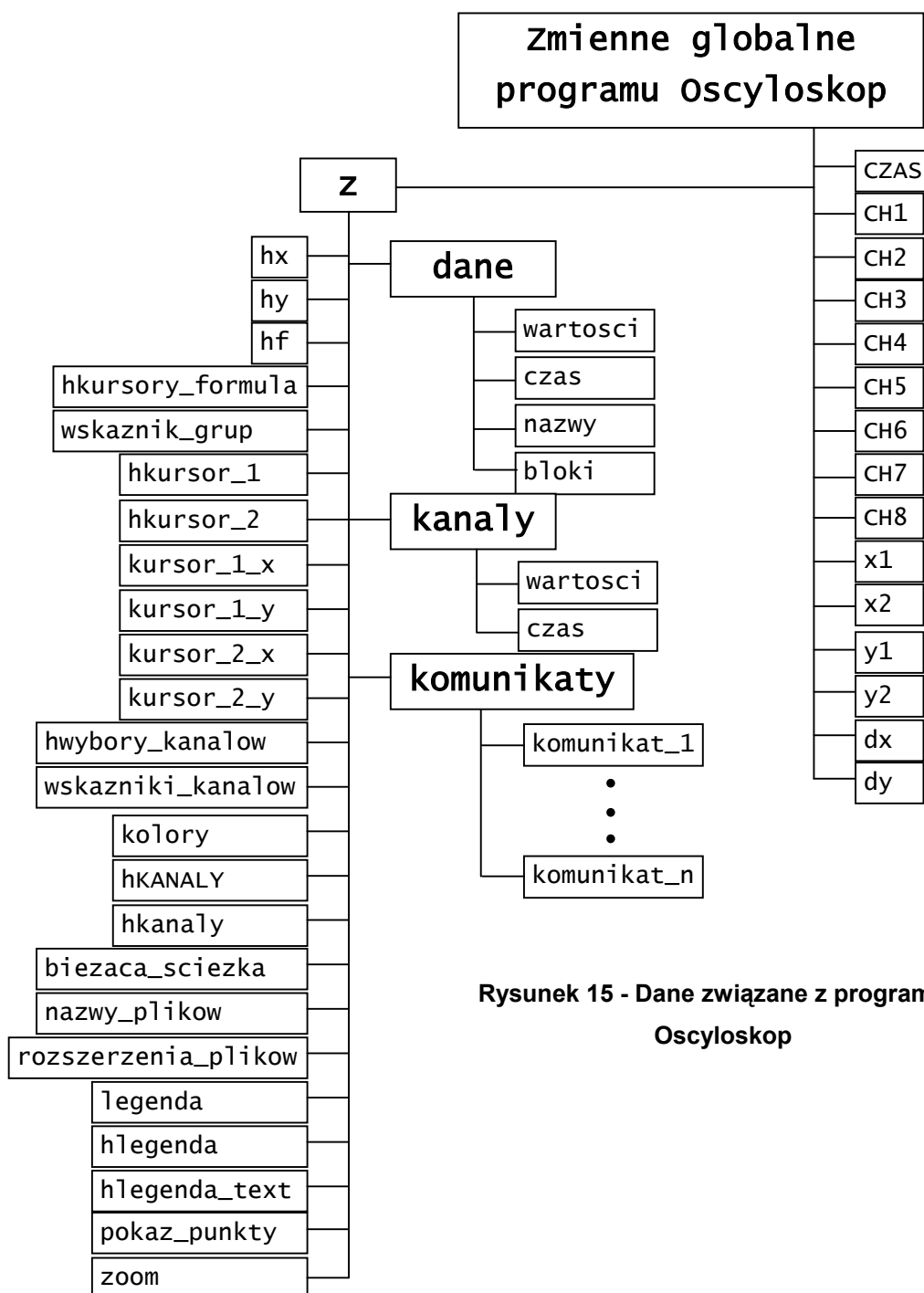
<sup>1</sup> format zmiennej jest dowolny, jednak w oknach programu można wyświetlić tylko dane liczbowe

parametry. wskaznik_jakosci_1 ... parametry. wskaznik_jakosci_n (nazwy opcjonalne)	dowolny <sup>1</sup>	wyniki symulacji prezentowane w ostatnim oknie kreatora – 4. Obliczone wyniki		plik obliczeń wyników symulacji
parametry. wariant_symulacji	liczba (całkowita)	pozwała wybrać wariant symulacji dla wszystkich eksperymentów (tutaj wariant symulacji określa dla jakiej kombinacji zmiennych parametrów napędu ma być przeprowadzona symulacja)	użytkownik może dowolnie edytować wartość zmiennej, pod warunkiem, że deskryptor wariantu oraz pliki symulacyjne będą uwzględniać tę wartość	parametr ładowany z pliku z ustawieniami początkowymi oraz zmieniany w oknie wyboru symulacji (lista rozwijana)
sciezka_glowna	łańcuch znakowy	główna ścieżka programu odczytywana w momencie uruchamiania	-	<i>Start.m</i>
sciezka_modeli	łańcuch znakowy	ścieżka w której program będzie szukał plików symulacyjnych, tam również będą zapisywane pliki *.mat z danymi zarejestrowanymi w czasie symulacji, domyślnie jest to katalog <i>Modele</i> w katalogu głównym programu	dowolna ścieżka osiągalna z poziomu Matlaba	<i>Start.m</i>
sciezka_obliczen	łańcuch znakowy	na tej ścieżce program szuka plików z parametrami, pliku obliczeń i pliku wyników	-	użytkownik określa tę ścieżkę poprzez podanie pliku parametrów – jego ścieżka staje się automatycznie ścieżką pliku obliczeń i wyników
sciezka_domyslna_obliczen	łańcuch znakowy	początkowa ścieżka, na której otwiera się okno dialogowe wyboru pliku parametrów; tutaj jest to katalog <i>Obliczenia</i> w katalogu głównym programu	dowolna ścieżka osiągalna z poziomu Matlaba	<i>Start.m</i>
sciezka_deskryptorow	łańcuch znakowy	położenie plików opisujących zawartość okien programu – deskryptorów; również na tej ścieżce musi znajdować się plik konfigurujący zawartość okien – <i>konfigurator.m</i>	dowolna ścieżka osiągalna z poziomu Matlaba, pod warunkiem że będzie zawierać oryginalny plik <i>konfigurator.m</i>	<i>Start.m</i>
nazwa_pliku_deskryptora_kreator	łańcuch znakowy	zmienna zawiera nazwę pliku tekstowego opisującego zawartość okna <i>Kreator (1/4): 1. Parametry napędu</i>	dowolna nazwa poprawnego pliku deskryptora <sup>2</sup> okna 1. znajdującego się na ścieżce deskryptorów	<i>Start.m</i>
nazwa_pliku_deskryptora_regulator	łańcuch znakowy	zmienna zawiera nazwę pliku tekstowego opisującego zawartość okna <i>Regulator(2/4): Parametry obliczone</i>	dowolna nazwa poprawnego pliku deskryptora <sup>2</sup> okna 2. znajdującego się na ścieżce deskryptorów	<i>Start.m</i>
nazwa_pliku_deskryptora_symulacje	łańcuch znakowy	zmienna zawiera nazwę pliku tekstowego opisującego zawartość okna <i>Symulacje (3/4): Wybór symulacji</i>	dowolna nazwa poprawnego pliku deskryptora <sup>2</sup> okna 3. znajdującego się na ścieżce deskryptorów	<i>Start.m</i>
nazwa_pliku_deskryptora_wariant	łańcuch znakowy	zmienna zawiera nazwę pliku tekstowego opisującego zawartość listy rozwijanej <i>Wariant symulacji</i> w oknie wyboru symulacji	dowolna nazwa poprawnego pliku deskryptora <sup>2</sup> listy wariantów symulacji znajdującego się na ścieżce deskryptorów	<i>Start.m</i>
nazwa_pliku_deskryptora_wyniki	łańcuch znakowy	zmienna zawiera nazwę pliku tekstowego opisującego zawartość okna <i>Wyniki (4/4): Obliczone wyniki</i>	dowolna nazwa poprawnego pliku deskryptora <sup>2</sup> okna 4. znajdującego się na ścieżce deskryptorów	<i>Start.m</i>
edit_max	liczba (całkowita)	określa liczbę okien edycyjnych dla okien <i>Kreator (1/4)</i> , <i>Regulator (2/4)</i> oraz <i>Wyniki (4/4)</i>	wartość stała równa 32,	<i>Start.m</i>
pozycja_okna	wektor 1x4	określa położenia i rozmiar wszystkich okien na ekranie	automatycznie dopasowane do rozdzielczości ekranu	<i>Start.m</i>

<sup>2</sup> więcej na temat plików opisujących okna w rozdziale 5.1., 5.2., 5.3. oraz 5.4.

#### 4.2.2. Program Oscyloskop.

W następnej kolejności omówione zostaną zmienne używane w programie Oscyloskop. W tym programie wszystkie ważne dane umieszczone zostały w globalnej strukturze o nazwie z. Dodatkowo, aby umożliwić użytkownikowi bezpośredni dostęp do danych, utworzone zostały zmienne globalne CH1...CH8 oraz x1, x2, y1, y2, dx, dy. Zostaną one omówione później. Poniżej, podobnie jak dla kreatora symulacji, zostaje przedstawione drzewo zmiennych programu Oscyloskop (rysunek 15). Ponieważ prawie cały kod programu znajduje się w jednym pliku, w tabeli 3 zawierającej opis zmiennych pominięto kolumnę *Miejsce utworzenia*.



Rysunek 15 - Dane związane z programem Oscyloskop

Tabela 3 - zmienne programu Oscyloskop

Nazwa zmiennej	Typ	Przeznaczenie	Możliwość modyfikacji
z.hx	uchwyt	uchwyt umożliwiający odwołania do pola tekstowego wyświetlającego aktualne położenie kursora na wykresie (oś x)	-
z.hy	uchwyt	uchwyt umożliwiający odwołania do pola tekstowego wyświetlającego aktualne położenie kursora na wykresie (oś y)	-
z.hfn	uchwyt	pozwala na odwołania do pola wyświetlającego wartość obliczoną wyrażenia związanego z położeniem kursorów	-
z.hkursory_formula	uchwyt	jest to uchwyt pola edycyjnego, w którym użytkownik może definiować dodatkowe wyrażenie związane z położeniem kursorów	-
z.hkursor_1	uchwyt	uchwyt do linii tworzących krzyż kursora pierwszego (czarnego)	-
z.hkursor_2	uchwyt	uchwyt do linii tworzących krzyż kursora drugiego (czerwonego)	-
z.kursor_1_x	liczba	wartość położenia w osi x pierwszego kursora	zmiana poprzez ruch suwakami kursorów lub zmianę zawartości wykresu
z.kursor_1_y	liczba	wartość położenia w osi y pierwszego kursora	zmiana poprzez ruch suwakami kursorów lub zmianę zawartości wykresu
z.kursor_2_x	liczba	wartość położenia w osi x drugiego kursora	zmiana poprzez ruch suwakami kursorów lub zmianę zawartości wykresu
z.kursor_2_y	liczba	wartość położenia w osi y drugiego kursora	zmiana poprzez ruch suwakami kursorów lub zmianę zawartości wykresu
z.hwybory_kanalow	wektor 1x8 uchwytów	uchwyty do list rozwijanych służących wyborowi zawartości poszczególnych kanałów <sup>1</sup>	-
z.wskazniki_kanalow	wektor 1x8 liczb (całkowitych)	wektor przypisań poszczególnych list rozwijanych; np. wektor o wartości [0 1 2 0 0 0 0] oznacza że tylko w kanale 2. i 3. wybrano jakieś dane i są to odpowiednio 1. i 2. wektor ze struktury z.dane.wartosci	zmieniane poprzez wybór pozycji z list rozwijanych poszczególnych kanałów; wartości mogą również ulec zmianie jeśli zostanie usunięta jakaś dana
z.kolory	macierz 63x3 liczb	każdy wiersz macierzy określa kolor kolejnego kanału (wykorzystane tylko 8), jest to standardowa paleta kolorów <i>colormap('lines')</i> Matlaba	można ręcznie edytować kolory wykresu poprzez odwołania do globalnej zmiennej z.kolory
z.hKANALY, hkanaly	wektory 1x8 uchwytów	są to uchwyty do pól tekstowych ( <i>CH1...CH8, ch1...ch8</i> )i umożliwiają dopasowanie koloru napisów przy poszczególnych kanałach do kolorów wykresów tych kanałów	-
z.biezaca_sciezka	łańcuch znakowy	przechowuje ścieżkę, której zawartość jest wyświetlana w oknie przeglądania plików	zmiana bieżącej ścieżki odbywa się po naciśnięciu przycisku <i>Ścieżka</i> lub jeśli program będzie wywołany z parą parametrów, np: <i>Oscyloskop('sciezka', 'd:\dokumenty\wykresy')</i>
z.nazwy_plikow	macierz komórkowa łańcuchów znakowych	przechowuje nazwy skojarzonych plików (*.mat i *.dat) znajdujących się na bieżącej ścieżce (bez rozszerzenia)	zmiana spisu następuje po zmianie bieżącej ścieżki lub po odświeżeniu jej zawartości przy pomocy przycisku <i>Odśwież</i>
z.rozszerzenia_plikow	macierz komórkowa łańcuchów znakowych	przechowuje rozszerzenia w/w plików	zmiana spisu następuje po zmianie bieżącej ścieżki lub po odświeżeniu jej zawartości przy pomocy przycisku <i>Odśwież</i>

<sup>1</sup> zawartość wykresu nie jest zdeterminowana tylko przez wymienione wybory; poprzez zmianę pól edycyjnych użytkownik może modyfikować zawartość poszczególnych kanałów (wektorów liczbowych), patrz rozdział 5.5. Instrukcja użytkownika – program Oscyloskop

z.legendą	macierz komórkowa łańcuchów znakowych	zawiera teksty wyświetlane w oknie legendy, np. CH1, CH2	tekst legendy zależy od wyboru zawartości kanałów oraz treści pól edycyjnych
z.hlegenda	uchwyt	uchwyt do osi (wykresu) w którym utworzono legendę	-
z.hlegenda_text	wektor uchwytów	wektor uchwytów do elementów legendy: kolejno tekstów, linii i tła legendy	-
z.pokaz_punkty	liczba	jeśli równa jeden, na wykresach będą zaznaczone kółkami punkty danych	zmiana poprzez naciśnięcie przycisku <i>Pokaż punkty/Ukryj punkty</i>
z.zoom	liczba	jeśli równa jeden, powiększone równe jeden	zmiana poprzez naciśnięcie przycisku <i>Wyłącz zoom/Włącz zoom</i>
z.dane.wartosci	macierz komórkowa wektorów liczb	w tej macierzy przechowywane są wektory z danymi załadowanymi do programu	ładowanie przy pomocy przycisku <i>Dodaj</i> , usuwanie po naciśnięciu <i>Usuń</i>
z.dane.czas	macierz komórkowa wektorów liczb	wektory czasu odpowiadające zawartości dane.wartosci	ładowanie przy pomocy przycisku <i>Dodaj</i> , usuwanie po naciśnięciu <i>Usuń</i>
z.dane.nazwy	macierz komórkowa łańcuchów znakowych	służy do przechowywania nazw sygnałów Simulinka wraz z nazwami plików z których pochodzą dane, np. <zaspis.mat> <i>polozenie</i>	zawartość każdej nazwy zależy od pierwotnej nazwy pliku oraz nazwy sygnału w Simulinku
z.dane.bloki	macierz komórkowa łańcuchów znakowych	zawiera nazwę bloku z którego pochodził zapis, np. <i>nowy/system1/Zapisz do pliku1/Zapisz</i> oznacza, że zapis pochodzi z symulacji <i>nowy.mdl</i> , z systemu o nazwie <i>system</i> a blok zapisu nazywał się <i>Zapisz do pliku1</i>	wartość zależy od zawartości modelu symulacyjnego, jeśli dane pochodzą z programu Oscyloskop (przycisk <i>Zapisz</i> ) to nazwa bloku będzie zawierać <i>Wyeksportowane z Oscyloskop v1.0</i>
z.kanaly.wartosci	macierz komórkowa wektorów liczb	przepisane z z.dane.wartosci lub utworzone na ich podstawie wektory z wartościami wyświetlanymi na wykresie	zależne od wyboru na listach rozwijanych i wartości pól edycyjnych dla poszczególnych kanałów
z.kanaly.czas	macierz komórkowa wektorów liczb	przepisane z z.dane.czas (z pierwszej pozycji); dla zgodności z danymi zapisywanymi poprzez blok Simulinka, przyjęto takie same wektory czasu dla wszystkich kanałów	pierwsza załadowana dana (poprzez <i>Dodaj</i> ) decyduje o zawartości tej struktury
z.komunikaty	macierz komórkowa łańcuchów znakowych	dla ujednoczenia wyglądu okien dialogowych, w tej strukturze znajdują się predefiniowane komunikaty wyświetlane w oknach ostrzeżeń, komunikatów i błędów	-
CZAS	wektor liczb	wektor czasu	zawartość wektora czasu jest zdeterminowana przez wektor czasu pierwszego załadowanego pliku (lub kolejnego pliku załadowanego po opróżnieniu bufora przebiegów)
CH1 ... CH8	wektory liczb	kopie z.dane.wartosci, na ich podstawie z uwzględnieniem zawartości pól edycyjnych dla poszczególnych kanałów są tworzone wartości z.kanaly.wartosci; zmienne CH1 ... CH8 zostały utworzone aby móc bezpośrednio operować na nazwach <i>CH1 ... CH8</i>	zawartość CH1 ... CH8 jest zmieniana poprzez wybór na listach rozwijanych
x1, x2, y1, y2, dx, dy	liczby	globalne kopie wartości kursorów, odpowiednio: położenia w osi x kursora pierwszego, drugiego, położenia w osi y kursora pierwszego i drugiego, przyrost w osi x (x1-x2) oraz przyrost w osi y (y1-y2)	wartość tych liczb zmienia się w zależności od położenia kursorów na wykresie, zawartości wykresów i wyboru kanału, na który mają być nałożone kursory (lista rozwijana)

## 5. Instrukcja użytkownika.

W tym rozdziale zostanie przedstawiona instrukcja użytkownika programów. W kolejnych punktach podane zostaną wszystkie informacje niezbędne do korzystania z funkcji programu. Treść poniższych paragrafów jest identyczna z zawartością plików pomocy \*.html dostępnych w programach po naciśnięciu przycisków *Pomoc*. Z uwagi na duży łączny rozmiar plików pomocy, poniżej zawarto treść tylko tych najważniejszych.

### 5.1. Parametry napędu.

Jest to pierwszy etap w procesie przeprowadzenia eksperymentów symulacyjnych. Aby rozpocząć pracę, użyj **Plik->Załaduj plik ustawień** i wybierz plik z parametrami początkowymi.

#### 5.1.1. Pliki parametrów.

Poprawny plik z parametrami początkowymi musi zawierać deklarację `global` parametry oraz co najmniej jeden wpis do pola struktury `parametry`, np. `parametry.Ts=0.1;`. Plik z parametrami początkowymi może być zarówno skryptem Matlaba, jak i funkcją. Jeśli jest funkcją, nie powinien przyjmować żadnych parametrów, ani zwracać wartości. Przykład poprawnej deklaracji: `function silnik_1_parametry()`. Stosowanie funkcji zamiast skryptów ma tę zaletę, że oszczędza pamięć (zmienne lokalne są usuwane po wykonaniu funkcji), oraz umożliwia tworzenie lokalnych funkcji (a skrypt - nie). [Tutaj](#) znajduje się wydruk przykładowego pliku z parametrami.

Dodatkowo, plik parametrów musi mieć nazwę kończącą się na `_parametry.m`. Ciąg znaków poprzedzający `_parametry.m` stanowi trzon, który jest wykorzystywany do wybrania odpowiednich plików obliczeń regulatora i obliczeń wyników (wskaźników).

### Przykład

Jeśli użytkownik wybierze plik o nazwie `nowy_parametry.m`, to do obliczania nastaw regulatora zostanie użyty plik o nazwie `nowy_obliczenia.m` a do obliczania wyników, plik `nowy_wyniki.m`. Wszystkie trzy pliki muszą znajdować się w tym samym katalogu. Jeśli plik o odpowiedniej nazwie nie zostanie odnaleziony, zostanie wyświetlony komunikat o błędzie i ładowanie plików zostanie przerwane. Domyślnym katalogiem dla tych plików jest katalog *Obliczenia*. Plik z parametrami można edytować poprzez **Plik->Otwórz plik parametrów** (wcześniej należy go załadować).

#### 5.1.2. Zapisywanie pliku parametrów.

Poza ładowaniem i otwieraniem plików parametrów, menu Plik zawiera również pozycję Zapisz parametry [16]. Służy ona do zapisywania parametrów po zmianie ich wartości. Przy zapisywaniu muszą być spełnione następujące warunki:

- aby parametr mógł być zapisany do pliku, jego definicja (np. `parametry.Ts=0.1;`) musi znajdować się w pierwotnym pliku z parametrami; mechanizm zapisywania nie przewiduje dodawania nowych wpisów, jedynie modyfikuje istniejące;
- nowa wartość wpisywana w polu edycyjnym musi być dowolnym poprawnym wyrażeniem Matlaba, którego obliczona wartość jest liczbą;

Domyślna nazwa dla zapisywanego pliku jest taka sama, jak bieżącego pliku z parametrami (są wtedy automatycznie spełnione wymagania co do poprawnej nazwy wymienione w punkcie 1.1.). Jeśli użytkownik wybierze inną nazwę, np. *nowy\_parametry.m*, zostaną również utworzone nowe pliki obliczeń: *nowy\_obliczenia.m* oraz wyników: *nowy\_wyniki.m*. Nowo utworzone pliki są co do zawartości identyczne z plikami pierwotnymi.

### 5.1.3. Pliki deskryptorów.

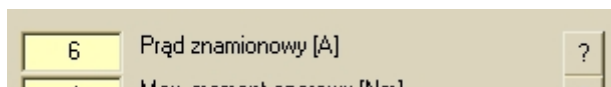
Aby w polach edycyjnych okna pojawiły się jakiekolwiek dane, należy to okno skonfigurować przy pomocy pliku deskryptora. Dla okna **1. Parametry napędu** jest to plik, którego nazwę określa globalna zmienna *nazwa\_pliku\_deskryptora\_kreator*. Nazwa ta jest zdefiniowana w pliku [Start](#). Dostęp do tego pliku można uzyskać wybierając **Plik->Otwórz plik deskryptora**. Plik deskryptora ma następującą budowę (tutaj przedstawiona jedna przykładowa linia takiego pliku):

[nazwa pola w strukturze parametry],[nazwa wyświetlana w oknie],[nazwa skojarzonego pliku pomocy],[kolor podświetlenia],

### Przykład

```
Iqn,Prąd znamionowy [A],Iqn.html,1,
```

Tak więc, jeśli plik z parametrami początkowymi zawiera wpis np. `parametry.Iqn=6;`, a plik deskryptora zawiera linię wymienioną wyżej, to w oknie pojawi się:



Plik deskryptora może definiować maksymalnie 32 parametry (32 linie) - jest to ilość dostępnych w oknie pól edycyjnych. Parametry pojawiają się w oknie w takim samym porządku, w jakim zostały zdefiniowane w pliku deskryptora (zaczynając od góry lewej kolumny). Jeśli jakiś parametr został zdefiniowany w deskrypcji a nie występuje w strukturze *parametry*, to nie zostanie wyświetlony w oknie. Wartości we wszystkich okienkach można edytować (przy czym nie muszą to być liczby, można tam wpisać każdą poprawną komendę Matlaba, w tym odwołania do innych parametrów).

### 5.1.4. Przechodzenie do obliczania parametrów regulatora.

Ostatnim elementem tego okna jest przycisk **Dalej** który powoduje uruchomienie [pliku obliczania parametrów](#) i przejście do okna **2. Parametry obliczone**.

## 5.2. Parametry obliczone.

To okno przeznaczone jest do prezentacji wyników przeprowadzonych w pliku obliczeń parametrów regulatora.

### 5.2.1. Pliki obliczeń.

Podobnie jak w przypadku parametrów początkowych, jedyne wymagania stawiane zawartości pliku obliczeń, to globalna deklaracja struktury *parametry*. Poza tym, nazwa pliku obliczeń musi mieć ten sam trzon, co nazwa pliku parametrów oraz zakończenie *\_obliczenia.m* (zobacz [przykład w punkcie 1.1](#)). Wybór nazwy pliku obliczeń jest zdeterminowany przez nazwę pliku parametrów. [Tutaj](#) znajduje się wydruk przykładowego pliku obliczeń regulatora.

### 5.2.2. Otwieranie pliku obliczeń.

Użycie polecenia **Plik->Otwórz** plik obliczeń regulatora powoduje otwarcie do edycji bieżącego pliku obliczeń nastaw regulatora. Po wprowadzeniu ewentualnych zmian w tym pliku, wystarczy cofnąć się do okna **1. Parametry napędu** i nacisnąć przycisk **Dalej**. Ponowne obliczenia zostaną przeprowadzone już na podstawie nowego pliku obliczeń.

### 5.2.3. Pliki deskryptorów.

Aby w polach edycyjnych okna pojawiły się jakiegokolwiek dane, należy to okno skonfigurować przy pomocy pliku deskryptora. Dla okna **2. Parametry obliczone** jest to plik, którego nazwę określa globalna zmienna `nazwa_pliku_deskryptora_regulator`. Nazwa ta jest zdefiniowana w pliku [Start](#). Dostęp do tego pliku można uzyskać wybierając **Plik->Otwórz plik deskryptora**. Plik deskryptora ma następującą budowę (tutaj przedstawiona jedna przykładowa linia takiego pliku):

[nazwa pola w strukturze parametry],[nazwa wyświetlana w oknie],[nazwa skojarzonego pliku pomocy],[kolor podświetlenia],

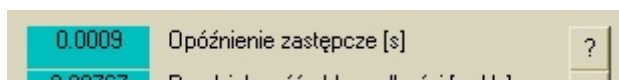
#### Przykład

```
tau,Opóźnienie zastępcze [s],tau.html,6,
```

Tak więc, jeśli plik obliczeń zapisuje jakieś pole struktury `parametry`, np.

```
parametry.tau=parametry.taug+parametry.tauf;;
```

a plik deskryptora zawiera linię wymienioną wyżej, to w oknie pojawi się:



Plik deskryptora może definiować maksymalnie 32 parametry (32 linie) - jest to ilość dostępnych w oknie pól edycyjnych. Parametry pojawiają się w oknie w takim samym porządku, w jakim zostały zdefiniowane w pliku deskryptora (zaczynając od góry lewej kolumny). Jeśli jakiś parametr został zdefiniowany w deskrypcji a nie występuje w strukturze `parametry`, to nie zostanie wyświetlony w oknie.

### 5.2.4. Cofanie do okna parametrów.

Naciśnięcie przycisku **Wstecz** w oknie obliczeń parametrów regulatora powoduje powrót do okna pierwszego - **1. Parametry napędu**. Poza tym, naciśnięcie tego przycisku spowoduje wyczyszczenie wszystkich parametrów obliczonych w pliku obliczeń. Do zdeterminowania, które zmienne mają być wyczyszczone, program wykorzystuje plik deskryptora. Wszystkie nazwy pól struktury `parametry` wymienione w tym pliku (pierwszy wyraz w linii, `tau` w przykładzie powyżej) potraktowane zostaną jako zmienne do wyczyszczenia. Czyszczenie tych zmiennych ma na celu zachowanie stabilnej pracy programu w przypadku modyfikacji pliku obliczeń w trakcie pracy programu. [Zobacz plik czyszczenia nastaw regulatora](#).

### 5.2.5. Przechodzenie do okna wyboru symulacji.

Po przejrzaniu wyników obliczeń nastaw regulatora, użytkownik może przejść do okna wyboru symulacji. Odbywa się to przez naciśnięcie przycisku **Dalej**.



## 5.3. Symulacje.

W tym oknie następuje wybór zestawu symulacji, które będą uruchomione i wariantu ich przeprowadzenia. Dodatkowo, dla każdej symulacji jest możliwe zdefiniowanie dwóch parametrów dodatkowych.

### 5.3.1. Parametry dodatkowe.

Parametry dodatkowe można zdefiniować na dwa sposoby: poprzez wpisanie ich wartości w [pliku parametrów](#) lub [pliku obliczeń](#), oraz poprzez zdefiniowanie ich w pliku deskryptora symulacji. Pliki opisowe (deskryptory) są dla okna **3. Wybór symulacji** tworzone podobnie jak dla pozostałych okien (zobacz np. [Pomoc na temat okna 1. Parametry napędu](#)). Różnice polegają na ilości wczytywanych parametrów. Pliki deskryptorów dla okna **3. Symulacje** mają następujący format:

```
[nazwa symulacji],[opis],[aktywny],[nazwa parametru 1.],[wartość parametru 1.],[nazwa parametru 2.],[wartość parametru 2.],[nazwa skojarzonego pliku pomocy],
```

### Przykład

Załóżmy, że użytkownik chce dla symulacji o nazwie te09.mdl określić czas trwania symulacji (np. 5 sekund). Niech zmienną zawierająca ten czas będzie parametry.koniec\_te09. Dodatkowo, chcemy, aby symulacja była domyślnie zaznaczona (wybrana, 1 na pozycji **[aktywny]**). Wtedy plik deskryptora (otwierany za pomocą **Plik->Otwórz plik deskryptora**) musi zawierać następującą linię:

```
te09.mdl,Czas narastania przyspieszenia do 90%,1,koniec_te09,5,,,te09.html,
```

Okno **3. Symulacje** będzie wtedy zawierać wiersz:



Dodatkowe, niewidoczne na pierwszy rzut oka informacje można odczytać z podpowiedzi po umieszczeniu kursora nad polem wyboru (nazwa pliku symulacji), lub przyciskiem ze znakiem ? (nazwa skojarzonego pliku pomocy).

Zauważ, że jeśli nie ma potrzeby definiowania dodatkowych parametrów, nie należy ich wpisywać (ani nazwy, ani wartości). Jednak należy pozostawić przecinki, które je pierwotnie rozdzielały.

Drugim sposobem na zdefiniowanie dodatkowego parametru jest umieszczenie jego definicji w [pliku parametrów](#) lub [pliku obliczeń](#). Wtedy w pliku deskryptora wystarczy pozostawić tylko jego nazwę (bez wartości):

```
te09.mdl,Czas narastania przyspieszenia do 90%,1,koniec_te09,,,,te09.html,
```

Liczba przecinków, jak poprzednio, musi się zgadzać.

Jeśli definicja parametru dodatkowego jest umieszczona w [pliku parametrów](#), to będzie możliwe zapisanie tego parametru (w ten sam sposób, jak parametry początkowe, zobacz [Pomoc na temat okna 1. Parametry napędu](#)). W przeciwnym wypadku zmienione poprzez pola edycyjne wartości

parametrów symulacji pamiętane będą tylko do momentu zakończenia programu lub ponownego załadowania pliku parametrów.

### 5.3.2. Wariant symulacji.

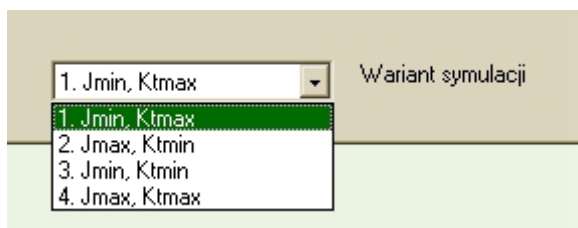
Oprócz parametrów dodatkowych, w oknie **3. Symulacje** można również wybrać wariant symulacji. Wariant symulacji jest liczbą całkowitą przechowywaną w zmiennej `parametry.wariant_symulacji` i jest używany do parametryzowania całego zestawu symulacji.

## Przykład

Zakładamy, że w symulacji występuje mechanizm wykorzystujący wartość `parametry.wariant_symulacji` (zobacz np. [Pomoc na temat bloku przełączanego wzmacnienia](#)). Założmy dodatkowo, że treść deskryptora wariantu symulacji (jego nazwa jest zawarta w zmiennej globalnej `nazwa_pliku_deskryptora_wariant` a ścieżka - w globalnej zmiennej `sciezka_deskryptorow`, obie zdefiniowane w pliku [Start](#)) jest następująca:

1. Jmin, Ktmax
2. Jmax, Ktmin
3. Jmin, Ktmin
4. Jmax, Ktmax

Wtedy w oknie **3. Wybór Symulacji** pojawi się lista rozwijana:



Wybór którejś z pozycji spowoduje przypisanie do zmiennej `parametry.wariant_symulacji` wartości odpowiadającej pozycji na liście. Pozycja początkowa (domyślna) listy jest przechowywana w pliku parametrów. Wartość początkową można zmieniać ręcznie edytując plik, lub wybierając inną pozycję na liście i korzystając z polecenia **Plik->Zapisz** parametry w oknie **1. Parametry napędu**.

### 5.3.3. Czyszczenie wyników poprzednich symulacji.

Polecenie to jest dostępne poprzez **Plik->Wyczyść** wyniki poprzednich symulacji. Spowoduje ono usunięcie plików `*.mat` z podkatalogu `Modele` (lub innego katalogu zadeklarowanego jako `sciezka_modeli`) i katalogu głównego programu. Pliki `*.mat` to pliki z uzyskanymi z symulacji danymi - zobacz [Pomoc na temat bloku Zapisz do pliku](#). W katalogu `Modele` znajdują się wyniki symulacji przeprowadzonych z poziomu kreatora symulacji, natomiast symulacje uruchomione ręcznie będą zapisywały wyniki do katalogu głównego programu.

### 5.3.4. Rozpoczęcie symulacji.

Po naciśnięciu przycisku **Dalej** rozpoczyna się przeprowadzanie symulacji. Dodatkowo, tuż przed uruchomieniem symulacji struktura `parametry` zostaje skopiowana do struktury `p`. Ułatwia to używanie nazw zmiennych w modelach symulacyjnych. Wszystkie uruchomione symulacje zostają otworzone. Po zakończeniu symulacji otwarte jest okno [4. Obliczone wyniki](#).

## 5.4. Obliczone wyniki.

W tym oknie następuje prezentacja wyników (wskaźników jakości) obliczonych na podstawie plików \*.mat wygenerowanych przez poszczególne symulacje.

### 5.4.1. Pliki wyników.

Podobnie jak w przypadku parametrów początkowych i pliku obliczeń regulatora, jedyne wymagania stawiane zawartości pliku obliczeń, to globalna deklaracja struktury `parametry`. Poza tym, nazwa pliku wyników musi mieć ten sam trzon, co nazwa pliku parametrów oraz zakończenie `_wyniki.m` (zobacz [przykład w punkcie 1.1](#)). Wybór nazwy pliku wyników jest zdeterminowany przez nazwę pliku parametrów. [Tutaj](#) znajduje się wydruk przykładowego pliku wyników.

### 5.4.2. Otwieranie pliku wyników.

Użycie polecenia **Plik->Otwórz plik obliczeń wyników** powoduje otwarcie do edycji bieżącego pliku obliczeń wskaźników. Po wprowadzeniu ewentualnych zmian w tym pliku, wystarczy cofnąć się do okna **3. Wybór symulacji** i nacisnąć przycisk **Dalej**. Ponowne obliczenia zostaną przeprowadzone już na podstawie nowego pliku wyników.

### 5.4.3. Pliki deskryptorów.

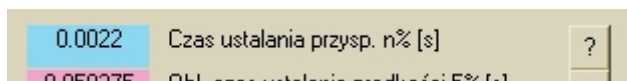
Aby w polach edycyjnych okna pojawiły się jakiegokolwiek dane, należy to okno skonfigurować przy pomocy pliku deskryptora. Dla okna **4. Obliczone wyniki** jest to plik, którego nazwę określa globalna zmienna `nazwa_pliku_deskryptora_wyniki`. Nazwa ta jest zdefiniowana w pliku [Start](#). Dostęp do tego pliku można uzyskać wybierając **Plik->Otwórz plik deskryptora**. Plik deskryptora ma następującą budowę (tutaj przedstawiona jedna przykładowa linia takiego pliku):

```
[nazwa pola w strukturze parametry],[nazwa wyświetlana w oknie],[nazwa skojarzonego pliku pomocy],[kolor podświetlenia],
```

## Przykład

```
ten,Czas ustalania przysp. n% [s],te5.html,3,
```

Tak więc, jeśli plik obliczeń wyników zapisuje jakieś pole struktury `parametry`, np. `parametry.ten=Porownanie_przyspieszenia_n.time(i);`, a plik deskryptora zawiera linię wymienioną wyżej, to w oknie pojawi się:



Plik deskryptora może definiować maksymalnie 32 parametry (32 linie) - jest to ilość dostępnych w oknie pól edycyjnych. Parametry pojawiają się w oknie w takim samym porządku, w jakim zostały zdefiniowane w pliku deskryptora (zaczynając od góry lewej kolumny). Jeśli jakiś parametr został zdefiniowany w deskrypcji a nie występuje w strukturze `parametry`, to nie zostanie wyświetlony w oknie.

### 5.4.4. Zapisywanie wyników.

Do zapisywania wyników służą dwa polecenia: **Plik->Zapisz wyniki do pliku** i **Plik->Dopisz wyniki do pliku**. Obydwa te polecenia powodują otwarcie okna dialogowego wyboru nazwy pliku \*.xls i wpisanie do arkusza Excela wszystkich pól struktury `parametry` wraz z ich wartościami [18]. Dodatkowo w pliku dodano następujące kategorie:

- Nazwa pliku parametrów:
- Nazwa pliku obliczeń:
- Nazwa pliku wyników:
- Wygenerowano:
- Komentarz:

Nazwa pliku parametrów zawiera pełną ścieżkę pliku z parametrami początkowymi, Nazwa pliku obliczeń - ścieżka do pliku obliczeń, Nazwa pliku wyników - ścieżkę do pliku wyników. Wygenerowano zawiera datę i godzinę zapisania danych. Komentarz to puste pole przeznaczone na komentarz użytkownika.

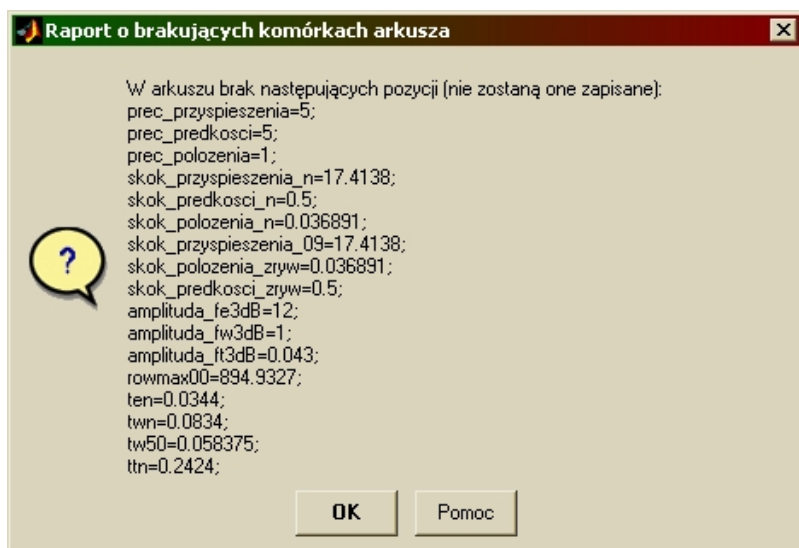
Sposób zapisu plików jest następujący: na podstawie nazw pól struktury `parametry` są utworzone kategorie (pierwsza kolumna arkusza). Następnie na odpowiednich pozycjach drugiej kolumny są wpisywane wartości parametrów. Jeśli wartości nie są skalarami, do arkusza zapisana jest ich tekstowa reprezentacja. Oznacza to, że komórki arkusza nie zawierają liczb, tylko tekst np. wektor [1 0 1] zostanie przedstawiony jako ciąg znaków "[1 0 1]" (wraz z cudzysłowem). Ostatnią operacją jest zapisanie w/w kategorii specjalnych. Podane operacje dotyczą polecenia **Plik->Zapisz wyniki do pliku**. W przypadku polecenia **Plik->Dopisz wyniki do pliku** zostanie przeprowadzone dopisanie wartości pól struktury `parametry` do istniejącego arkusza. Tak więc w istniejącym arkuszu stworzona zostanie kolejna kolumna z wartościami parametrów.

## Przykład

Poniżej przedstawiono fragment arkusza po dwukrotnym dopisaniu danych:

amplituda_fe3dB	17,4137931	12	12
amplituda_fw3dB	0,5	1	1
amplituda_ft3dB	0,036891325	0,043	0,043
ft3dB0	3,072554241	3,072554241	4
fw3dB0	10,27843442	10,27843442	10
fe3dB0	132,6288091	132,6288091	120

**Uwaga:** Z powodu dynamicznej zmiany rozmiarów struktury `parametry`, może wystąpić sytuacja, że ilość pól w zapisanym wcześniej skoroszycie będzie inna, niż ilość parametrów do dopisania. Mogą wtedy wystąpić dwie sytuacje. Po pierwsze ilość parametrów w strukturze `parametry` będzie mniejsza, niż ilość kategorii w arkuszu. Wtedy pola, których nie będzie można odczytać ze struktury `parametry` po prostu pozostaną puste. Po drugie, ilość pól struktury `parametry` przekroczy ilość kategorii arkusza. Zostanie wtedy wyświetlony komunikat o brakujących polach i zostaną wymienione nazwy pól, których nie można zapisać wraz z ich wartościami:



Problem ten jest opisany w pliku [Pomoc na temat braku komórek w arkuszu Excel](#).

#### 4.4.5. Cofanie do okna symulacji.

Naciśnięcie przycisku Wstecz w oknie obliczeń wyników powoduje powrót do okna trzeciego - **3. Wybór symulacji**. Poza tym, naciśnięcie tego przycisku spowoduje wyczyszczenie wszystkich parametrów obliczonych w pliku obliczeń wyników. Do zdeterminowania, które zmienne mają być wyczyszczone, program wykorzystuje plik deskryptora. Wszystkie nazwy pól struktury `parametry` wymienione w tym pliku (pierwszy wyraz w linii, `ten` w przykładzie powyżej) potraktowane zostaną jako zmienne do wyczyszczenia. Czyszczenie tych zmiennych jest potrzebne, aby w oknie wyników zawsze pojawiały się aktualne dane (tzn. zgodne z zawartością plików \*.mat wygenerowanych przez symulację, [zobacz plik czyszczenia wyników symulacji](#)). Jeśli w katalogu *Modele* lub katalogu głównym programu znajdują się poprawne pliki \*.mat z danymi z symulacji, to wyniki obliczone na ich podstawie zostaną wyświetlone nawet wtedy, gdy użytkownik nie uruchomił żadnej symulacji. Aby całkowicie wyczyścić pozostałości z poprzednich symulacji, należy użyć polecenia [Plik->Wyczyść wyniki poprzednich symulacji](#) znajdującego się w oknie **3. Wybór symulacji**.

#### 5.4.6. Otwieranie programu Oscyloskop.

Ostatnim elementem menu jest przycisk Oscyloskop. Służy on do uruchamiania wirtualnego oscyloskopu, w którym można przeglądać przebiegi czasowe sygnałów zarejestrowanych w czasie symulacji. Program ten korzysta z identycznego formatu danych, co plik obliczeń wyników (wskaźników jakości). Jeżeli do jego uruchomienia zostanie użyty przycisk Oscyloskop w oknie 4. Obliczone wyniki, to jego domyślna ścieżka (ścieżka na której szukane są pliki \*.mat) zostanie ustawiona na katalog *Modele* w katalogu głównym - jest to domyślny katalog, w którym zapisywane są pliki \*.mat z danymi z symulacji. [Zobacz pomoc na temat programu Oscyloskop](#).

## 5.5. Program Oscyloskop.

Program Oscyloskop został utworzony, aby można było przeglądać i obrabiać przebiegi czasowe uzyskane w czasie symulacji. Do zapisywania danych w odpowiednim formacie używany jest blok [Zapisz do pliku](#).

### 5.5.1. Uruchamianie programu.

Program może zostać uruchomiony na dwa sposoby: poprzez naciśnięcie przycisku Oscyloskop w ostatnim oknie kreatora symulacji ([Wyniki \(4/4\): Obliczone wyniki](#)), lub wywołanie z linii

poleceń. Jeśli program jest uruchamiany poprzez przycisk, to jego domyślna ścieżka zostanie ustawiona na katalog Modele w katalogu głównym Serwo-CAD. Jeśli użytkownik uruchamia program z linii poleceń, może przekazać domyślną ścieżkę poprzez parametr.

## Przykład

Załóżmy, że bieżąca ścieżka Matlab'a jest ustawiona na katalog *Oscyloskop*. Aby uruchomić program Oscyloskop z wybraną przez użytkownika ścieżką domyślną, należy w linii poleceń wpisać:

```
Oscyloskop('sciezka',string_okreslajacy_sciezke);
```

gdzie `string_okreslajacy_sciezke` jest łańcuchem znakowym zawierającym domyślną ścieżkę programu.

### 5.5.2. Wyświetlanie przebiegów.

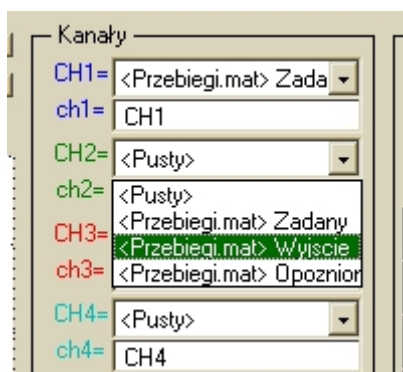
Jeśli na bieżącej ścieżce programu znajduje się chociaż jeden plik z przebiegami, będzie możliwe jego dodanie do bufora przebiegów. Lista plików jest wyświetlana w ramce o nazwie **Pliki na ścieżce**. Po wybraniu pliku, wystarczy użyć przycisku **Dodaj**. Spowoduje to wyświetlenie w ramce **Bufor przebiegów** spisu sygnałów znajdujących się w załadowanym pliku. Format wyświetlania jest następujący:

```
<nazwa_pliku> nazwa sygnału w Simulinku
```

Dzięki takiemu formatowi możliwe jest szybkie zidentyfikowanie przebiegu.

**Uwaga:** Program umożliwia ładowanie wielu plików, pod warunkiem, że mają one ten sam wektor czasu. Wektor czasu z pierwszego (tzn. załadowanego kiedy bufor przebiegów jest pusty) pliku jest użyty jako wektor czasu dla wszystkich kolejnych sygnałów. Z możliwości wyświetlania przebiegów o różnych wektorach czasu zrezygnowano, ponieważ podczas zapisu przebiegów do pliku nie została by utrzymana zgodność z formatem Simulinka ([zobacz Pomoc na temat bloku Zapisz do pliku](#)).

Aby w oknie wykresu wyświetlone zostały załadowane dane, wystarczy dla któregoś z kanałów wybrać z listy rozwijanej nazwę sygnału. Nazwa sygnału na listach rozwijanych jest identyczna, jak w **Buforze przebiegów** :



### 5.5.3. Modyfikacja przebiegów.

Oprócz listy rozwijanej wyboru sygnału, do każdego kanału oscyloskopu zostało przyporządkowane pole edycyjne. Służy ono do modyfikowania zawartości wyświetlanych kanałów. Podczas kiedy zmienne `CH1..CH8` przechowują dane wybrane z listy rozwijanej, zawartość wykresu zależy od wartości zmiennych `ch1..ch8` (zobacz obrazek powyżej). Domyślnie,

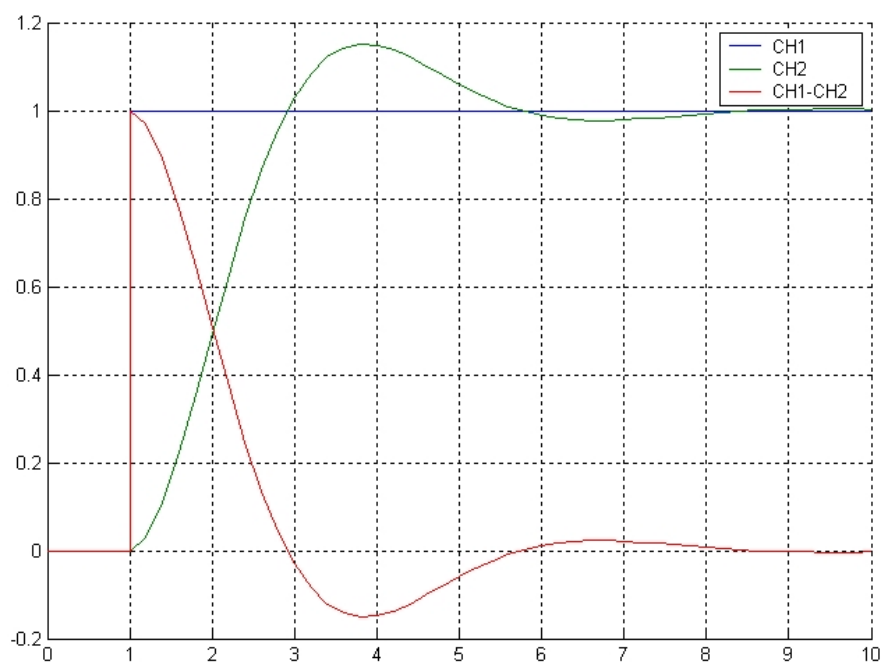
zawartość kanałów  $ch1 \dots ch8$  jest równa zawartości zmiennych  $CH1 \dots CH8$  czyli w efekcie równa danym wybranym na listach rozwijanych. Jednak po wpisaniu poprawnego wyrażenia Matlab'a w polach edycyjnych poszczególnych kanałów, można modyfikować zawartość wykresu.

## Przykład

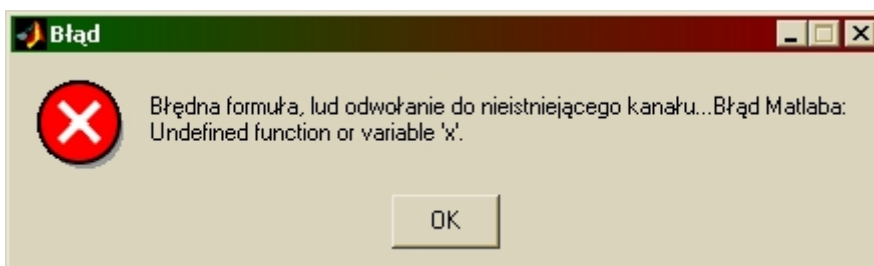
Załóżmy, że w dostępnych danych występuje przebieg wartości zadanej i wyjściowej układu regulacji. Chcąc obejrzeć przebieg uchybu, należy do dwóch kanałów załadować odpowiednio przebieg wartości zadanej i wyjściowej, a w polu edycyjnym trzeciego wpisać różnicę tych kanałów:



Wykres będzie zawierał wtedy:

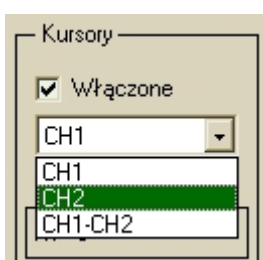


W polach edycyjnych można wpisywać dowolne wyrażenie Matlab'a odnoszące się do zmiennych  $CH1 \dots CH2$  oraz zmiennej  $CZAS$ . Nie jest jednak możliwe odwoływanie rekurencyjne, tzn. nie można we wpisanych wyrażeniach korzystać ze zmiennych  $ch1 \dots ch8$  (czyli wyników wyrażeń innych pól). Z możliwości tej zrezygnowano dla zachowania czytelności i pewności działania mechanizmu. Jeśli wpisana formuła będzie niepoprawna, zostanie wyświetlony następujący komunikat:

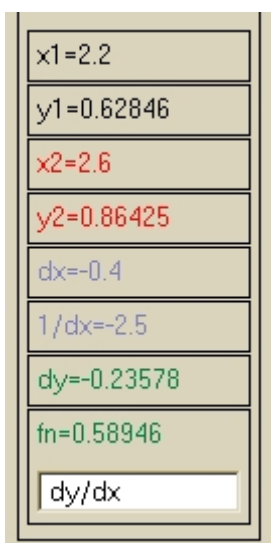


#### 5.5.4. Używanie kursorów.

Podobnie jak w prawdziwym oscyloskopie, w programie Oscyloskop zaimplementowano kursory. Kursory służą do odczytywania wartości sygnału w danej chwili czasu. Dostępne są dwa kursory. Obydwa kursory zawsze dotyczą tego samego przebiegu. Aby włączyć lub wyłączyć wyświetlanie kursorów, zaznacz lub odznacz pole **Włączone** w ramce **Kursory**. Aby wybrać przebieg, którego mają dotyczyć kursory, użyj listy rozwijanej w ramce **Kursory**:



W ramce **Kursory** wyświetlone są wartości związane z kursorami:



Są to kolejno:

- $x_1$  - wartość w osi x (czasu) kursora pierwszego (czarnego)
- $y_1$  - wartość w osi y kursora pierwszego
- $x_2$  - wartość w osi x kursora drugiego (czerwonego)
- $y_2$  - wartość w osi y kursora drugiego
- $dx$  - różnica  $x_1 - x_2$
- $1/dx$  - odwrotność  $dx$
- $dy$  - różnica  $y_1 - y_2$
- $f_n$  - wartość wyrażenia wpisanego w dolnym polu edycyjnym



W w/w polu edycyjnym można wpisać dowolne wyrażenie Matlab, które odwołuje się do zmiennych  $x_1$ ,  $x_2$ ,  $y_1$ ,  $y_2$ ,  $dx$ ,  $dy$ .

Do poruszania kursorów służą suwaki umieszczone ponad wykresem. Dodatkowo, jeśli użytkownik zaznaczy pole obok suwaka, będzie można użyć myszy do przesuwania kursora po wykresie. Po wciśnięciu lewego klawisza myszy kursor zostaje zatrzymany a przeciąganie myszą - wyłączone.

### 5.5.5. Zapisywanie przebiegów.

Obrobione przez użytkownika kanały można zapisać w nowym pliku. Plik ten jest zgodny z formatem obsługiwanym przez program, więc można go ładować jako nową daną do programu. Do pliku zapisywane są aktualnie wyświetlone przebiegi. Użytkownik wybiera nazwę pliku \*.mat (z rozszerzeniem lub bez) po czym ma możliwość utworzenia własnych opisów do kanałów:



The image shows a dialog box with a title bar that reads "Podaj nazwy dla zapisywanych kanałów (bez znaków < > ,)". Inside the dialog, there are four text input fields labeled "Kanal 1:", "Kanal 2:", "Kanal 3:", and "Kanal 4:". The first three fields contain the text "CH1", "CH2", and "CH3" respectively. The fourth field contains "CH3.\*CH2". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Opisy te są później wyświetlane w buforze przebiegów co umożliwia identyfikację zawartości plików. Jeśli użytkownik nie poda własnych opisów kanałów, do pliku będą zapisane te same nazwy, które wyświetlone są na legendzie (wynikają one z zawartości pól edycyjnych poszczególnych kanałów). Dodatkowo, po zatwierdzeniu nazw kanałów użytkownik zostanie zapytany, czy ma być utworzony plik z komentarzem do danych. Jeśli odpowiedź będzie pozytywna, na tej samej ścieżce, na której zapisano plik \*.mat zostanie utworzony plik tekstowy o takiej samej nazwie (i rozszerzeniu txt). Zostaje on automatycznie otworzony. Dla łatwiejszej orientacji, w pierwszej linii tego pliku umieszczona jest data i godzina utworzenia.

### 5.5.6. Narzędzia.

Na pasku narzędzi umieszczono dodatkowe narzędzia. Poniżej umieszczono opis ich działania.

#### Zoom.

Jest to narzędzie umożliwiające powiększanie zawartości wykresu. Działa ono niemal identycznie do wbudowanego w standardowe figury Matlab. Różnica polega na tym, że oprócz przycisku **Włącz zoom/Wyłącz zoom** dodano również przycisk **Zoom reset** umożliwiający powrót do takiego ustawienia zakresu osi, aby wszystkie dane były widoczne. Włączanie i wyłączanie powiększenia jest pomocne wtedy, kiedy podczas powiększenia wykresu chcemy użyć myszy do przeciągania kursorów. Jeśli nie wyłączylibyśmy powiększenia, to zamiast zaznaczenia żądanej pozycji kursora nastąpiłoby kolejne powiększenie.

#### Pokaż punkty.

Ten przycisk umożliwia pokazywanie/ukrywanie punktów danych na wykresach. Dzięki pokazywaniu punktów danych możliwe jest np. określenie, jaki był okres próbkowania, lub mierzenie zmian zachodzących w pojedynczym kroku czasowym.

### **Drukowanie.**

Ponieważ opcja drukowania wbudowana w Matlaba dotyczy całej figury, wydruki sporządzone bezpośrednio z programu byłyby nieczytelne. Aby temu zaradzić, można utworzyć nowe okno, do którego skopiowane są wykresy wraz z kursorami i legendą. W tym oknie można również używać narzędzi typowych dla podstawowej figury Matlaba, a więc powiększania, kopiowania do schowka, drukowania, dodawania opisów i innych. Aby otworzyć okno drukowania wystarczy nacisnąć przycisk **Drukowanie**.

### **Pomoc.**

Wciśnięcie tego przycisku powoduje otwarcie tego właśnie pliku pomocy. Aby plik mógł być wyświetlony, katalog *Oscyloskop* musi znajdować się na tej samej ścieżce, co plik *Pomoc.m* oraz katalog z plikami pomocy - *Pomoc*. Warto o tym pamiętać wtedy, kiedy program *Oscyloskop* ma być używany niezależnie od pakietu *Serwo-CAD*.

## **5.6. Blok Zapisz do pliku.**

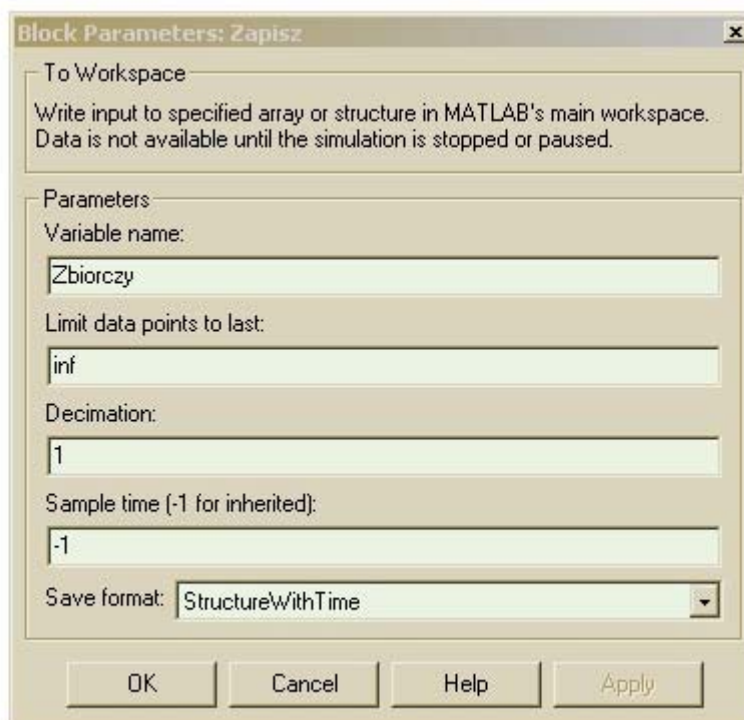
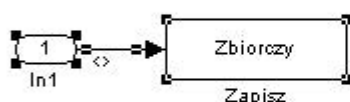


Zapisz do pliku

Blok ten został stworzony na potrzeby programu *Oscyloskop* i pliku obliczeń wyników (wskaźników jakości) układu regulacji. Służy do zapisywania przebiegów czasowych symulacji w plikach \*.mat. Można go dodać do symulacji poprzez przeglądarkę bibliotek: *Bloki dodatkowe użytkownika* -> *Zapisz do pliku*. Aby biblioteka była widziana w przeglądarce, katalog *Modele* musi zostać dodany do ścieżki (lub być bieżącym katalogiem).

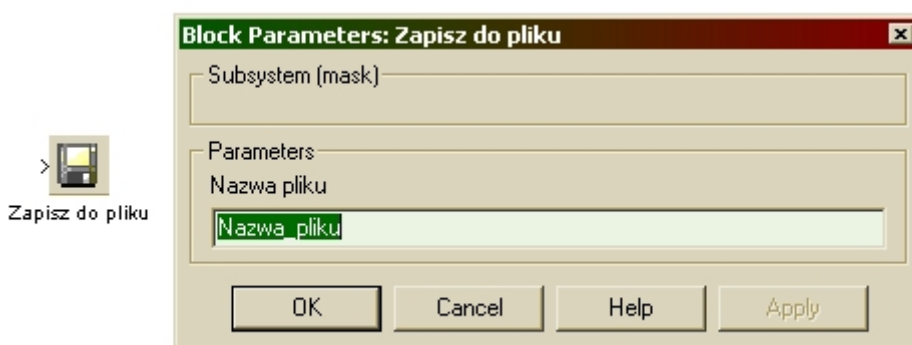
### **5.6.1. Sposób zapisu danych.**

Blok wykorzystuje standardowy bloczek służący zapamiętywaniu danych w zmiennej: *simout* [17]. W bloku tym wybrano opcję zapisu *StructureWithTime* (struktura wraz z wektorem czasu):



### 5.6.2. Podawanie nazwy pliku.

Aby zdefiniować nazwę pliku w którym mają zostać zapisane przebiegi, należy dwukrotnie kliknąć blok, co spowoduje wyświetlenie następującego okna dialogowego:



**Uwaga:** ponieważ nazwa pliku jest identyczna z nazwą zmiennej tymczasowej, musi ona być poprawną nazwą zmiennej Matlaba (zobacz pomoc funkcji `isvarname`). W szczególności, nazwa pliku nie powinna zawierać rozszerzenia `*.mat` - zostaje ono dopisane automatycznie.

### 5.6.3. Format zapisu.

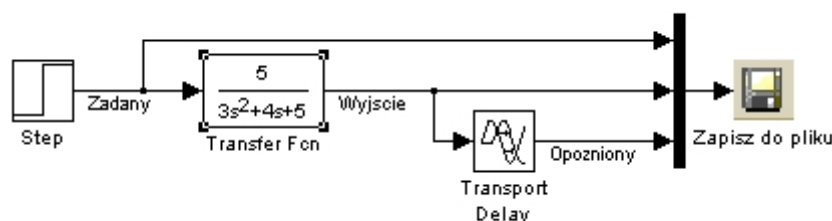
Aby umożliwić użytkownikowi korzystanie z danych zapisywanych w plikach, lub preparowanie ich z innego rodzaju danych, poniżej przedstawiony jest format zapisu danych bloczka *Zapisz do pliku*. Zakładamy, że wartości w poszczególnych kanałach są skalarnie (tzn. że w danej chwili czasu, każdy kanał jest reprezentowany przez pojedynczą liczbę):

- `nazwa_pliku.time`: wektor czasu o wymiarze  $N \times 1$ ; zawiera wartości czasu w kolejnych krokach symulacji

- `nazwa_pliku.signals.values`: macierz o wymiarze  $N \times M$ , wartości poszczególnych sygnałów;  $N$  jest ilością próbek,  $M$  - ilością zarejestrowanych sygnałów
- `nazwa_pliku.signals.dimensions`: liczba określająca ilość kanałów ( $M$ )
- `nazwa_pliku.signals.label`: zawiera nazwy poszczególnych sygnałów (zobacz przykład poniżej)
- `nazwa_pliku.blockName`: nazwa bloku w którym zapisano dane

## Przykład

Poniżej przedstawiony został przykład wykorzystania bloku do zapisania trzech zmiennych.



Jak widać, każdemu sygnałowi nadana została nazwa. Umożliwia to późniejszą identyfikację kanałów. Po przeprowadzeniu symulacji, na bieżącej ścieżce Matlaba zostanie utworzony plik o nazwie *Przebiegi* (nazwa wpisana w bločku) i rozszerzeniu *mat*. Plik ten zawiera jedną zmienną (strukturę) o nazwie *Przebiegi*. Można ją załadować przy użyciu polecenia `load Przebiegi`.

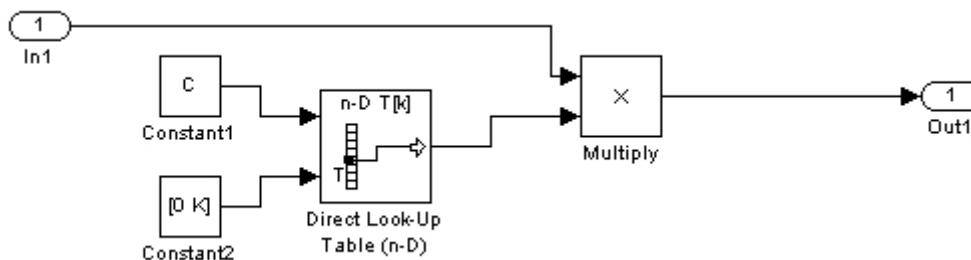
## 5.7. Blok Przelączone wzmocnienie



Jest to blok umożliwiający zmianę wzmocnienia w zależności od zmiennej typu całkowitego. Jest wykorzystywany podczas zmiany wariantu symulacji.

### 5.7.1. Budowa.

Blok wykorzystuje tabelę przeglądową. Dodatkowo jest zamaskowany, aby można było w wygodny sposób podawać wartości wzmocnień i nazwę zmiennej przełączającej:

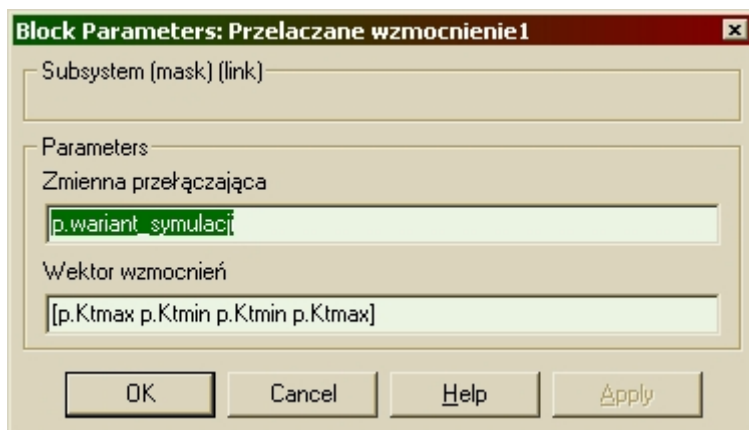


Ponieważ wbudowany w Simulinkę blok *Direct Look-Up Table (n-D)* wykorzystuje indeksowanie od zera, wektor wzmocnień przesunięto o jedną pozycję. Ma to na celu uzyskanie zgodności ze

stosowanym w Matlabie sposobem indeksowania od jeden (czyli kiedy zmienna przełączająca  $c$  jest równa jeden, wybrany zostaje pierwszy element z wektora wzmacnień).

### 5.7.2. Podawanie parametrów.

Okno dialogowe podawania parametrów wygląda następująco:



Na rysunku powyżej pokazano przykład wykorzystania bloku w roli zmiennego wzmacnienia stałej momentu silnika. Zmienna przełączająca może być dowolną liczbą całkowitą przyjmującą wartości od 1 do ilości elementów wektora wzmacnień. Wektor wzmacnień (1 x N czyli wierszowy) może być dowolnej długości.

## 6. Zastosowanie oprogramowania do projektowania odpornego regulatora serwonapędu.

W tym rozdziale zostanie przedstawiony przykład wykorzystania opisywanego oprogramowania do syntezy regulatora serwonapędu. Projekt ten w pełni wykorzystuje wszystkie możliwości programów. Omówione zostaną wszystkie etapy od podania parametrów początkowych do analizy wskaźników jakości.

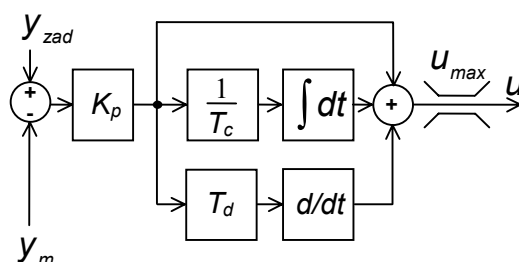
### 6.1. Koncepcja odpornego regulatora prędkości i położenia dla serwonapędu.

Regulacja odporna jest rozwiązaniem problemu zmiennych parametrów obiektu. Na potrzeby regulacji odpornej, w pracach [1], [2] i [3] zostały przedstawione metody doboru struktur i nastaw regulatorów gwarantujących zadowalające wyniki sterowania dla całego zakresu zmienności parametrów obiektu. Są to struktury wykazujące różne działanie w zależności od tego, czy zmianie ulega wartość zadana, czy uchyb regulacji (opis podobnego regulatora można znaleźć np. w [4], str. 76).

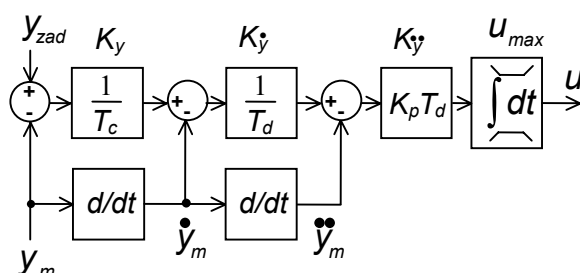
Na potrzeby sterowania serwonapędami, regulatorom stawiane są następujące wymagania:

- zerowy uchyb ustalony;
- brak przeregulowania w odpowiedzi skokowej (regulacja położenia);
- przebieg wartości wyjściowej (najczęściej położenia) ma być zbliżony do czasooptymalnego;
- prędkość dochodzenia do wartości zadanej jest niezależna od zmieniających się (w założonych granicach) parametrów.

Aby poprawić działanie regulatora w nieliniowym zakresie pracy (kiedy jego sygnały wchodzi w ograniczenia) stosuje się przekształcenie podstawowej formy regulatora przedstawionej na rysunku 16 do postaci z rysunku 17 (zaczepnięte z [1]):



Rysunek 16 - konwencjonalna struktura regulatora PID



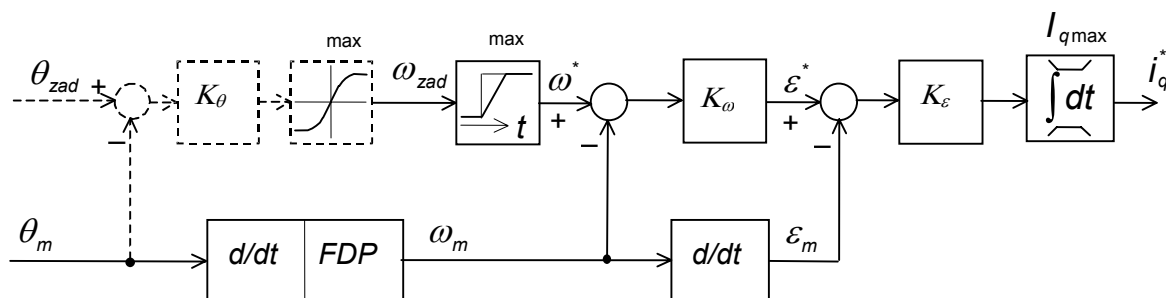
Rysunek 17 - zmodyfikowana struktura PID

Powyższy układ ma następujące zalety:

dzięki przesunięciu członu całkującego na koniec toru przekształcania sygnału wystarczy tylko jeden ogranicznik;

- wychodzenie regulatora z ograniczenia odbywa się z wyprzedzeniem, zanim uchyb regulacji prędkości będzie zerowy; dzięki temu dochodzenie do stanu ustalonego prędkości po wyjściu regulatora z ograniczenia odbywa się bez przeregulowania - inaczej niż w poprzedniej strukturze.

Końcową wersją regulatora jest następująca struktura nieliniowa z rysunku 18:



Rysunek 18 - zmodyfikowana struktura z blokami nieliniowymi w regulatorze położenia

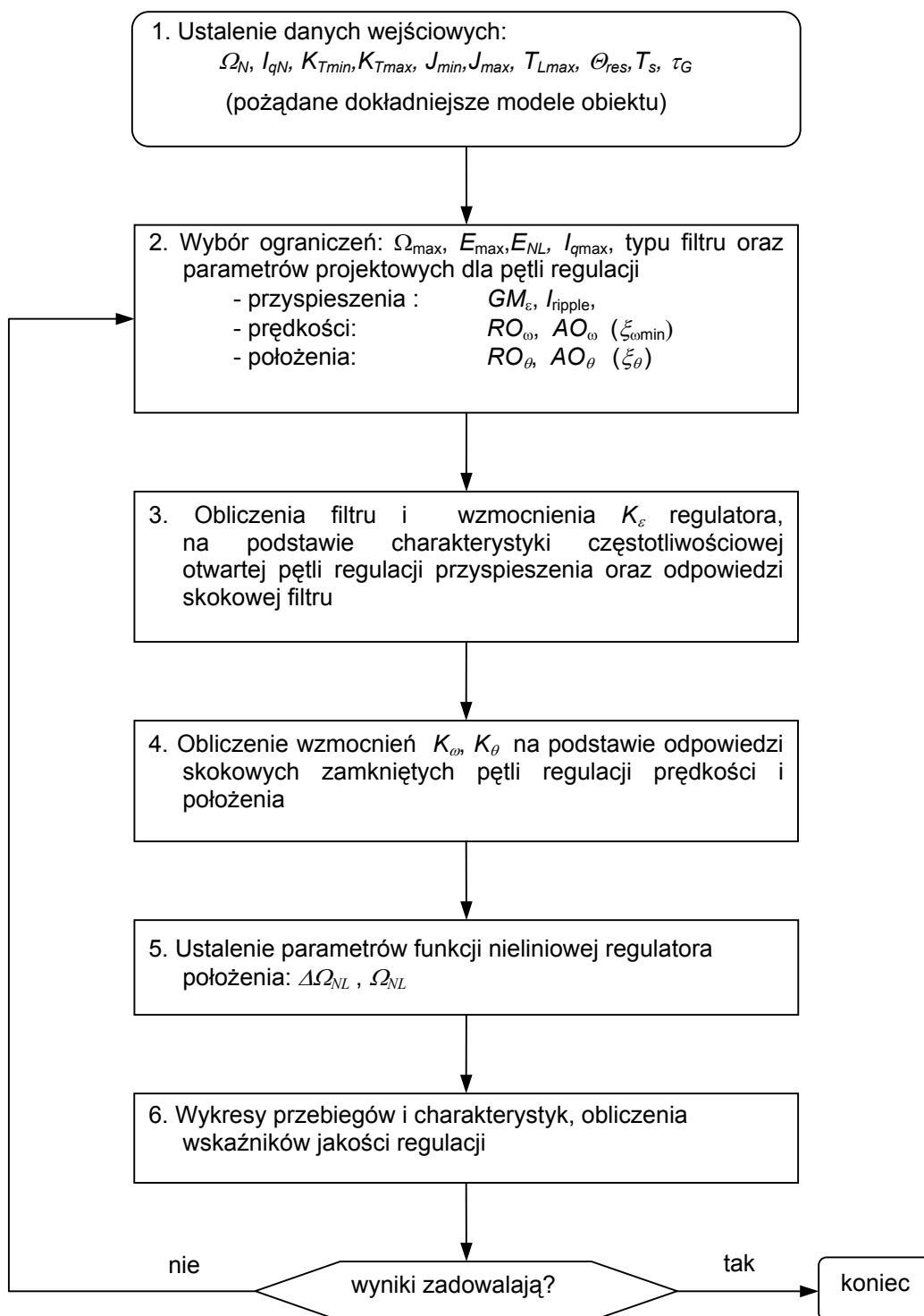
Występują tu 3 dodatkowe bloki:

- nieliniowy człon statyczny w regulatorze położenia, zapewniający ograniczenie prędkości zadanej  $\omega_{zad}$  do wartości  $\omega_{max}$  oraz brak przeregulowania po dużej, skokowej zmianie położenia zadanego  $\theta_{zad}$ ;
- ogranicznik dynamiki zmian prędkości zadanej (czyli przyspieszenia) do wartości  $\epsilon_{max}$ , zapewniający niezależność czasu dochodzenia do ustalonej prędkości od momentu oporowego i momentu bezwładności napędu;
- filtr dolnoprzepustowy (FDP) korygujący rezonansową charakterystykę mechaniczną napędu oraz zmniejszający pulsacje sygnału  $\omega_m$  wywołane skończoną rozdzielczością przetwornika mierzonego położenia  $\theta_m$ . Zadaniem tego filtra jest zapewnienie dużego tłumienia w paśmie zaporowym i małego opóźnienia w paśmie przepustowym. Opóźnienie

wnoszone przez filtr jest jednym z najważniejszych składników całkowitego opóźnienia w pętli regulacji prędkości  $\tau$ .

## 6.2. Algorytm projektowania regulatora.

Do projektowania regulatora stosowany jest algorytm z rysunku 19 [2]:

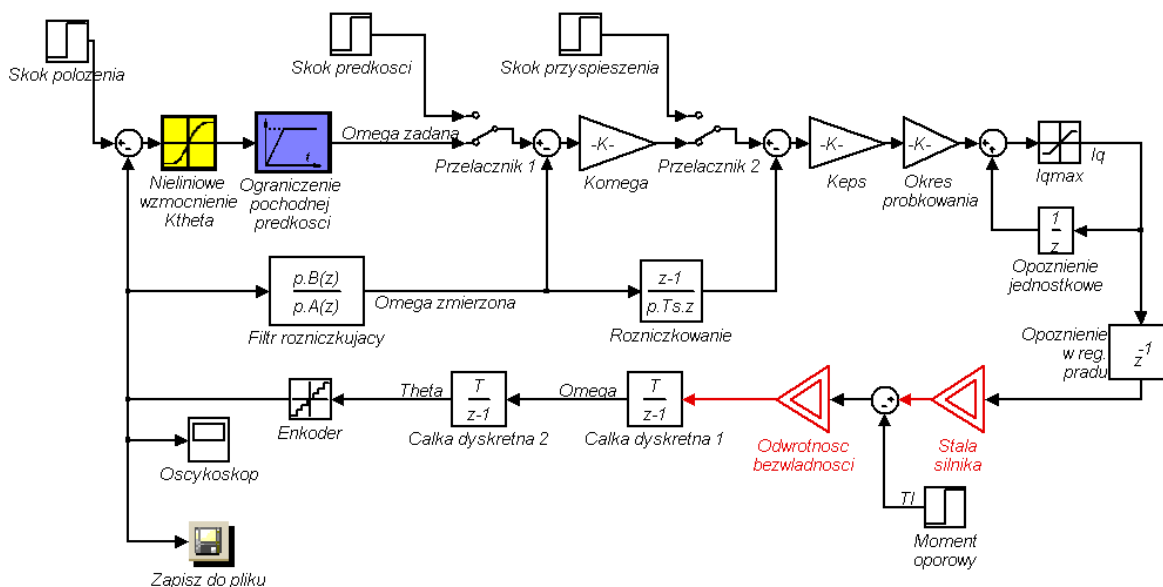


Rysunek 19 - schemat procesu projektowania regulatora



### 6.3. Przygotowanie symulacji.

Symulacja będzie dotyczyć regulatora prędkości i położenia dla synchronicznego silnika momentowego o magnesach trwałych. Na rysunku 20 przedstawiony jest diagram symulacyjny.



Rysunek 20 - podstawowy diagram symulacyjny

Jak widać zastosowano model dyskretny zarówno regulatora, jak i silnika. Do zamodelowania filtra użyto standardowego bloku transmitancji dyskretniej. Enkoder przedstawiono za pomocą bloku *Quantizer*. Dla umożliwienia wygodnej zmiany wariantu symulacji (ze względu na wartości zmiennych parametrów napędu) użyto bloczków przełączanego wzmocnienia (*Odwrótność bezwładności* oraz *Stala silnika*). Dodatkowo, do zapamiętywania danych otrzymanych w symulacji służy blok *Zapisz do pliku*. Bloki te opisano odpowiednio w rozdziałach 5.6 oraz 5.7.

Do przeprowadzenia symulacji wykorzystano parametry zawarte w tabeli 4 (definicje tych parametrów znajdują się w pliku *Servo\_1\_parametry.m*<sup>1</sup>):

<sup>1</sup> źródło pliku wprowadzania parametrów zostało przedstawione w 7. Załącznik

Tabela 4 - zestawienie parametrów napędu

Parametr	Symbol	Nazwa	Wartość
Okres próbkowania regulatora (jednocześnie okres dyskretyzacji modelu silnika)	$T_s$	Ts	0,0001 [s]
Rozdzielczość enkodera	$\Theta_{res}$	Thetares	512000 [1/obr] lub 0,000012272 [rad]
Nominalna wartość prądu silnika	$I_{qN}$	Iqn	6 [A]
Maksymalna wartość momentu oporowego	$T_{Lmax}$	Tlmax	4 [Nm]
Opóźnienie wnoszone przez obwód regulacji prądu	$\tau_G$	taug	0,0001 [s]
Minimalna oczekiwana wartość zapasu wzmocnienia w regulatorze przyspieszenia	$GM_{\min}$	GMepsmin	3,1416
Maksymalna wartość prądu silnika	$I_{qmax}$	Iqmax	6 [A]
Maksymalna wartość tętnień prądu	$I_{ripple}$	Iripple	0,2 [A]
Nominalna prędkość obrotowa silnika	$\Omega_N$	Omegan	0,5 [rad/s]
Maksymalna wartość stałej momentu silnika	$K_{Tmax}$	Ktmax	17,5 [Nm/A]
Minimalna wartość stałej momentu silnika	$K_{Tmin}$	Ktmin	17,5 [Nm/A]
Maksymalna wartość momentu bezwładności wirnika	$J_{\min}$	Jmin	0,75 [kgm <sup>2</sup> ]
Maksymalna wartość momentu bezwładności wirnika	$J_{\max}$	Jmax	5,8 [kgm <sup>2</sup> ]
Założona maksymalna wartość przeregulowania bezwzględnego w pętli regulacji prędkości	$AO_{\omega}$	AOomega	0,05 [rad/s]
Założona maksymalna wartość przeregulowania względnego w pętli regulacji prędkości	$RO_{\omega}$	ROomega	0,1
Założona maksymalna wartość przeregulowania bezwzględnego w pętli regulacji położenia	$RO_{\theta}$	ROtheta	0
Początkowa wartość współczynnika tłumienia pętli regulacji położenia	$\xi_{\theta}$	ksitheta0	0,85

## 6.4. Obliczenia regulatora.

Po wprowadzeniu założonych parametrów napędu przechodzimy do obliczenia parametrów regulatora. Do ich przeprowadzenia użyta zostanie procedura przedstawiona w tabelach 5 i 6 [2]:

Tabela 5 – projektowanie regulatora przyspieszenia i filtru

Dobór regulatora przyspieszenia oraz filtru w torze obliczania prędkości		
Działanie	Symbole i wzory	objaśnienia
1. Ustalenie danych wejściowych	$\Theta_{res}$ - $T_s$ - $I_{qN}$ - $K_{Tmax}$ - $J_{min}$ - $T_{Lmax}$ - $\tau_G$ -	błąd rozdzielczości pomiaru kąta [rad] okres próbkowania i obliczeń [s] znamionowy prąd czynny [A] największa wartość stałej momentu [Nm/A] najmniejszy moment bezwładności [kgm <sup>2</sup> ] największa wartość momentu oporowego [Nm] opóźnienie [s] w pętli regulacji przyspieszenia (poza filtrem)
2. Wybór parametrów projektowych	$I_{ripple} \ll I_{qmax} - \frac{T_{Lmax}}{K_{Tmin}}$	(warunek zachowania astatyzmu regulacji)
3. Oszacowanie opóźnienia filtru	$SR\tau \approx 0.5$ - $\tau_F \approx \sqrt{SR\tau \cdot \frac{\pi}{2} \cdot \Theta_{res} \cdot \frac{J_{min}}{K_{Tmax}} \cdot \frac{1}{I_{ripple}} \cdot \frac{1}{GM_{\epsilon min}} + \left(\frac{\tau_G}{2}\right)^2} - \frac{\tau_G}{2}$	iloczyn opóźnienia przez maksimum odpowiedzi skokowej opóźnienie (s)
4. Projekt filtru (wzory przykładowe, dla prostego filtru różniczkującego FIR)	$N_F > \frac{2 \cdot \tau_F}{T_s}$ $b_i = \frac{1}{N_F T_s}, 0, \dots, 0, -\frac{1}{N_F T_s}$ $\tau_F = N_F \frac{T_s}{2}$ $SR_{max} = \frac{1}{N_F \cdot T_s}$ $\Omega_{res} = \Theta_{res} \cdot SR_{max}$	rząd filtru różniczkującego współczynniki filtru różniczkującego opóźnienie wprowadzane przez filtr maksimum odpowiedzi skokowej błąd rozdzielczości wyliczanej prędkości
5. Obliczenie $K_\epsilon$	$\tau = \tau_G + \tau_F$ $\nu_{\epsilon max} = \frac{2\pi}{4\tau} \cdot GM_{\epsilon min}^{-1}$ $K_\epsilon = \nu_{\epsilon max} \cdot \frac{J_{min}}{K_{Tmax}}$	całkowite opóźnienie w pętli przyspieszenia najwyższa częstotliwość odcięcia (rad/s) wzmacnienie regulatora (As)
6. Sprawdzenie	$\Omega_{res} \cdot K_\epsilon < I_{ripple}$	- powrót do punktu 4 lub 2, gdy nie spełnione

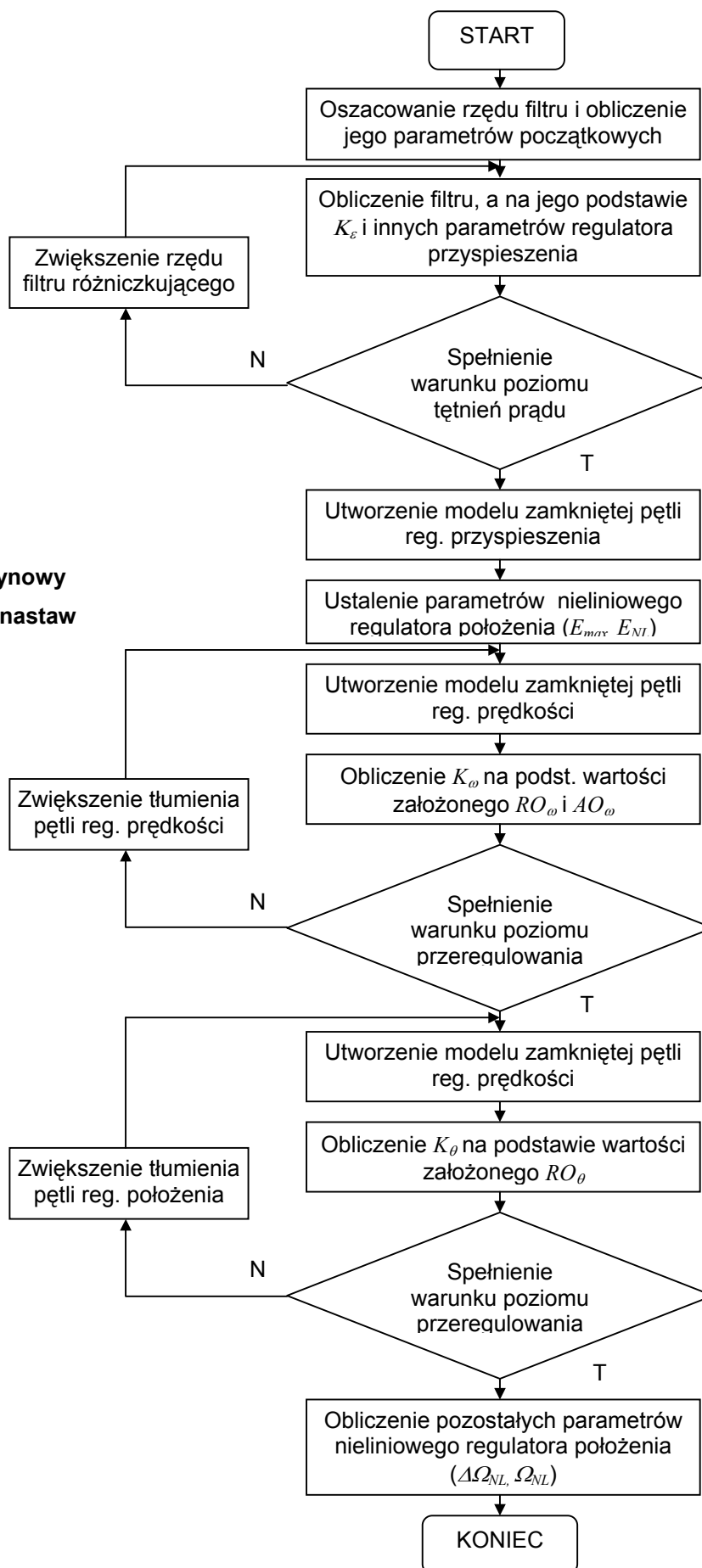
Tabela 6 - projektowanie regulatorów prędkości i położenia

Dobór parametrów regulatorów prędkości i położenia kąowego		
Działanie	Symbole i wzory	objaśnienia
1. Dane wejściowe	$v_{\varepsilon \max}$ - $\Omega_N$ - $K_{T \min}, K_{T \max}$ - $J_{\min}, J_{\max}$ - $T_{L \max}$ - $PVR = \frac{K_{T \max} \cdot J_{\max}}{K_{T \min} \cdot J_{\min}}$ - $v_{\varepsilon \min} = \frac{v_{\varepsilon \max}}{PVR}$ -	najwyższa częstotliwość odcięcia pętli regulacji przyspieszenia prędkość znamionowa [rad/s] najmniejsza i największa wartość stałej momentu [Nm/A] największy i najmniejszy moment bezwładności [kgm <sup>2</sup> ] największa wartość momentu oporowego [Nm] stosunek zmienności parametrów (Parameter Variation Ratio) najniższa częstotliwość odcięcia pętli regulacji przyspieszenia
2. Wybór ograniczeń	$\Omega_{\max} \approx \Omega_N$ - $E_{\max} \leq \frac{K_{T \min} \cdot I_{q \max} - T_{L \max}}{J_{\max}}$ - $E_{NL} \approx E_{\max}$ -	ograniczenie prędości zadanej w regulatorze położenia kąowego ograniczenie pochodnej prędości zadanej przyspieszenie kształtowane przez regulator położenia kąowego
3. Wybór parametrów projektowych: $\xi_{\omega}, \xi_{\theta}$	$\xi_{\text{omin}}$ - $\xi_{\theta}$ - (związki między współczynnikiem tłumienia $\xi$ a przeregulowaniem RO, AO w odpowiedzi skokowej) $\xi = \max(\xi_1(RO), \xi_2(AO))$ $RO = \exp\left(-\frac{\xi_1 \cdot \pi}{\sqrt{1 - \xi_1^2}}\right); \quad \xi_1 = \frac{-\ln(RO)}{\sqrt{\pi^2 + \ln^2(RO)}}$ $AO_{\omega} = \frac{E_{\max}}{v_{\varepsilon \min}} \cdot 2\xi_{2\omega} \cdot \exp\left(-\frac{\xi_{2\omega} \cdot (\pi - \arccos(\xi_{2\omega}))}{\sqrt{1 - \xi_{2\omega}^2}}\right)$	najmniejszy współczynnik tłumienia w pętli regulacji prędkości współczynnik tłumienia w pętli regulacji położenia RO - przeregulowanie względne; zakres liniowy, zerowe war. początkowe AO <sub>ω</sub> - przeregulowanie prędkości ω (rad/s) po wyjściu ze strefy ε = E <sub>max</sub> AO <sub>θ</sub> - przeregulowanie kąta θ (rad) po wyjściu z strefy ε ≈ E <sub>NL</sub> Z uwagi na mocno uwikłaną postać wzorów nie są przeprowadzane obliczenia ξ <sub>θ</sub> ze względu na RO <sub>θ</sub> i AO <sub>θ</sub> (zob. algorytm z rysunku 21 na str. 55).

4. Obliczenie parametrów regulatorów	$K_{\omega} = \frac{v_{\varepsilon \min}}{4 \cdot \xi_{\omega \min}^2}$	-	wzmocnienie proporcjonalnego regulatora prędkości
	$K_{\theta} = \frac{K_{\omega}}{4 \cdot \xi_{\theta}^2}$	-	wzmocnienie części liniowej regulatora kąta
	$\Delta \Omega_{NL} = \frac{E_{NL}}{2 \cdot K_{\theta}}$	-	obniżenie pierwiastkowej części charakterystyki
	$\Omega_{NL} = \Delta \Omega_{NL}$	-	najniższa prędkość na charakterystyce nieliniowej

Powyższe wzory zostały zastosowane do zbudowania algorytmu obliczeń (przedstawia on kroki 3., 4. i 5. algorytmu projektowania regulatora) zamieszczonego na rysunku 21

Rysunek 21 - maszynowy algorytm obliczania nastaw regulatora



Obliczenia przeprowadzane są w pliku *Serwo\_1\_obliczenia.m*. Jego kod umieszczony został w Załączniku.

## 6.5. Zestawienie dostępnych eksperymentów.

Do sprawdzenia jakości działania zaprojektowanego regulatora służy zestaw czternastu eksperymentów symulacyjnych. Zostały one wybrane z kilkudziesięciu możliwych ze względu na łatwość analizy wyników, możliwość porównania wyników z wartościami założonymi na etapie projektowania oraz ich przydatność w praktycznym rozwiązaniu serwonapędu. Tabela 5 prezentuje nazwy symulacji, ich przeznaczenie oraz dodatkowe parametry symulacji, które może wprowadzić użytkownik.

Tabela 7 - zestawienie dostępnych symulacji

Nazwa wyznaczanego wskaźnika	Sym-bol	Nazwa pliku symulacyjnego	Przeznaczenie	Parametry
Czas ustalania przyspieszenia n%	$t_{en\%}$	ten.mdl	obliczanie czasu ustalania z dokładnością n% wartości przyspieszenia po skoku wartości zadanej przyspieszenia	$skok\_przyspieszenia\_n$ – [rad/s <sup>2</sup> ] wartość zadana przyspieszenia  $prec\_przyspieszenia$ – [%] dokładność docelowa
Czas ustalania prędkości n%	$t_{en\%}$	twn.mdl	obliczanie czasu ustalania z dokładnością n% wartości prędkości po skoku wartości zadanej prędkości	$skok\_predkosci\_n$ – [rad/s] wartość zadana prędkosci  $prec\_predkosci$ – [%] dokładność docelowa
Czas ustalania położenia n%	$t_{en\%}$	ttn.mdl	obliczanie czasu ustalania z dokładnością n% wartości położenia po skoku wartości zadanej położenia	$skok\_polozenia\_n$ – [rad] wartość zadana położenia  $prec\_przyspieszenia$ – [%] dokładność docelowa
Czas narastania przyspieszenia do 90%	$t_{e09}$	te09.mdl	obliczanie czasu narastania przyspieszenia do 90% wartości zadanej	$skok\_przyspieszenia\_09$ – [rad/s <sup>2</sup> ] wartość zadana przyspieszenia
Czas narastania prędkości do 90%	$t_{e09}$	tw09.mdl	obliczanie czasu narastania prędkości do 90% wartości zadanej	$skok\_predkosci\_09$ – [rad/s] wartość zadana prędkosci
Czas narastania położenia do 90%	$t_{e09}$	tt09.mdl	obliczanie czasu narastania położenia do 90% wartości zadanej	$skok\_polozenia\_09$ – [rad] wartość zadana położenia
IE prędkości po skoku obciążenia	$IE_{\omega}$	IEw.mdl	całka z uchybu regulacji prędkości po skoku obciążenia	$skok\_obc\_predkosc$ – [Nm] wartość skoku momentu oporowego

IE położenia po skoku obciążenia	$IE_{\theta}$	IEt.mdl	całka z uchybu regulacji położenia po skoku obciążenia	$skok\_obc\_polozenie$ – [Nm] wartość skoku momentu oporowego
Max. zryw po skoku prędkości	$\rho_{\theta max}$	rowmax.mdl	Obliczanie maksymalnej wartości zrywu (pochodnej przyspieszenia) po wystąpieniu skoku wartości zadanej prędkości	$skok\_predkosci\_zryw$ – [rad/s] wartość skoku prędkości
Max. zryw po skoku położenia	$\rho_{\omega max}$	rotmax.mdl	Obliczanie maksymalnej wartości zrywu (pochodnej przyspieszenia) po wystąpieniu skoku wartości zadanej położenia	$skok\_polozenia\_zryw$ – [rad] wartość skoku położenia
Spadek 3dB w pętli przyspieszenia	$\Delta G_{\varepsilon}$	fe3dB.mdl	sprawdzanie spadku wzmocnienia pętli przyspieszenia dla sinusoidalnej wartości zadanej	$fe3dB0$ – [Hz] częstotliwość wymuszenia
Spadek 3dB w pętli prędkości	$\Delta G_{\omega}$	fw3dB.mdl	sprawdzanie spadku wzmocnienia pętli prędkości dla sinusoidalnej wartości zadanej	$fw3dB0$ – [Hz] częstotliwość wymuszenia
Spadek 3dB w pętli położenia	$\Delta G_{\theta}$	ft3dB.mdl	sprawdzanie spadku wzmocnienia pętli położenia dla sinusoidalnej wartości zadanej	$ft3dB0$ – [Hz] częstotliwość wymuszenia
Badanie stabilności pętli przysp.	$GM$	GM.mdl	wyliczanie rzeczywistej wartości marginesu wzmocnienia pętli regulacji przyspieszenia	-

## 6.6. Przeprowadzenie przykładowych symulacji.

Z palety przedstawionych powyżej symulacji przeprowadzone zostaną następujące:

- pomiar czasu ustalania położenia z dokładnością 5% i 0.1%;
- pomiar wartości spadku wzmocnienia w pętli regulacji prędkości;
- badanie stabilności pętli regulacji przyspieszenia oraz pomiar rzeczywistego marginesu wzmocnienia tej pętli (z uwzględnieniem nadrzędnych pętli).

Po wykonaniu pliku *serwo\_1\_obliczenia.m* uzyskujemy następujący zestaw parametrów modelu symulacyjnego (tabela 6):

**Tabela 8 - zestawienie obliczonych parametrów regulatora**

Parametr	Symbol	Nazwa	Wartość
Całkowite opóźnienie w pętli regulacji	$\tau$	tau	0,0009 [s]
Opóźnienie wprowadzane przez filtr	$\tau_F$	tauf	0.0008 [s]
Rozdzielczość obliczania prędkości	$\Omega_{res}$	Omegares	0.00767 [rad/s]



Zmienność parametrów obiektu	$PVR$	PVR	7.7333
Wzmocnienie regulatora przyspieszenia	$K_\varepsilon$	Keps	51.3922 [A/rad/s <sup>2</sup> ]
Maksymalna pulsacja odcięcia w pętli regulacji przyspieszenia	$V_{\varepsilon max}$	vepsmax	555.5543 [rad/s]
Minimalna pulsacja odcięcia w pętli regulacji przyspieszenia	$V_{\varepsilon min}$	vepsmin	71.8389 [rad/s]
Wzmocnienie regulatora prędkości	$K_\omega$	Komega	51.3922 [1/s]
Tłumienie pętli regulacji prędkości	$\xi_\omega$	ksiomega	0.59116
Rząd filtru różniczkującego	$N_f$	Nf	16
Wzmocnienie regulatora położenia	$K_\theta$	Ktheta	15.346 [1/s]
Tłumienie pętli regulacji położenia	$\xi_\theta$	ksitheta	0.915
Obniżenie pierwiastkowej części charakterystyki regulatora położenia <sup>1</sup>	$\Delta\Omega_{NL}$	Domeganl	0.56737 [rad/s]
Minimalna prędkość na nieliniowej części charakterystyki regulatora położenia	$\Omega_{NL}$	Oomega	0.56737 [rad/s]
Ograniczenie prędkości zadanej w regulatorze położenia	$\Omega_{max}$	Omegamax	0.5 [rad/s]
Ograniczenie narastania prędkości w regulatorze położenia <sup>2</sup>	$E_{max}$	Emax	17.4138 [rad/s <sup>2</sup> ]

Parametry te zostaną wykorzystane do przeprowadzenia poniższych symulacji.

#### 6.6.1. Pomiar czasu ustalania położenia z dokładnością 5 % i 0.1%;

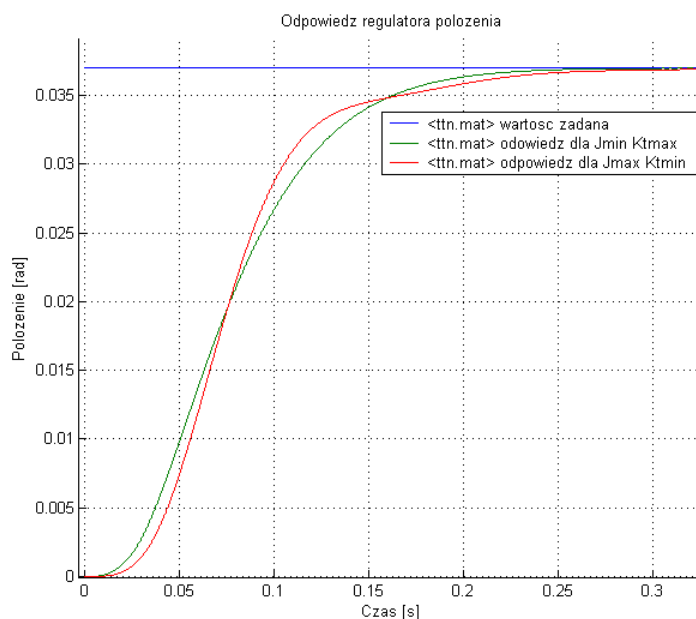
Symulacja dotyczy zakresu liniowego pracy. Jedynym sygnałem ograniczanym przez regulator w tej symulacji jest przyspieszenie kształtowane przez regulator położenia. Ograniczenie tego przyspieszenia zostało ustalone na poziomie  $E_{max}$ :

$$E_{max} \leq \frac{K_{T \min} \cdot I_{q \max} - T_{L \max}}{J_{\max}}$$

Jak widać ograniczenie to dobrano tak, aby dla przypadku największej bezwładności i najmniejszej stałej silnika, zadane przyspieszenie nie było większe, niż to wynika z ograniczenia prądu silnika. Na rysunku 22 przedstawione są przebiegi czasowe położenia dla skrajnych przypadków stosunku stałej momentu i momentu bezwładności:

<sup>1</sup> zob. [2] rozdział 2.4. Nieliniowy regulator położenia kąowego

<sup>2</sup> ograniczenie wynikające z istnienia ogranicznika pochodnej wartości zadanej prędkości



Rysunek 22 - odpowiedź skokowa układu regulacji położenia

Wartość skoku położenia została przyjęta tak, aby osiągnąć maksymalną wartość na charakterystyce liniowej regulatora położenia:

$$\theta_{zad} = 1 \cdot \frac{\Omega_{NL}}{K_{\theta}}$$

Jak widać przebiegi odpowiedzi różnią się nieznacznie. W tabeli 7 przedstawiono porównanie otrzymanego czasu regulacji dla dwóch przypadków dokładności i skrajnych wariantów parametrów napędu.

Tabela 9 - szybkość regulacji położenia

Przypadek	$t_{\theta 5\%}$ [s]		$t_{\theta 0.1\%}$ [s]	
	Założona wartość	Zmierzona wartość	Założona wartość	Zmierzona wartość
$J_{min}, K_{Tmax}$	$< 3 / K_{\theta}$	0.1649	-	0.2708
$J_{max}, K_{Tmin}$	$3 / K_{\theta} = 0.19549$	0.1715		0.355

### 6.6.2. Sprawdzenie spadku wzmocnienia w pętli prędkości.

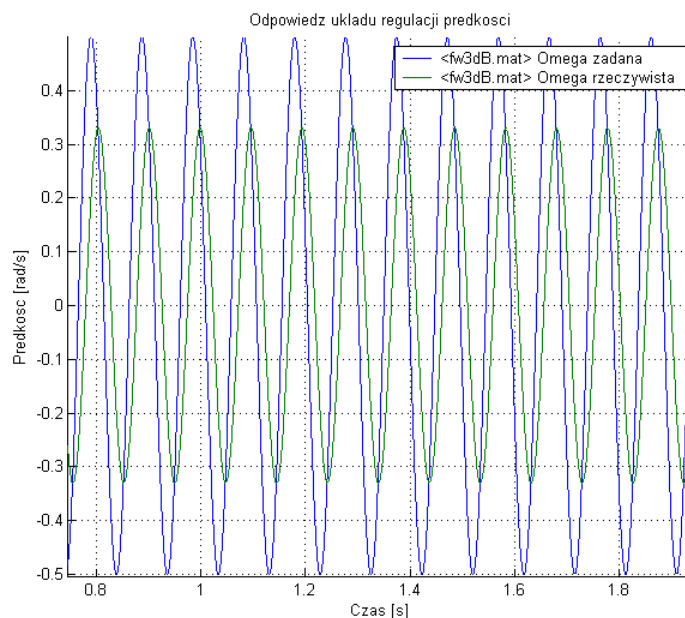
W tej symulacji zostanie przeprowadzone badanie wartości spadku wzmocnienia w pętli prędkości. Regulator położenia jest odłączony od układu, a wartością zadaną regulatora prędkości jest sinusoida. Jej amplitudę dobrano tak, aby układ działał w zakresie liniowym, a jej częstotliwość jest prognozowaną częstotliwością spadku 3dB pętli:

$$f_{\omega 3dB} \approx K_{\omega} / 5 = 10.2784 \text{ [Hz]}$$

Z kolei amplituda sinusoidy została ustalona na poziomie ograniczenia prędkości (w regulatorze położenia). Wymuszenie ma więc postać:

$$\omega_{zad} = 1 \cdot \Omega_{max} \cdot \sin(2\pi K_{\omega}/5 \cdot t) = 1 \cdot 0.5 \cdot \sin(64.58 \cdot t) [rad/s]$$

Rysunek 23 przedstawia wykres przebiegu wartości zadanej i prędkości na wale.

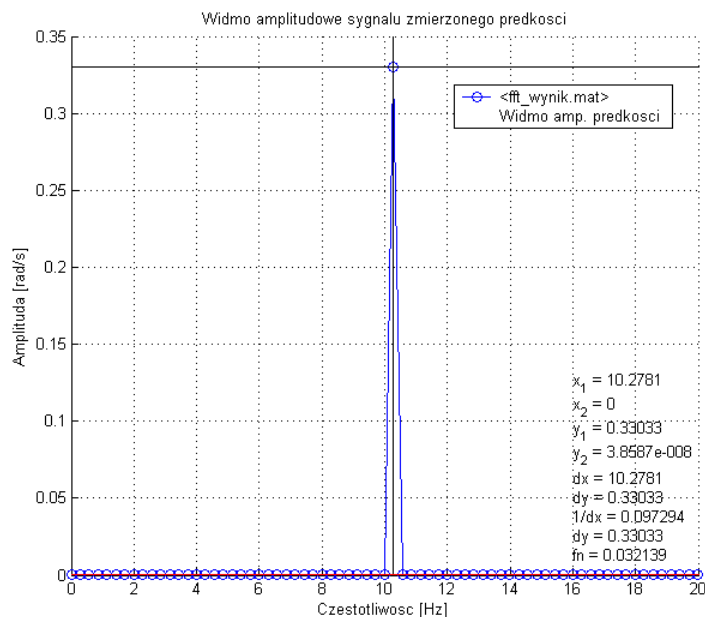


**Rysunek 23 - wartość zadana i zmierzona prędkości**

Na wykresie widać tylko kilka okresów przebiegu. Cała symulacja obejmuje okres 50 przebiegów (ma to na celu zwiększenie dokładności). Wyznaczenie spadku polega na obliczeniu wyrażenia

$$\Delta G_{\omega} = 20 \cdot \log_{10} \left( \frac{A_1}{A_{zad}} \right)$$

gdzie  $\Delta G_{\omega}$  – spadek wzmocnienia [dB],  $A_{zad}$  – amplituda sygnału zadanego [rad/s],  $A_1$  – amplituda pierwszej harmonicznej prędkości na wale [rad/s]. Wartość  $A_1$  jest obliczana za pomocą dyskretnej transformaty Fouriera [15] (rysunek 24):



**Rysunek 24 - widmo sygnału zmierzonego prędkości dla przypadku  $J_{min} K_{Tmax}$**

Dane brane do obliczeń są wstępnie obrobione. Polega to na wykryciu przejścia przez zero podczas narastania sygnału na początku symulacji, oraz podczas opadania sygnału pod koniec symulacji. Dzięki temu do analizy FFT użyta jest dokładna krotność okresu sygnału zadanego. Poza tym, do analizy nie są brane przebiegi przejściowe występujące na początku symulacji. Wyniki przedstawione są w poniższej tabeli:

**Tabela 10 - spadek wzmocnienia w pętli regulacji prędkości**

Przypadek	$\Delta G_{\omega}$ [dB] przy $f = 10.2784$ [Hz]	
	Założona wartość	Zmierzona wartość
$J_{min}, K_{Tmax}$	-3	-3.6005
$J_{max}, K_{Tmin}$		-4.0991

### 6.6.3. Badanie stabilności pętli regulacji przyspieszenia.

W tej symulacji aktywna jest tylko ta część regulatora, która jest odpowiedzialna za regulację przyspieszenia. Jak wspomniano w rozdziale 6.4. Obliczenia regulatora, dobór wzmocnienia w regulatorze przyspieszenia wynika między innymi z warunku zachowania założonego marginesu wzmocnienia. Margines stabilności jest najmniejszy dla przypadku, kiedy stała momentu silnika ma największą wartość a bezwładność – najmniejszą. Dla tego też przypadku będzie przeprowadzona symulacja.

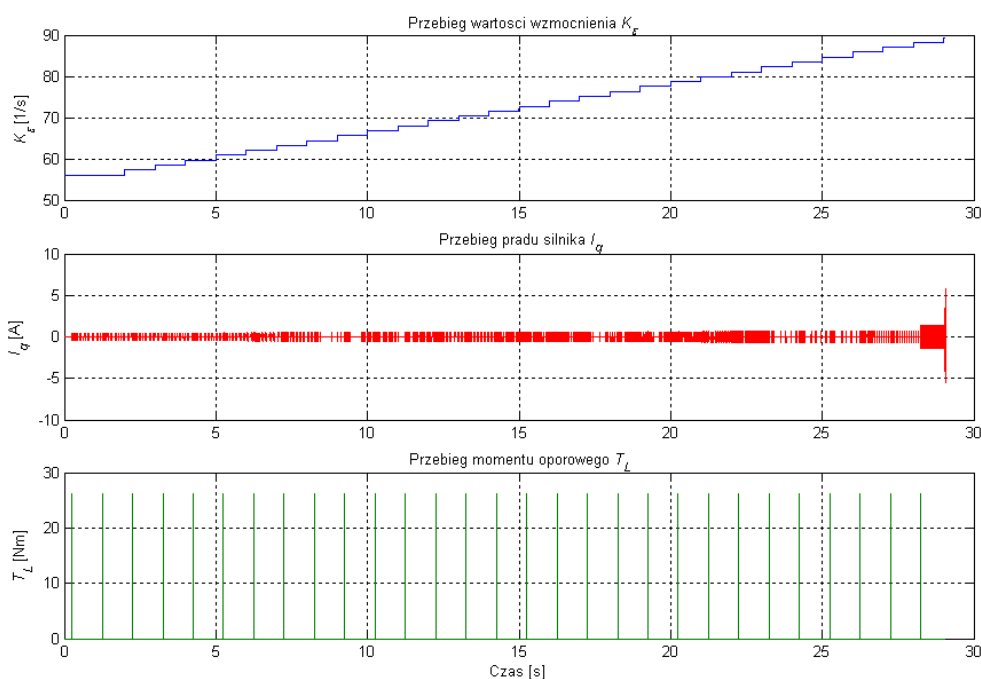
Symulacja polega na stopniowym zwiększaniu wzmocnienia w regulatorze przyspieszenia, aż do momentu utraty stabilności w układzie. Wartość początkowa wzmocnienia wynosi:

$$K_{\varepsilon}(0) = 0.75 \cdot GM_{\varepsilon min} K_{\varepsilon 0}$$

gdzie  $K_{\varepsilon}(0)$  – początkowa wartość wzmocnienia,  $GM_{\varepsilon min}$  – założony minimalny margines wzmocnienia,  $K_{\varepsilon 0}$  – wartość obliczona wzmocnienia regulatora przyspieszenia.

Sposobem na stwierdzenie utraty stabilności jest wykrycie wejścia wartości zadanej prądu silnika w ograniczenie. Zdarzenie to powoduje przerwanie symulacji. Między kolejnymi zmianami wzmacnienia występuje przerwa na ustalenie się odpowiedzi układu. Po upływie 1/4 sekundy od zmiany wzmacnienia, układ zostaje wytrącony z równowagi. Zadanie to jest realizowane przez krótki impuls momentu oporowego. Czas jego trwania przyjęto umownie na jeden okres próbkowania. Z kolei jego wartość jest taka, że w silniku na biegu jałowym wywołała by przyspieszenie dwukrotnie większe, niż ograniczenie przyspieszenia w regulatorze położenia ( $E_{max}$ ). Warto nadmienić, że w idealnym układzie liniowym, wartość momentu oporowego wywołująca niestabilność (przy odpowiednio dużym wzmacnieniu) byłaby bardzo mała. W rzeczywistości do wytrącenia układu z równowagi jest potrzebna dużo większa wartość momentu. Jest to jednak dość złożone zjawisko spowodowane nieliniowością wprowadzaną przez enkoder i trudno je analizować. Na koniec należy wspomnieć, że wartość zadana przyspieszenia jak i warunki początkowe symulacji są zerowe.

Na rysunku 25 przedstawiono na wspólnym wykresie przebieg zmiany wzmacnienia, prąd silnika oraz impulsy momentu oporowego.



**Rysunek 25 - przebieg badania marginesu wzmacnienia**

Symulacja przyniosła następujące wyniki

**Tabela 11 - badanie marginesu wzmacnienia**

$GM$ dla przypadku $J_{min} K_{Tmax}$	
Założona wartość	Zmierzona wartość
$GM_{emin} = \pi$	3.7562

## 6.7. Zebranie wyników i podsumowanie.

Przedstawione w poprzednim podrozdziale symulacje w sposób skrótowy demonstrują możliwości omawianej struktury regulatora. Oprócz tych wyników, w tabeli 10 zostaną zebrane inne wskaźniki jakości uzyskane z symulacji. Część z nich zostanie porównana z wartościami założonymi na etapie projektowania. Wyniki te są efektem przeprowadzenia wszystkich czternastu dostępnych symulacji.

Tabela 12 - zestawienie wyników wszystkich symulacji

Wskaźnik	Sym-bol	Wartość przewidywana/założona	Wartość uzyskana z symulacji	Uwagi
Czas ustalania przyspieszenia z dokładnością 5% [s]	$t_{\varepsilon 5\%}$	-	0.0022	symulacja w liniowym zakresie regulacji
Czas ustalania prędkości z dokładnością 5% [s]	$t_{\omega 5\%}$	$t_{\omega 5\%} < 3/K_{\omega} = 0.0584$	0.0541	symulacja w liniowym zakresie regulacji
Czas ustalania prędkości z dokładnością 5% [s]	$t_{\theta 5\%}$	$t_{\theta 5\%} < 3/K_{\theta} = 0.1955$	0.1649	symulacja w liniowym zakresie regulacji <sup>1</sup>
Czas narastania przyspieszenia do 90% wartości zadanej [s]	$t_{\varepsilon 09}$	$t_{\varepsilon 09} \approx 1.2\tau \cdot GM_{\varepsilon} - \tau_F = 0.0026$	0.0017	symulacja w liniowym zakresie regulacji
Czas narastania prędkości do 90% wartości zadanej [s]	$t_{\omega 09}$	$t_{\omega 09} \approx 2/K_{\omega} = 0.0389$	0.0416	symulacja w liniowym zakresie regulacji
Czas narastania położenia do 90% wartości zadanej [s]	$t_{\theta 09}$	$t_{\theta 09} \approx 2/K_{\theta} = 0.1303$	0.1405	symulacja w liniowym zakresie regulacji
Całka uchybu prędkości po skoku obciążenia [rad]	$IE_{\omega}$	$IE_{\omega} = \frac{T_{Lmax}}{K_{\omega} \cdot K_{\varepsilon} \cdot K_T} = 0.000186$	0.000186	skok obciążenia równy maksymalnemu momentowi obciążenia $T_{Lmax}$
Całka uchybu położenia po skoku obciążenia [rad·s]	$IE_{\theta}$	$IE_{\theta} = \frac{T_{Lmax}}{K_{\theta} \cdot K_{\omega} \cdot K_{\varepsilon} \cdot K_T} = 1.2173e-005$	1.217e-005	skok obciążenia równy maksymalnemu momentowi obciążenia $T_{Lmax}$
Zryw po skoku położenia [rad/s] <sup>3/2</sup>	$\rho_{\theta max}$	$\rho_{\omega max} \approx E_{max} \cdot K_{\omega} = 894.9$	85441.0	skok położenia nie powoduje wyjścia z liniowego zakresu

<sup>1</sup> por. rozdział 6.6.1.

<sup>2</sup> zob. komentarz pod tabelą

Zryw po skoku prędkości [rad/s] <sup>3/2</sup>	$\rho_{wmax}$	$\rho_{\omega max} \approx E_{max} \cdot K_{\omega} = 894.9$	50450.2	skok położenia nie powoduje wyjścia z liniowego zakresu
Spadek wzmocnienia w pętli przyspieszenia [dB]	$\Delta G_{\epsilon}$	- 3	-2.2171	częstotliwość wymuszenia: $f_{\epsilon 3dB} \approx \frac{0.25 \dots 0.5}{\tau \cdot GM_{\epsilon}} =$ 88.4192 ... 176.8384 [Hz], do symulacji przyjmujemy wartość maksymalną
Spadek wzmocnienia w pętli prędkości [dB]	$\Delta G_{\omega}$	- 3	-3.6005	częstotliwość wymuszenia: $f_{\omega 3dB} \approx K_{\omega} / 5 = 10.2784$ [Hz]
Spadek wzmocnienia w pętli położenia [dB]	$\Delta G_{\theta}$	- 3	-2.9318	częstotliwość wymuszenia: $f_{\theta 3dB} \approx K_{\theta} / 5 = 3.0692$ [Hz]
Margines wzmocnienia pętli regulacji przyspieszenia	$GM$	$\pi$	3.7562	początkowa wartość wzmocnienia: $0.57 \cdot GM_{emin} \cdot Ke^2$

Z tabeli wynika, że zaproponowana w [2] struktura i sposób wyznaczania parametrów regulatora dają obiecujące wyniki. Większość ze wskaźników uzyskanych na drodze symulacji ma lepsze wartości, niż przewidywane na etapie analizy modelu liniowego. Należy jednak pamiętać, że model nie uwzględnia wielu czynników mających wpływ na jakość modelowania serwonapędu.

Będą to przede wszystkim:

- rzeczywista charakterystyka (zarówno statyczna jak i dynamiczna) układu regulacji momentu silnika;
- sprężystość wału;
- tarcie.

Czynniki te powodują, że część z otrzymanych wyników jest mało wiarygodna. Ma to miejsce przede wszystkim w eksperymencie dotyczącym pomiaru zrywu po skoku wartości zadanej prędkości lub położenia. W tym eksperymencie uzyskana wartość zrywu jest bardzo zawyżona z powodu nie uwzględnienia właściwości filtrujących układu regulacji prądu. Oczywiście, nawet przy zastosowaniu dokładnego modelu układu regulacji prądu, wartość ta będzie przewyższać założony poziom zrywu (z uwagi na obecność enkodera będącego źródłem zakłóceń impulsowych w pętli regulacji przyspieszenia). Z kolei tarcie może wywoływać problemy ze stabilnością pętli regulacji położenia przy dochodzeniu do wartości zadanej.

Kolejną sprawą, na którą należy zwrócić uwagę, to fakt, że model silnika jest dyskretny.

Zrezygnowano modelu z ciągłego z powodu złożoności obliczeniowej takiego hybrydowego modelu, co pociąga za sobą długi czas symulacji. Poza tym, dla małych okresów próbkowania różnice pomiędzy modelem dyskretnym a ciągłym są niewielkie.

<sup>2</sup> por. rozdział 6.6.3.

Jednym z ciekawych zjawisk zauważonych w czasie symulacji systemu jest wpływ nieliniowości wprowadzanej przez enkoder na stabilność działania pętli regulacji przyspieszenia. Dla małych zmian położenia kąтового wału (rzędu rozdzielczości enkodera) element ten działa jak strefa martwa. Powoduje to tłumienie wpływu impulsowych obciążeń na pracę regulatora przyspieszenia. Szczególnie wyraźnie można to zauważyć podczas badania zapasu wzmocnienia (występuje tutaj lokalna stabilność nawet dla wzmocnień znacznie przekraczających wartość wynikającą z zapasu wzmocnienia).



## 7. Podsumowanie.

Stworzone na potrzeby pracy oprogramowanie stanowi skromną namiastkę profesjonalnych systemów CAD. Celem pracy nie było jednak stworzenie rozbudowanego narzędzia, lecz zbudowanie w miarę prostego i przejrzystego interfejsu wspomagającego projektowanie i symulację regulatorów dla serwonapędów. Słowo „wspomaganie” jest tu jak najbardziej na miejscu. To od projektanta zależy jakie algorytmy obliczeniowe i modele zostaną zastosowane. Oprogramowanie to stanowi jedynie środowisko, w którym zostają zintegrowane w miarę możliwości wszystkie działania związane z projektowaniem i analizą układu. Mam tu na myśli następujące etapy: wprowadzanie danych, obliczenie parametrów układu, eksperymenty symulacyjne, zebranie wyników i ich graficzną reprezentację. Taki tok postępowania wydaje się logiczny i ma w mniemaniu autora zastosowanie nie tylko do projektowania regulatorów serwonapędu, lecz wszędzie tam, gdzie występują wyżej wymienione etapy. Oprogramowanie stworzono w środowisku Matlab w wersji R13, a modele symulacyjne w Simulink 5. Dostępne w tych programach narzędzia w sposób nieoceniony ułatwiły autorowi programowanie. Należy wymienić kilka z nich, które były najintensywniej używane i wpłynęły na użyteczność programu. Są to:

- modelowanie i analiza układów liniowych (badanie odpowiedzi skokowych modeli liniowych);
- całościowa symulacja systemu z uwzględnieniem nieliniowości – modele Simulinka;
- obsługa plików tekstowych (w tym formatowany odczyt i wyszukiwanie z wykorzystaniem wyrażeń regularnych): konfiguracja okien, zapis/odczyt parametrów wejściowych;
- obsługa arkusza Excela – generowanie zestawień z wynikami symulacji;
- obsługa zapisu i odczytu danych wynikowych symulacji – umożliwienie eksportu przebiegów z diagramów symulacyjnych do programu Oscyloskop;
- proste programowanie interfejsu graficznego z pomocą narzędzia GUIDE.

Oprócz stworzenia narzędzia, ważną częścią pracy było jego praktyczne zastosowanie do realizacji konkretnego zadania. Zadaniem tym było przygotowanie odpowiedniego zestawu funkcji obliczeniowych i symulacji służących do projektowania i analizy odpornego na zmiany parametrów układu regulacji prędkości i położenia dla serwonapędu. Rodzaj struktury i sposób obliczania nastaw regulatora zaczerpnięto z [1] i [2]. Struktura ta opiera się na trój-pętlowym układzie regulacji podporządkowanej z nadrzędnym nieliniowym regulatorem położenia kąowego, liniowym, proporcjonalnym regulatorem prędkości i całującym regulatorem przyspieszenia. Z kolei algorytm obliczania nastaw regulatora składa się w głównej mierze z 3 etapów:

- obliczenie parametrów filtru różniczkującego i wzmocnienia regulatora przyspieszenia ze względu na maksymalny poziom tętnień prądu (a więc pośrednio i momentu);
- ustalenie wzmocnienia regulatora prędkości ze względu na założony poziom przeregulowania w jego odpowiedzi skokowej

- ustalenie wzmocnienia regulatora i parametrów jego nieliniowej części ze względu na przeregulowanie w jego odpowiedzi skokowej.

Otrzymane wyniki są zadowalające i sugerują poprawność modelowania systemu i metodykę projektowania regulatora.

Autor wyraża nadzieję, że zaproponowane narzędzia są w miarę możliwości pozbawione błędów, proste w obsłudze i znajdą praktyczne zastosowanie podczas prac badawczych.

## Spis ilustracji i tabel

Rysunek 1 - symulacja regulatora dla silnika prądu stałego stworzona w Simulinku .....	8
Rysunek 2 - symulacja regulatora dla silnika prądu stałego stworzona w Vissimie .....	9
Rysunek 3 - okno główne narzędzia SISO Design Tool .....	10
Rysunek 4 - odpowiedź skokowa zaprojektowanego układu .....	10
Rysunek 5 - charakterystyka amplitudowo - fazowa układu zamkniętego .....	11
Rysunek 6 - model prostego układu regulacji automatycznej .....	12
Rysunek 7 - okno optymalizacji parametrów .....	12
Rysunek 8 - pierwsze okno kreatora - wprowadzanie parametrów .....	15
Rysunek 9 - wyniki obliczenia parametrów regulatora .....	16
Rysunek 10 - okno wyboru symulacji .....	17
Rysunek 11 - ostatnie okno kreatora - wyniki symulacji .....	18
Rysunek 12 - okno programu Oscyloskop .....	19
Rysunek 13- struktura katalogów i plików .....	21
Rysunek 14 - zmienne związane z kreatorem symulacji .....	25
Rysunek 15 - Dane związane z programem Oscyloskop .....	27
Rysunek 16 - konwencjonalna struktura regulatora PID .....	46
Rysunek 17 - zmodyfikowana struktura PID .....	47
Rysunek 18 - zmodyfikowana struktura z blokami nieliniowymi w regulatorze położenia .....	47
Rysunek 19 - schemat procesu projektowania regulatora .....	48
Rysunek 20 - podstawowy diagram symulacyjny .....	49
Rysunek 21 - maszynowy algorytm obliczania nastaw regulatora .....	54
Rysunek 22 - odpowiedź skokowa układu regulacji położenia .....	58
Rysunek 23 - wartość zadana i zmierzona prędkości .....	59
Rysunek 24 - widmo sygnału zmierzonego prędkości dla przypadku $J_{min} K_{Tmax}$ .....	60
Rysunek 25 - przebieg badania marginesu wzmocnienia .....	61
Tabela 1 - pliki wchodzące w skład programów .....	22
Tabela 2 - zmienne kreatora symulacji .....	25
Tabela 3 - zmienne programu Oscyloskop .....	28
Tabela 4 - zestawienie parametrów napędu .....	50
Tabela 5 – projektowanie regulatora przyspieszenia i filtru .....	51
Tabela 6 - projektowanie regulatorów prędkości i położenia .....	52
Tabela 7 - zestawienie dostępnych symulacji .....	55
Tabela 8 - zestawienie obliczonych parametrów regulatora .....	56
Tabela 9 - szybkość regulacji położenia .....	58
Tabela 10 - spadek wzmocnienia w pętli regulacji prędkości .....	60
Tabela 11 - badanie marginesu wzmocnienia .....	61
Tabela 12 - zestawienie wyników wszystkich symulacji .....	62

## Bibliografia

- [1] Deskur J.: Odporny regulator PID serwonapędu, IX Sympozjum Energoelektronika w Nauce i Dydaktyce, Poznań 20 – 22 września 2004
- [2] Deskur J.: Dobór nastaw regulatora PID odpornego na zakłócenia. Materiały niepublikowane autora, 2004
- [3] Deskur J.: Praktyczna metoda projektowania regulatorów prędkości i położenia odpornych na zakłócenia i zmiany parametrów napędu. V Krajowa Konferencja Naukowa Sterowanie w Energoelektronice i Napędzie Elektrycznym SENE, Łódź-Arturówek, 14-16 listopada 2001r ., tom I, str.105-110
- [4] Brzózka J.: Regulatory i układy automatyki, MIKOM, Warszawa 2004
- [5] Bisztyga K.: Sterowanie i regulacja silników elektrycznych, WNT, Warszawa 1989
- [6] Tomera M. Układ sterowania z regulatorem PID, Wyższa Szkoła Morska w Gdyni Katedra Automatyki Okrętowej, podręcznik laboratorium teorii sterowania
- [7] Płyta CD: materiały\Cd2\metoda\Vissim
- [8] Strona domowa Matlab: <http://www.mathworks.com/products/matlab/>
- [9] Strona domowa Simulink: <http://www.mathworks.com/products/simulink/>
- [10] Strona domowa Vissim: <http://www.vissim.com/products/index.html>
- [11] Strona domowa Scilab: <http://scilabsoft.inria.fr/>
- [12] Dokumentacja SISO Design Tool:  
<http://www.mathworks.com/access/helpdesk/help/toolbox/control/quick/guiquic2.html>
- [13] Dokumentacja NCD: pomoc Matlabu 6.5 pod tytułem Nonlinear Control Design Blockset; w nowszych wersjach Matlabu NCD został zastąpiony przez Response Optimization:  
<http://www.mathworks.com/access/helpdesk/help/toolbox/sloptim/>
- [14] Spis plików pomocy dla Control System Toolbox:  
<http://www.mathworks.com/access/helpdesk/help/toolbox/control/>
- [15] Artykuł na temat przetwarzania sygnałów (w tym FFT):  
<http://www.egr.msu.edu/decs/facilities/software/matlab/matlabses7.pdf>
- [16] Stosowanie wyrażeń regularnych: <http://www.regular-expressions.info/tutorial.html>
- [17] Eksportowanie danych z Simulink:  
<http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/slref/toworkspace.html>
- [18] Przesyłanie danych między Matlabem a Excelem:  
[http://www.mathworks.com/access/helpdesk/help/techdoc/matlab\\_external/ch07cl19.html](http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_external/ch07cl19.html)

## Załącznik.

### Plik wprowadzania danych.

```
function Serwo_1_parametry() %ta linia jest opcjonalna, to moze byc tez skrypt
                             %roznica polega na oszczednosci pamieci w przypadku
                             %zastosowania funkcji

global parametry;
#####
%Regulator przyspieszenia i filtr
#####
% dane wejsciowe
parameter.Ts=0.0001; %[Ts] okres probkowania i obliczen
parameter.Ttheta=1.2272e-005; %[rad] blad rozdzielczosci i pomiaru kata
parameter.Iqn=6; %[A] nominalny prad czynny
parameter.Tlmax=4; %[Nm] najwiekszy moment oporowy
parameter.taug=0.0001; %[s] opoznienie w petli regulacji przysp. (bez filtru)
#####
%Parametry projektowe
#####
parameter.GMepsmin=3.1416; %[] margines wzmoocnienia dla parametry. regulacji przysparametry.
parameter.Igmax=6; %[A] ~Iqn, poziom ograniczenia syg. wyjsciowego
parameter.Triple=0.2; %[A] (0.01*Iqn) dopuszczalny poziom tetnien w sygnale pradu
parameter.Iq % zadanego i*q
#####
%Regulator predkosci i polozenia
#####
% dane wejsciowe
parameter.Omega=0.5; %[rad/s] predkosci znamionowa
parameter.Ktmax=17.5; %[Nm/A] najwieksza wartosc stalej momentu
parameter.Ktmin=17.5; %[Nm/A] najmniejsza wartosc stalej momentu
parameter.Jmin=0.75; %[kgm^2] najmniejszy moment bezwladnosci
parameter.Jmax=5.8; %[kgm^2] najwiekszy moment bezwladnosci

parameter.AOmega=0.05; %[rad/s] (0.1*Omega) przeregulowanie bezwzgledne w reg. predkosci
parameter.ROmega=0.1; %[] przeregulowanie wzglednie w reg. predkosci

parameter.ROtheta=0; %[] przeregulowanie wzglednie w reg. polozenia
%nie podajemy przeregulowan, tylko wspolczynnik tlumienia
parameter.ksitheta=0.85; %[] poczatkowa wartosc wspolczynnika tlumienia
#####
%Parametry symulacyjne: te zmienne tez mog' zostac zapisane za pomoca Plik-Zapisz parametry
#####
parameter.skok_obc_polozenie=4; %[Nm] skok obciazenia w symulacji IEt.mdl (calka z
% uchybu w petli polozenia)
parameter.skok_obc_predkosc=4; %[Nm] skok obciazenia w symulacji IEw.mdl (calka z
% uchybu w petli predkosci)
parameter.prec_przyspieszenia=5; %[%] dokladnosc w ekperymentcie czasu regulacji
% petli przyspieszenia
parameter.prec_predkosci=5; %[%] dokladnosc w ekperymentcie czasu regulacji
% petli predkosci
parameter.prec_polozenia=1; %[%] dokladnosc w ekperymentcie czasu regulacji
% petli polozenia
#####
%Domyslny wariant symulacji
#####
parameter.wariant_symulacji=1; %[] 1 dla Jmin, Ktmax
% 2 dla Jmax, Ktmin
% 3 dla Jmin, Ktmin
% 4 dla Jmax, Ktmax
```

### Plik obliczeń nastaw regulatora.

```
function Serwo_1_obliczenia() %ta linia jest opcjonalna, to moze byc tez skrypt
                              %roznica polega na oszczednosci pamieci w przypadku
                              %zastosowania funkcji

global parametry
global p
global pozycja_okna
%March 17, 2005 07:37:37 PM
%program zwraca w postaci struktury 'parametry' wartosci nastaw zwiazanych z regulatorem
%jako parametr wejsciowy przesylna jest ta sama struktura, zawierajaca na poczatku parametry
%stale okreslone z gory w plikach o nazwach Parametry_stale_xx
login('Rozpoczeto obliczanie nastaw regulatora');
p=parametry; %to jest tylko pomocnicze przepisane
#####
%Przepisanie wartosci poczatkowych
#####
p.ksitheta=p.ksitheta0;
p.ksiomega=p.ksiomega0;
p.GMepsmin=p.GMepsmin0;
#####
%Przeniesione z Parametry_stale
#####
p.PVR=(p.Ktmax*p.Jmax)/(p.Ktmin*p.Jmin); %[] stosunek zmienności parametrow (Parameter
% Variation Ratio)
#####
%Pierwszy krok sprawdzenie Keps ze wzgledu na Iriple
#####
powtorz=1; %flaga poprawiania w reg. przysp
zmien_Nf=0; %flaga automatycznej zmiany rzędu filtru
while(powtorz),
    #####
    %Wstepne wartosci zwiazane z filtrem
    #####
endwhile
```

```

p.SRtau=0.5; %[] iloczyn opoznienia i maksimum odpowiedzi
%[] skokowej
p.tauf=sqrt(p.SRtau*(pi/2)*p.Thetares*(p.Jmin/p.Ktmax)*(1/p.Iripple)*(1/p.GMepsmin)+(p.taug/2)^2)-(p.taug/2); %[] oszacowanie opoznienia filtru
%#####
%Przyklad obliczenia prostego filtru rozniczujacego
%#####
if ~zmien_Nf,
    p.Nf=ceil(2*p.tauf/p.Ts); %[] poczatkowy rzad filtru
else
    p.Nf=p.Nf+1; %[] automatyczne powiekszenie rzędu
end
bi=1/(p.Nf*p.Ts); %[] wartosci pierwszego i ostatniego bi
p.B=zeros(1,p.Nf+1); %[] przygotowanie wektora wsp. bi
p.B(1)=bi; %[] przypisanie pierwszego i
p.B(p.Nf+1)=-bi; %[] ostatniego wyrazu wektora B
p.A=zeros(1,p.Nf+1); %[] wspolczynniki mianownika
p.A(1)=1; %[] tylko jeden element niezerowy

p.tauf=p.Nf*p.Ts/2; %[] nowo obliczone opoznienie filtru
p.SRmax=1/(p.Nf*p.Ts); %[] maksimum odpowiedzi skokowej filtru
p.Omegares=p.Thetares*p.SRmax; %[] [rad/s] blad rozdzielczosci wyliczanej predkosci
%#####
%Obliczenie Keps
%#####
p.tau=p.taug+p.tauf; %[] s] calkowite opoznienie w petli przysp.
p.vepsmax=(2*pi)/(4*p.tau*p.GMepsmin); %[] [rad/s] najwyzsza czestotliwosc odcięcia
p.Keps=(p.vepsmax*p.Jmin)/(p.Ktmax); %[] wzmacnienie regulatora
%#####
%Sprawdzenie warunku poprawnosci obliczen
if (p.Omegares*p.Keps<p.Iripple),
    powtorz=0;
else
    zmien_Nf=1;
end
end
%Na koniec stworzenie modelu
epsOL1=tf([p.Keps],[1 0])*p.Ktmax/p.Jmin;%otwarta petla regulacji przyspieszenia I
epsOL2=tf([p.Keps],[1 0])*p.Ktmin/p.Jmax;%otwarta petla regulacji przyspieszenia II
[num,den]=pade(p.tau,3); %przyblizenie calkowitego opoznienia czlonem 3 rzędu
opoznienie=tf(num,den); %stworzenie toru opoznienia
epsCL1=feedback(epsOL1,opoznienie); %zamknieta petla regulacji przyspieszenia I
epsCL2=feedback(epsOL2,opoznienie); %zamknieta petla regulacji przyspieszenia II
%#####
%Wybor parametrow projektowych ksiomega i ksitheta
%#####
%Dane wejsciowe
p.vepsmin=p.vepsmax/p.PVR; %[] [rad/s] najnizsza czestotliwosc odcięcia petli reg.
%[] przyspieczenia
p.Omegamax=p.Omegam; %[] [rad/s] ~Omegan, ograniczenie predkosci zadanej w
regulatorze polozenia katowego
p.Emax=(p.Ktmin*p.Iqmax-p.Tlmax)/p.Jmax; %[] [rad/s^2] =<(Ktmin*Iqmax-Tlmax)/Jmax - ograniczenie
%[] pochodnej predkosci zadanej
p.Enl=p.Emax; %[] [rad/s^2] przyspieszenie ksztaltowane przez regulator
%[] polozenia katowego
%#####
%rozwiazanie rownania na wspolczynnik tlumienia petli regulacji predkosci
syms ksiomega1 ksiomega2 z; %zmiennne symboliczne
s='z=2*ksiomega2*exp((-ksiomega2*(pi-acos(ksiomega2)))/(sqrt(1-ksiomega2^2)))';
%[] funkcja symboliczna reprezentujaca zaleznosc wiazaca
%[] ksiomega2 z zalozonymi parametrami
s=subs(s,z,p.AOomega/(p.Emax/p.vepsmin));%podstawienie pomocnicze
p.ksiomega2=subs(solve(s,ksiomega2));%rozwiazanie symboliczne ze wzgledu na ksiomega2
p.ksiomega1=-log(p.ROomega)/sqrt(pi^2+(log(p.ROomega))^2);
%[] wyliczenie (numeryczne) ksiomega2
p.ksiomega=max([p.ksiomega1 p.ksiomega2]);%wybor wiekszej wartosci tlumienia
%#####
%#####
%Regulatory predkosci i polozenia
%#####
%Obliczenia wzmoocnien regulatorow
%#####
%Obliczenie p.Komega
%#####
[num,den]=pade(p.tau,4);
opoznienie=tf(num,den); %przyblizenie calkowitego opoznienia

powtorz=1; %flaga poprawiania regulatora predkosci

while(powtorz),
    p.Komega=p.vepsmin/(4*(p.ksiomega)^2); %[]
    omegaOL1=epsCL1*p.Komega*tf(1,[1 0]); %otwarta petla regulacji predkosci I
    omegaOL2=epsCL2*p.Komega*tf(1,[1 0]); %otwarta petla regulacji predkosci II

    omegaCL1=feedback(omegaOL1,1); %zamkniecie petli I
    omegaCL2=feedback(omegaOL2,1); %zamkniecie petli II

    [y1,t]=step(omegaCL1); %badanie skoku jednostkowego w regulatorze predkosci I
    [y2,t]=step(omegaCL2); %badanie skoku jednostkowego w regulatorze predkosci II
    ROomega=max([y1;y2])-1; %wyliczenie przer. wzglednego dla gorszego przypadku
    if (ROomega<p.ROomega),
        powtorz=0;
    else
        p.ksiomega=p.ksiomega+0.001; %jesli zbyt duze przeregulowanie, to zwieksz tlumienie
    end
end

end

%#####
%Obliczenie p.Ktheta

```

```

#####
powtorz=1;

hfig=figure;
set(hfig,'Color',[1 1 1],'Name','Odpowiedzi skokowe dla reg. polozenia (zakres liniowy)');
set(hfig,'Position',pozycja_okna,'NumberTitle','off');
grid on;
hold on;
hstep1=line(0,0);
hstep2=line(0,0);
set(hstep1,'Color','r');
set(hstep2,'Color','k');
legend([hstep1 hstep2],'dla \itKt_{max}\rm i \itJ_{min}','dla \itKt_{min}\rm i \itJ_{max}');
komunikat=text(0,0,'');

set([hstep1 hstep2],'EraseMode','normal','MarkerSize',18)
set([hstep1 hstep2],'LineWidth',1);
set(gcf,'renderer','Zbuffer');

while(powtorz),
p.Ktheta=p.Komega/(4*(p.ksitheta)^2);    %[]          wzmacnienie reg. polozenia

pos=[0.01*get(gca,'Xlim') 0.9*get(gca,'Ylim')];% wyswietlenie wartosci wzmacnienia i tlumienia
set(komunikat,'String',{'Wzmacnienie regulatora \it{K}_{\theta}\rm : ' num2str(p.Ktheta)};...
['Tlumienie petli polozenia \it{\xi}_{\theta}\rm : ' num2str(p.ksitheta)};...
'Position',[pos(2) pos(4)]);

thetaOL1=omegaCL1*p.Ktheta*tf(1,[1 0]);
thetaOL2=omegaCL2*p.Ktheta*tf(1,[1 0]);
thetaCL1=feedback(thetaOL1,1);
thetaCL2=feedback(thetaOL2,1);

[y1,t1]=step(thetaCL1);
[y2,t2]=step(thetaCL2);

set(hstep1,'XData',t1,'YData',y1);
set(hstep2,'XData',t2,'YData',y2);
drawnow;
ROtheta=max([y1;y2])-1;
if(ROtheta<p.ROtheta),
    powtorz=0;
else
    p.ksitheta=p.ksitheta+0.001;
end
end
close(hfig);
%Obliczenia parametrow czlonu nieliniowego
p.DOmeگان1=p.Enl/(2*p.Ktheta);    %[]          obnizenie czesci pierwiastkowej ch-ki
p.Omeگان1=p.DOmeگان1;    %[]          najnizsza predkosc na ch-ce nieliniowej
#####
%Obliczenia parametrow potrzebnych do symulacji
#####
p.skok_przyspieszenia_n=p.Emax;    %[rad/s^2]    wstepne przyjecie skoku przyspieszenia
%          o takiej wartosci jaka wynika z
%          z ograniczenia dynamiki w reg. polozenia
p.skok_predkosci_n=p.Omegamax;    %[rad/s]    wstepne przyjecie skoku predkosci
%          o wartosci maksymalnej predkosci silnika
p.skok_polozenia_n=p.Omeگان1/p.Ktheta;    %[rad/s]    wstepne przyjecie skoku polozenia
%          o maksymalnej wartosci na
%          liniowej czesci ch-ki reg. polozenia
p.skok_przyspieszenia_09=p.Emax;    %[rad/s^2]    wstepne przyjecie skoku przyspieszenia
%          o takiej wartosci jaka wynika z
%          z ograniczenia dynamiki w reg. polozenia
p.skok_predkosci_09=p.Omegamax;    %[rad/s]    wstepne przyjecie skoku predkosci
%          o wartosci maksymalnej predkosci silnika
p.skok_polozenia_09=p.Omeگان1/p.Ktheta;    %[rad/s]    wstepne przyjecie skoku polozenia
%          o maksymalnej wartosci na
%          liniowej czesci ch-ki reg. polozenia
p.skok_polozenia_zryw=p.Omeگان1/p.Ktheta;    %[rad/s]    wstepne przyjecie skoku polozenia
%          o maksymalnej wartosci na
%          liniowej czesci ch-ki reg. polozenia
p.skok_predkosci_zryw=p.Omegamax;    %[rad/s]    wstepne przyjecie skoku predkosci
%          o wartosci maksymalnej predkosci silnika
p.amplituda_fe3dB=p.Emax;    %[rad/s^2]    amplituda wymuszenia w symulacji spadku
%          3dB petli przyspieszenia
p.amplituda_fw3dB=p.Omegamax;    %[rad/s^2]    amplituda wymuszenia w symulacji spadku
%          3dB petli predkosci
p.amplituda_ft3dB=p.Omeگان1/p.Ktheta;    %[rad/s^2]    amplituda wymuszenia w symulacji spadku
%          3dB petli predkosci
#####
%obliczenie zalozonych wartosci wskaznikow
#####
p.ft3dB0=p.Ktheta/5;    %[Hz]    zalozona czestotliwosc -3dB petli polozenia
p.fw3dB0=p.Komega/5;    %[Hz]    zalozona czestotliwosc -3dB petli predkosci
p.fe3dB0=0.375/(p.tau*p.GMepsmin);    %[Hz]    (zamiast 0.375 moze byc 0.25 do 0.5)
%          zalozona czestotliwosc -3dB petli przyspieszenia
p.rotmax00=p.Emax*p.Komega;    %[rad/s^3]    przewidywana wartosc zrywu po skoku
%          polozenia
p.rowmax00=p.Emax*p.Komega;    %[rad/s^3]    przewidywana wartosc zrywu po skoku
%          predkosci
#####
%przepisanie pelnej nazwy struktury
#####
parametry=p;
clear global p;
login('Zakończono obliczanie nastaw regulatora');

```

## Plik obliczeń wyników – wskaźników jakości.

```

function Serwo_1_wyniki()                %ta linia jest opcjonalna, to moze byc tez skrypt
                                        %roznica polega na oszczednosci pamieci w przypadku
                                        %zastosowania funkcji

global parametry
global sciezka_glowna sciezka_modeli

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obliczenie czasu ustalania sie z 5% dokladnoscia przyspieszenia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist([sciezka_modeli '\ten.mat']),
    load([sciezka_modeli '\ten.mat']);
    zadany=ten.signals.values(:,1);
    zadany=zadany(length(zadany));           %zadany jest ostatnia wartoscia z sygnalu zadanego
    rzeczywisty=ten.signals.values(:,2);
    i=1;
    while ((rzeczywisty(i)<(1-0.01*parametry.prec_przyspieszenia)*zadany)),
        i=i+1;
    end
    parametry.ten=ten.time(i);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obliczenie czasu ustalania sie z 5% dokladnoscia predkosci
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist([sciezka_modeli '\twn.mat']),
    load([sciezka_modeli '\twn.mat']);
    zadany=twn.signals.values(:,1);
    zadany=zadany(length(zadany));           %zadany jest ostatnia wartoscia z sygnalu zadanego
    rzeczywisty=twn.signals.values(:,2);
    i=0;
    while ((rzeczywisty(length(rzeczywisty)-i)>(1-0.01*parametry.prec_predkosci)*zadany)...
        &&(rzeczywisty(length(rzeczywisty)-i)<(1+0.01*parametry.prec_predkosci)*zadany))
        i=i+1;
    end
    parametry.twn=twn.time(length(rzeczywisty)-i);
    parametry.tw50=3/parametry.Komega;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obliczenie czasu ustalania sie z 5% dokladnoscia polozenia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist([sciezka_modeli '\ttn.mat']),
    load([sciezka_modeli '\ttn.mat']);
    zadany=ttn.signals.values(:,1);
    zadany=zadany(length(zadany));           %zadany jest ostatnia wartoscia z sygnalu zadanego
    rzeczywisty=ttn.signals.values(:,2);
    i=0;
    while ((rzeczywisty(length(rzeczywisty)-i)>(1-0.01*parametry.prec_polozenia)*zadany)...
        &&(rzeczywisty(length(rzeczywisty)-i)<(1+0.01*parametry.prec_polozenia))),
        i=i+1;
    end
    parametry.ttn=ttn.time(length(rzeczywisty)-i);
    parametry.tt50=3/parametry.Ktheta;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obliczenie czasu narastania do 90% przyspieszenia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist([sciezka_modeli '\te09.mat']),
    load([sciezka_modeli '\te09.mat']);
    zadany=te09.signals.values(:,1);
    zadany=zadany(length(zadany));           %zadany jest ostatnia wartoscia z sygnalu zadanego
    rzeczywisty=te09.signals.values(:,2);
    i=1;
    while ((rzeczywisty(i)<0.9*zadany)),
        i=i+1;
    end
    parametry.te09=te09.time(i);
    parametry.te090=1.2*parametry.tau*parametry.GMepsmin-parametry.tauf;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obliczenie czasu narastania do 90% predkosci
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist([sciezka_modeli '\tw09.mat']),
    load([sciezka_modeli '\tw09.mat']);
    zadany=tw09.signals.values(:,1);
    zadany=zadany(length(zadany));           %zadany jest ostatnia wartoscia z sygnalu zadanego
    rzeczywisty=tw09.signals.values(:,2);
    i=1;
    while (rzeczywisty(i)<0.9*zadany),
        i=i+1;
    end
    parametry.tw09=tw09.time(i);
    parametry.tw090=2/parametry.Komega;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%obliczenie czasu narastania do 90% polozenia
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist([sciezka_modeli '\tt09.mat']),
    load([sciezka_modeli '\tt09.mat']);
    zadany=tt09.signals.values(:,1);
    zadany=zadany(length(zadany));           %zadany jest ostatnia wartoscia z sygnalu zadanego
    rzeczywisty=tt09.signals.values(:,2);
    i=1;
    while (rzeczywisty(i)<0.9*zadany),
        i=i+1;
    end
    parametry.tt09=tt09.time(i);
    parametry.tt090=2/parametry.Ktheta;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

%obliczenie calki uchybu predkosci po skoku obciazenia
#####
if exist([sciezka_modeli '\IEW.mat']),
    load([sciezka_modeli '\IEW.mat']);
    wynik=IEW.signals.values(:,1);
    parametry.IEomega=wynik(length(wynik));

    switch(parametry.wariant_symulacji)
        case 1
            Kt=parametry.Ktmax;
        case 2
            Kt=parametry.Ktmin;
        case 3
            Kt=parametry.Ktmin;
        case 4
            Kt=parametry.Ktmax;
    end
    parametry.IEomega0=parametry.skok_obc_predkosc/(parametry.Keps*parametry.Komega*Kt);
end
%cd(sciezka_glowna)
#####
%obliczenie calki uchybu polozenia po skoku obciazenia
#####
if exist([sciezka_modeli '\IET.mat']),
    load([sciezka_modeli '\IET.mat']);
    wynik=IET.signals.values(:,1);
    parametry.IEtheta=wynik(length(wynik));

    switch(parametry.wariant_symulacji)
        case 1
            Kt=parametry.Ktmax;
        case 2
            Kt=parametry.Ktmin;
        case 3
            Kt=parametry.Ktmin;
        case 4
            Kt=parametry.Ktmax;
    end
    parametry.IEtheta0=parametry.skok_obc_polozenie/(parametry.Keps*parametry.Komega*...
        parametry.Ktheta*Kt);
end
#####
%obliczenie czestotliwosci -3dB petli polozenia
#####
if exist([sciezka_modeli '\ft3dB.mat']),
    load([sciezka_modeli '\ft3dB.mat']);
    parametry.ft3dB=parametry.ft3dB0;
    zadany=ft3dB.signals.values(:,1);

    dana=ft3dB.signals.values(:,2);

    for i=1:length(dana),
        if ((dana(length(dana)-i+1)>=0) && (dana(length(dana)-i)<0)),
            break;
        end
    end

    for j=floor(length(dana)/4):length(dana)-1,
        if ((dana(j)<=0) && (dana(j+1)>0)),
            break;
        end
    end
    temp=dana(j:length(dana)-i);
    dana=temp;
    %figure
    %plot(dana, '.')
    N=length(dana);
    %ilosc probek

    T=parametry.Ts;
    F=fft(dana);
    %okres
    %transormata
    mag=abs([F(1)/N ; F(2:floor(N/2))/(N/2)]);
    %wyliczenie amplitud skladowych
    f=[0:floor((N-1)/2)]*(1/(N*T));
    %wektor czestotliwosci

    %figure;
    %plot(f,mag,'o-')
    a0=max(zadany);
    a1=max(mag);
    %harmoniczna podstawowa

    parametry.gt3dB=20*log10(a1/a0);
end
#####
%obliczenie czestotliwosci -3dB petli predkosci
#####
if exist([sciezka_modeli '\fw3dB.mat']),
    load([sciezka_modeli '\fw3dB.mat']);
    parametry.fw3dB=parametry.fw3dB0;
    zadany=fw3dB.signals.values(:,1);

    dana=fw3dB.signals.values(:,2);

    for i=1:length(dana),
        if ((dana(length(dana)-i+1)>=0) && (dana(length(dana)-i)<0)),
            break;
        end
    end

    for j=floor(length(dana)/4):length(dana)-1,
        if ((dana(j)<=0) && (dana(j+1)>0)),
            break;
        end
    end
end

```

```

temp=dana(j:length(dana)-i);

dana=temp;
%figure
%plot(dana)

N=length(dana); %ilosc probek

T=parametry.Ts; %okres
F=fft(dana); %transformata
mag=abs([F(1)/N ; F(2:floor(N/2))/(N/2)]); %wyliczenie amplitud skladowych
f=[0:floor((N-1)/2)]*(1/(N*T)); %wektor czestotliwosci

if 0,
h=figure;
figure(h);
stem(f,mag);
title('Widmo amplitudowe predkosci');
xlabel('Czestotliwosc [Hz]');
ylabel('Amplituda [rad/s]');
fft_wynik.signals.values=mag;
fft_wynik.time=f;
fft_wynik.signals.label='Widmo amp. predkosci';
fft_wynik.blockName='Wygenerowane z serwo_1_wyniki.m';
save fft_wynik fft_wynik
end
a0=max(zadany);
a1=max(mag); %harmoniczna podstawowa

parametry.gw3dB=20*log10(a1/a0);
end
#####
%obliczenie czestotliwosci -3dB petli przyspieszenia
#####
if exist([sciezka_modeli '\fe3dB.mat']),
load([sciezka_modeli '\fe3dB.mat']);
parametry.fe3dBmax=parametry.fe3dBmax0;
parametry.fe3dBmin=parametry.fe3dBmin0;
zadany=fe3dB.signals.values(:,1);

dana=fe3dB.signals.values(:,2); %uzyskanie wektora z danej

for i=1:length(dana), %wyszukanie przejścia przez zero na koncu
if ((dana(length(dana)-i+1)>=0) && (dana(length(dana)-i)<0)),
break;
end
end

for j=floor(length(dana)/4):length(dana)-1, %wyszukanie przejścia po 1/4 czasu całego
if ((dana(j)<=0) && (dana(j+1)>0)),
break;
end
end

temp=dana(j:length(dana)-i);
dana=temp;
%figure
%plot(dana)
N=length(dana); %ilosc probek

T=parametry.Ts; %okres
F=fft(dana); %transformata
mag=abs([F(1)/N ; F(2:floor(N/2))/(N/2)]); %wyliczenie amplitud skladowych
f=[0:floor((N-1)/2)]*(1/(N*T)); %wektor czestotliwosci

%figure;
%plot(f,mag,'o-')
a0=max(zadany);
a1=max(mag); %harmoniczna podstawowa

parametry.ge3dB=20*log10(a1/a0);
end
#####
%obliczenie zrywu po skoku polozenia
#####
if exist([sciezka_modeli '\rotmax.mat']),
load([sciezka_modeli '\rotmax.mat']);
parametry.rotmax0=parametry.rotmax00;
wynik=rotmax.signals.values(:,1);
parametry.rotmax=max(abs(wynik));
end
#####
%obliczenie zrywu po skoku predkosci
#####
if exist([sciezka_modeli '\rowmax.mat']),
load([sciezka_modeli '\rowmax.mat']);
parametry.rowmax0=parametry.rowmax00;
wynik=rowmax.signals.values(:,1);
parametry.rowmax=max(abs(wynik));
end
#####
%obliczenie rzeczywistej wartosci marginesu wzmoocnienia
#####
if exist([sciezka_modeli '\GM.mat']),
load([sciezka_modeli '\GM.mat']);
wynik=GM.signals.values(:,1);
parametry.GMepsmin_kopia=parametry.GMepsmin;
parametry.GM=max(wynik)/parametry.Keps;
end
#####

```

## Zawartość płyty CD.

\gui – katalog z programem (opis rozmieszczenia i znaczenia plików znajduje się w rozdziale 4.1.);

\tekst – dokumenty tekstowe, rysunki i inne składniki pracy, w tym ten dokument (Praca magisterska.doc);

\materiały – dokumenty tekstowe, prezentacje oraz programy dotyczące metody projektowania odpornych regulatorów serwonapędów w tym [1]: \CD2\prezentacje\ENID\_04\_Deskur.ppt, [2]:

\Cd3\Dobór nastaw regulatorów5a.doc, [3]: \CD2\metoda\Praktyczna metoda projektowania regulatorow odpornych.doc, [6]:, [7]: pid\_lab.pdf, [15]: matlabses7.pdf i inne;

\narzędzia – przydatne narzędzia programistyczne: edytor tekstowy Crimson Editor: cedt370r.exe, program do eksportowania zawartości plików \*.m do dokumentów \*.html, \*.rtf, i \*.tex:

\highlight\highlight.m, program do wyświetlania transmitancji na wykresie Matlaba: tf2text.m,

program do odczytywania pozycji kursora w bieżącym oknie Matlaba: gpos.m