



Imię i nazwisko studenta: Łukasz Karbowniczek
Nr albumu: 175072
Studia drugiego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Automatyka i Robotyka
Specjalność: Komputerowe systemy sterowania

PRACA DYPLOMOWA MAGISTERSKA

Tytuł pracy w języku polskim: Sterownik klimatyzacji stanowisko laboratoryjne

Tytuł pracy w języku angielskim: Air conditioning controller lab stand

Potwierdzenie przyjęcia pracy	
Opiekun pracy	Kierownik Katedry/Zakładu (pozostawić właściwe)
<i>podpis</i>	<i>podpis</i>
dr inż. Piotr Kaczmarek	

Data oddania pracy do dziekanatu:



OŚWIADCZENIE dotyczące pracy dyplomowej zatytułowanej: Sterownik klimatyzacji stanowisko laboratoryjne

Imię i nazwisko studenta: Łukasz Karbowniczek

Data i miejsce urodzenia: 01.07.1995, Wąbrzeźno

Nr albumu: 175072

Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki

Kierunek: automatyka i robotyka

Poziom kształcenia: drugi

Forma studiów: stacjonarne

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2018 poz. 1191 z późn. zm.) i konsekwencji dyscyplinarnych określonych w ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. 2018 poz. 1668 z późn. zm.),¹ a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana praca dyplomowa została opracowana przeze mnie samodzielnie.

Niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. pracy dyplomowej, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji pracy dyplomowej z załączoną wersją elektroniczną.

Gdańsk, dnia
podpis studenta

¹ Ustawa z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce:

Art. 312. ust. 3. W przypadku podejrzania popełnienia przez studenta czynu, o którym mowa w art. 287 ust. 2 pkt 1–5, rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 312. ust. 4. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 5, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o podejrzeniu popełnienia przestępstwa.

STRESZCZENIE

Praca magisterska dotyczy wykonanego stanowiska laboratoryjnego do badania algorytmów sterowania temperaturą powietrza.

W rozdziale pierwszym przedstawiono cel pracy oraz krótko omówiono zastosowane rodzaje algorytmów. Rozdział drugi stanowi przedstawienie zagadnienia regulacji adaptacyjnej z podziałem na jej główne rodzaje i scharakteryzowano niektóre z nich. W kolejnym rozdziale zaprezentowano opis identyfikacji systemu, jej podział, stosowane w niej klasy modeli oraz omówiono szerzej metodę LS (ang. Least Squares) i jej rekursywną postać RLS (ang. Recursive Least Squares). Czwarty rozdział rozwija pojęcie sterowania predykcyjnego i przedstawia szczegółowo algorytm GPC (ang. Generalized Predictive Control) wraz z jego podstawowymi właściwościami. W rozdziale piątym zaprezentowano opis budowy stanowiska laboratoryjnego: założenia projektowe, szczegółowe dane techniczne oraz zdjęcia każdego z elementów systemu. Następny rozdział przedstawia wyniki badań charakterystyki obiektu, wybór struktury modelu oraz opis algorytmu identyfikacji metodą RLS. Przedostatni rozdział stanowi omówienie sposobu implementacji klasycznego sterownika predykcyjnego oraz jego adaptacyjnej wersji. Przedstawiono sposób działania obu typów regulacji na rzeczywistym obiekcie i zamieszczono wyniki eksperymentów w postaci wykresów sygnałów sterujących i wyjściowych. Ostatni rozdział zawiera wnioski końcowe i podsumowanie działań wykonanych w ramach niniejszej pracy magisterskiej.

Słowa kluczowe: Regulacja adaptacyjna, Identyfikacja systemu, Sterowanie predykcyjne, Recursive Least Squares, Generalized Predictive Control, Self-Tuning Generalized Predictive Control.

Dziedzina nauki i techniki, zgodnie z wymogami OECD: Elektrotechnika, Elektronika, Inżynieria informatyczna, Robotyka i automatyka.

ABSTRACT

The master's thesis concerns the laboratory stand prepared for the study of air temperature control algorithms.

The first chapter presents the purpose of the work and briefly discusses the types of algorithms used. The second chapter presents the issue of adaptive regulation divided into its main types and some of them have been characterized. The next chapter presents the description of the system identification, its division, the model classes used in it, and the LS (Least Squares) method and its recursive form of RLS (Recursive Least Squares) are discussed in more detail. The fourth chapter develops the concept of predictive control and presents in detail the GPC algorithm (Generalized Predictive Control) along with its basic properties. Chapter five presents a description of the construction of the laboratory stand: design assumptions, detailed technical data and photos of each of the system components. The next chapter presents the results of research on the characteristics of the object, the choice of model structure and a description of the RLS identification algorithm. The penultimate chapter is a discussion of how to implement the classic predictive controller and its adaptive version. The way of operation of both types of regulation on a real object is presented and the results of experiments in the form of control and output signals are presented. The last chapter contains the final conclusions and a summary of the activities carried out under this master's thesis.

Keywords: Adaptive control, System identification, Model Predictive Control, Recursive Least Squares, Generalized Predictive Control, Self-Tuning Generalized Predictive Control.

Field of science and technology in accordance with OECD requirements: Electrical engineering, Electronic engineering, Information engineering, Robotics and automatic control.

SPIS TREŚCI

Wykaz ważniejszych oznaczeń i skrótów.....	7
1. Wstęp i cel pracy.....	9
2. Regulacja adaptacyjna.....	11
2.1. Wstęp.....	11
2.2. Podstawowe rodzaje regulacji adaptacyjnej.....	12
2.2.1. Harmonogramowanie wzmacniania.....	12
2.2.2. Systemy adaptacyjne z modelem referencyjnym.....	14
2.2.3. Regulator samonastrajalny.....	14
2.2.4. Sterowanie dualne.....	15
3. Identyfikacja systemu.....	17
3.1. Podział metod identyfikacji.....	18
3.2. Metoda Najmniejszych Kwadratów.....	18
3.3. Rekurencyjna Metoda Najmniejszych Kwadratów.....	20
3.3.1. Algorytm.....	20
3.3.2. Identyfikacja w czasie rzeczywistym.....	21
3.4. Identyfikowalność.....	22
3.5. Weryfikacja modelu.....	22
3.6. Kryterium informacyjne Akaike'go.....	23
4. Sterowanie predykcyjne.....	25
4.1. Idea sterowania predykcyjnego.....	25
4.2. Rodzaje sterowników predykcyjnych.....	27
4.2.1. Dynamic Matrix Control.....	27
4.2.2. Model Algorithmic Control.....	27
4.2.3. Predictive Functional Control.....	28
4.2.4. Extended Prediction Self-Adaptive Control.....	28
4.2.5. Extended Horizon Adaptive Control.....	28
4.3. Generalized Predictive Control.....	28
4.3.1. Model.....	29
4.3.2. Funkcja kosztu.....	29
4.3.3. Algorytm sterowania.....	30
4.3.4. Dostrajanie regulatora.....	31
5. Budowa stanowiska.....	35
5.1. Założenia projektowe.....	35
5.2. Nagrzewnica powietrza.....	36
5.3. Zasilanie układu.....	40
5.4. Jednostka sterująca.....	41
5.5. Czujniki temperatury.....	43
5.6. Moduł sterownika PWM.....	44

6.	Implementacja algorytmu identyfikacji.....	45
6.1.	Badanie charakterystyki obiektu.....	45
6.2.	Wybór struktury modelu.....	46
6.2.1.	Identyfikacja obiektu modelem ARX pierwszego rzędu.....	47
6.2.2.	Identyfikacja obiektu modelem ARX drugiego rzędu.....	49
6.2.3.	Identyfikacja obiektu modelem ARX trzeciego rzędu.....	51
6.3.	Zestawienie wyników.....	53
6.4.	Kryterium informacyjne Akaike'go.....	53
6.5.	Identyfikacja obiektu w czasie rzeczywistym.....	54
6.6.	Wnioski.....	57
7.	Implementacja algorytmu sterownika.....	59
7.1.	Generalized Predictive Control.....	59
7.1.1.	Horyzont predykcji.....	61
7.1.2.	Sterowanie w warunkach stałych.....	62
7.1.3.	Sterowanie w warunkach zmiennych.....	64
7.1.4.	Wnioski.....	65
7.2.	Self-Tuning Generalized Predictive Control.....	65
7.2.1.	Sterowanie w warunkach zmiennych.....	66
7.2.2.	Wnioski.....	69
8.	Podsumowanie.....	71
	Wykaz literatury.....	73
	Wykaz rysunków.....	75
	Wykaz tabel.....	77
	Dodatek A.....	79
	Dodatek B.....	81
	Dodatek C.....	83

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

Oznaczenia

- $y(t)$ – sygnał wyjściowy w dyskretnej chwili t
 $u(t)$ – sygnał sterujący w dyskretnej chwili t
 $\eta(t)$ – sygnał szumu białego w dyskretnej chwili t
 d – opóźnienie transportowe obiektu
 z^{-1} – operator opóźnienia jednostkowego
 $A(z^{-1})$ – wielomian sygnału wyjściowego modelu ARX/CARIMA
 $B(z^{-1})$ – wielomian sygnału wejściowego modelu ARX/CARIMA
 $C(z^{-1})$ – wielomian sygnału zakłócającego modelu ARX/CARIMA
 na – stopień wielomianu $A(z^{-1})$
 nb – stopień wielomianu $B(z^{-1})$
 nc – stopień wielomianu $C(z^{-1})$
 φ^T – wektor regresji
 θ – wektor parametrów modelu
 ma – liczba estymowanych parametrów wielomianu $A(z^{-1})$
 mb – liczba estymowanych parametrów wielomianu $B(z^{-1})$
 λ – współczynnik zapominania
 τ – czas ustalania się odpowiedzi skokowej obiektu
 T_s – okres próbkowania
 $G_{ARX(ma,mb)}$ – transmitancia dyskretna modelu $ARX(ma,mb)$
 \hat{y} – wektor przewidywanych wyjść obiektu określonych na horyzoncie predykcji
 u – wektor przyszłych wartości sygnału sterującego
 w – wektor przyszłych wartości trajektorii referencyjnej
 q – współczynnik wagowy przyrostów sterowania
 H – horyzont predykcji
 L – horyzont sterowania

Skróty

- AIC – Akaike Information Criterion
AR – Autoregressive
ARMA – Autoregressive Moving Average
ARMAX – Autoregressive Moving Average with Exogenous Input
ARX – Autoregressive with Exogenous Input
CARIMA – Controlled Autoregressive Integrated Moving Average
DC – Dual Control
DMC – Dynamic Matrix Controller
EF-RLS – Exponential Forgetting Recursive Least Squares
EHAC – Extended Horizon Adaptive Control

EPSAC – Extended Prediction Self-Adaptive Control
GPC – Generalized Predictive Control
GS – Gain Scheduling
IV – Instrumental Variable
LMS – Least Mean Square
LS – Least Squares
MAC – Model Algorithmic Control
MPC – Model Predictive Control
MPHC – Model Predictive Controller
MRAS – Model Reference Adaptive Systems
PFC – Predictive Functional Control
PID – Proportional Integral Derivative
PRBS – Pseudo Random Binary Sequence
PWM – Pulse Width Modulation
RLS – Recursive Least Squares
ST-GPC – Self-Tuning Generalized Predictive Control
STR – Self-Tuning Regulator

1. WSTĘP I CEL PRACY

Celem pracy magisterskiej było wykonanie stanowiska laboratoryjnego do badania algorytmów sterowania temperaturą powietrza w systemach klimatyzacji. Niniejszy projekt można podzielić na trzy etapy: budowę stanowiska laboratoryjnego, implementację algorytmów sterowania predykcyjnego (klasycznego i adaptacyjnego) na zbudowanym obiekcie oraz wykonanie serii eksperymentów sprawdzających charakterystykę obiektu i układu zamkniętego.

Prace rozpoczęto od sformułowania założeń projektowych, które określiły, jakie cechy powinien posiadać obiekt sterowania. Zbudowano stanowisko do sterowania temperaturą powietrza, które przepływa przez kanał. Prędkość przepływu powietrza wymuszana jest (oraz regulowana) za pomocą wentylatora podłączonego do modułu PWM (ang. Pulse Width Modulation). Dzięki temu obiekt sterowania posiada zmienne opóźnienie transportowe, które powoduje, że układ charakteryzuje się zmiennymi warunkami operacyjnymi.

Przed przystąpieniem do realizacji części właściwej pracy, scharakteryzowano pojęcie regulacji adaptacyjnej i dokonano przeglądu metod należących do tej klasy sterowania. Opisano również ideę regulacji predykcyjnej oraz opisano szerzej metodę Generalized Predictive Control (GPC), jako wybraną strategię sterowania. Algorytm GPC działa w oparciu o model obiektu, którego parametry identyfikowane są metodą RLS (ang. Recursive Least Squares). Objasniono sposób wykorzystania tego algorytmu do identyfikacji w czasie rzeczywistym.

Dokonano implementacji adaptacyjnego sterownika predykcyjnego zbudowanego w oparciu o połączenie dwóch, zaimplementowanych wcześniej algorytmów: GPC i RLS. Sporządzono wykresy sygnałów wejściowych i wyjściowych oraz zmiennych parametrów modelu obiektu. Wykonano serie badań dla różnych nastaw prędkości obrotowej wentylatora, czyli tym samym dla różnego opóźnienia transportowego obiektu.

Implementację algorytmów wykonano na komputerze Raspberry Pi w języku programowania Python przy użyciu środowiska PyCharm w wersji profesjonalnej. Wybrano ten rodzaj oprogramowania, ponieważ pozwala on na bardzo intuicyjne konfigurowanie Raspberry Pi w trybie zdalnym, przez co uniknięto konieczności stosowania dodatkowych podzespołów (m.in. monitora, klawiatury). Wyniki eksperymentów zapisywano do pliku tekstowego a następnie tworzono wykresy w środowisku Matlab.

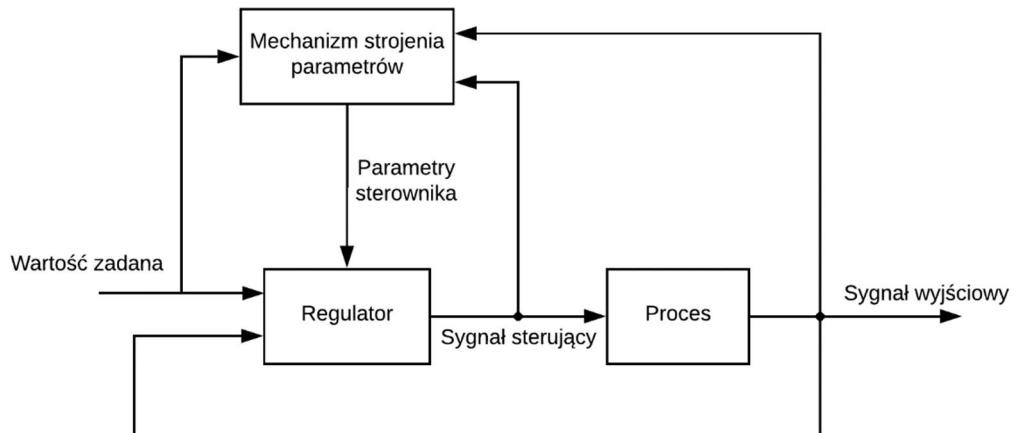
2. REGULACJA ADAPTACYJNA

2.1. Wstęp

W życiu codziennym wyrażenie „adaptacja” oznacza zmianę zachowania po to, aby dostosować się do nowych okoliczności. Uogólniając to pojęcie na systemy sterowania, regulator adaptacyjny potrafi zmieniać swoje zachowanie w zależności od zmian dynamiki procesu i charakteru zakłóceń.

Można zadać sobie pytanie o różnicę pomiędzy sterownikiem adaptacyjnym a klasycznym sterownikiem ze sprzężeniem zwrotnym. Kiedy na układ oddziałują zakłócenia, które wcześniej się nie pojawiały lub nie są dobrze znane, występuje zjawisko niedopasowania parametrów regulatora do właściwości obiektu. W konsekwencji, jakość procesu regulacji ulega pogorszeniu i mogą występować problemy z nadmiernym uchybem i zbyt dużymi oscylacjami sygnału wyjściowego. Rozwiążaniem problemu może być estymacja zakłóceń oddziałujących na obiekt i odpowiednia zmiana parametrów regulacji, które w klasycznym sterowniku ze sprzężeniem zwrotnym nie występują.

W sterowniku adaptacyjnym istnieje mechanizm odpowiedzialny za dostrajanie jego parametrów do zmian obiektu. Regulator adaptacyjny posiada więc dwie pętle: sprzężenia zwrotnego oraz strojenia parametrów (rys. 2.1). Zwykle pętla odpowiedzialna za strojenie parametrów jest wolniejsza od pętli sprzężenia zwrotnego, co oznacza, że w większości przypadków zmiana parametrów występuje rzadziej niż zmiana sygnału sterującego [1].



Rys. 2.1. Schemat blokowy sterownika adaptacyjnego [1]

Sterowanie adaptacyjne znajduje swoje zastosowanie w procesach, które są bardzo skomplikowane, mało znane lub zmieniają się w czasie w sposób nieprzewidywalny. Zapotrzebowanie na tego typu algorytmy wzrasta w przemyśle chemicznym, ponieważ stopień zaawansowania procesów chemicznych ciągle rośnie. Przykładowym procesem w tym sektorze, wspomaganym przez regulację adaptacyjną, jest reakcja polimeryzacji [2].

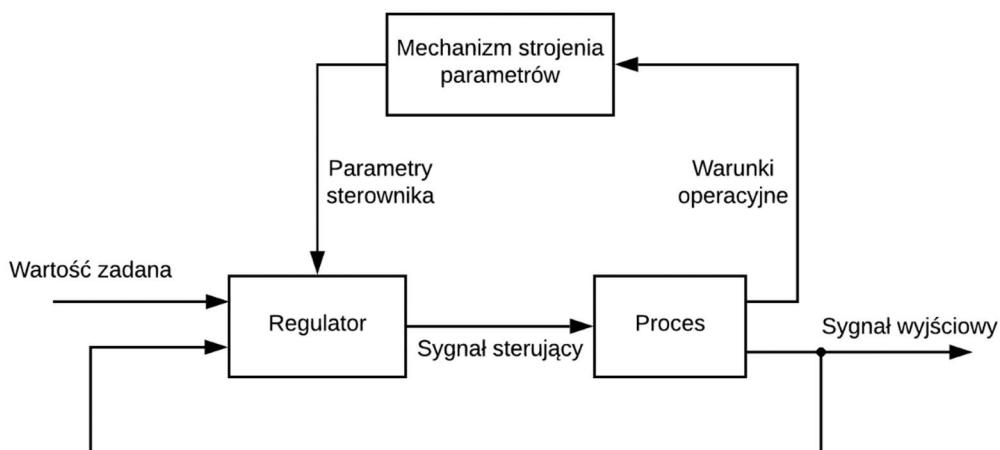
2.2. Podstawowe rodzaje regulacji adaptacyjnej

Pierwsze sterowniki adaptacyjne powstały w latach pięćdziesiątych dwudziestego wieku i miały one prostą budowę. Zauważono potrzebę zapewnienia dobrej jakości regulacji dla kilku różnych punktów pracy, co mogło zostać osiągnięte poprzez zmianę nastaw regulatora. W późniejszych latach korzystano z modelowania obiektów dynamicznych i estymacji parametrów, aby sterowanie było jak najdokładniejsze a nowoczesna teoria sterowania pozwala na implementację sterowników samonastrajalnych. Wszystkie wymienione wyżej typy regulacji adaptacyjnej zostały omówione w kolejnych podrozdziałach. Opisana została również teoria sterowania dualnego [1].

2.2.1. Harmonogramowanie wzmacnienia

Podejście oparte na harmonogramowaniu wzmacnienia GS (ang. Gain Scheduling) jest jednym z najprostszych typów sterowania adaptacyjnego i polega na dekompozycji zadania projektowania regulatora nieliniowego na grupę zadań projektowania regulatorów liniowych i odpowiednim przełączaniu pomiędzy nimi w taki sposób, aby zapewnić, że docelowy nieliniowy system będzie miał pożądaną dynamikę.

Koncepcja sterowania adaptacyjnego wykorzystującego harmonogramowanie wzmacnienia przedstawiona jest na rys. 2.2. Na schemacie blokowym można zaobserwować pętlę sterowania ze sprzężeniem zwrotnym oraz pętlę sterowania parametrami regulatora.

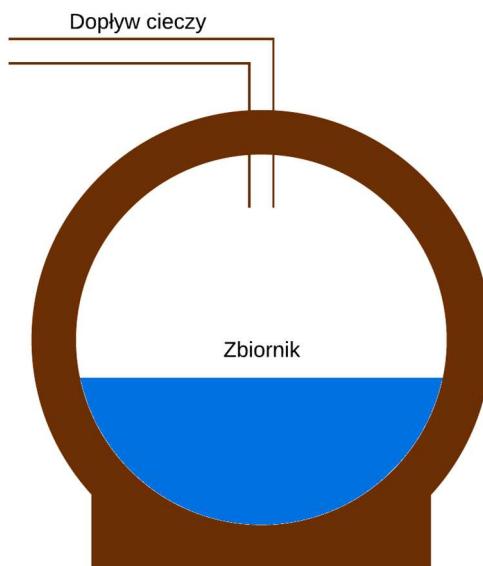


Rys. 2.2. Schemat blokowy systemu harmonogramowania wzmacnienia [1]

Zadaniem mechanizmu harmonogramowania wzmacnienia jest przyporządkowywanie wartości parametrów strojenia regulatora odpowiednim wartościom warunków operacyjnych. Harmonogramowanie może być realizowane za pomocą funkcji bądź tablic nastawnych.

Przykładem obiektu, który nadaje się do tego typu sterowania jest zbiornik o niejednolitym przekroju poprzecznym np. w kształcie kuli (rys. 2.3). Zadanie polega na napełnianiu zbiornika w taki sposób, aby poziom cieczy znajdującej się w nim narastał jednostajnie. Regulator tradycyjny np. PID (ang. Proportional Integral Derivative), z parametrami ustalonymi

dla konkretnego punktu pracy, nie będzie dobrze regulować wartości sterującej, podczas gdy punkt pracy się zmieni. Jeśli zastosujemy nastawy regulatora dla zbiornika napełnionego w połowie dla stanu, w którym zbiornik ten jest całkowicie pusty, poziom wody będzie się podniósł bardzo szybko. Aby poziom cieczy wzrastał równomiernie, należy zmniejszyć natężenie przepływu dla pustego i prawie pełnego zbiornika i odpowiednio zwiększyć je, gdy jest on napełniony w połowie. Stąd istnieje potrzeba stosowania adaptacji w tego typu aplikacjach. Zagadnieniem krytycznym podczas implementacji takiego sterownika jest optymalne określenie tablicy nastaw lub funkcji przyporządkowującej poszczególne nastawy wzmocnienia w zależności od aktualnego stanu obiektu. Oznacza to, że wymagana jest dobra znajomość zmienności obiektu. Nie przeprowadza się tutaj estymacji parametrów obiektu [3].



Rys. 2.3. Zbiornik o niejednolitym przekroju poprzecznym

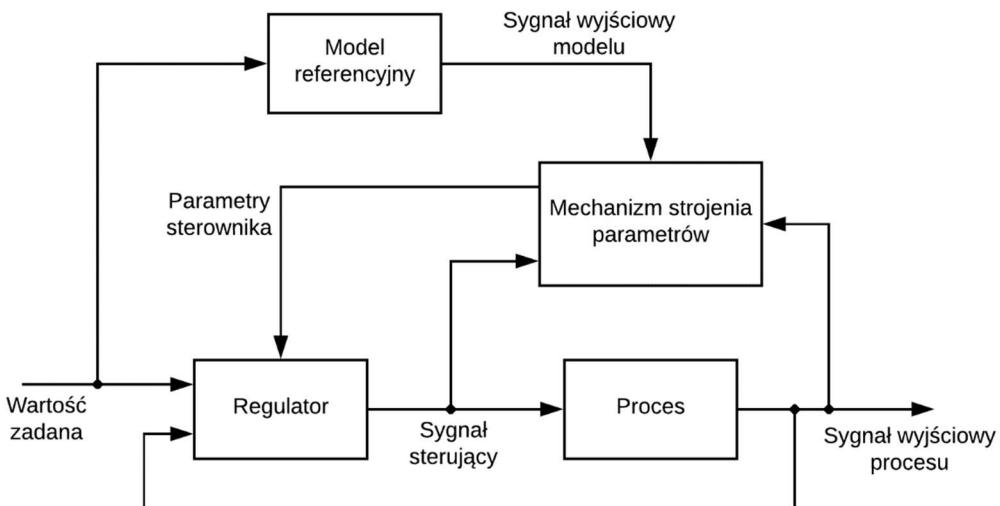
Mechanizm adaptacyjnego strojenia wzmocnienia stanowi ulepszenie klasycznego regulatora PID ale także i wielu innych sterowników. Wykorzystywany jest w lotnictwie do systemów sterowania samolotem. Masa samolotu spada wraz z ubytkiem paliwa, przez co wymagana jest zmiana parametrów procesu sterowania.

Sterowniki te często używane są również w przemyśle morskim. Funkcja autopilota na statku realizowana jest zazwyczaj za pomocą regulatora PID, jednak podczas silnego wiatru oraz w zależności od całkowitej masy statku jego prędkość może ulec zmianie. Wyeliminowanie tych zmian osiągane jest przez zmianę wzmocnienia układu i stałych czasowych: całkowania i różniczkowania.

Harmonogramowanie wzmocnienia wykorzystuje się również w motoryzacji do kontrolowania proporcji mieszanek paliwowo-powietrznej czy w przemyśle chemicznym do kontrolowania skali pH w zbiornikach z mieszankami kwasowo-zasadowymi [1, 3].

2.2.2. Systemy adaptacyjne z modelem referencyjnym

Systemy adaptacyjne z modelem referencyjnym MRAS (ang. Model–Reference Adaptive System) posiadają dwie pętle sterowania. Jedna pętla, nazywana zewnętrzna, jest odpowiedzialna za organizację regulacji ze sprzężeniem zwrotnym, natomiast druga (pętla wewnętrzna) dostosowuje parametry sterownika w taki sposób, aby błąd regulacji, który jest różnicą pomiędzy sygnałem wyjściowym procesu a sygnałem wyjściowym modelu referencyjnego, był jak najmniejszy. Poniżej przedstawiono podstawową strukturę takiego sterownika (rys. 2.4) [1].



Rys. 2.4. Schemat blokowy przedstawiający strukturę MRAS [1]

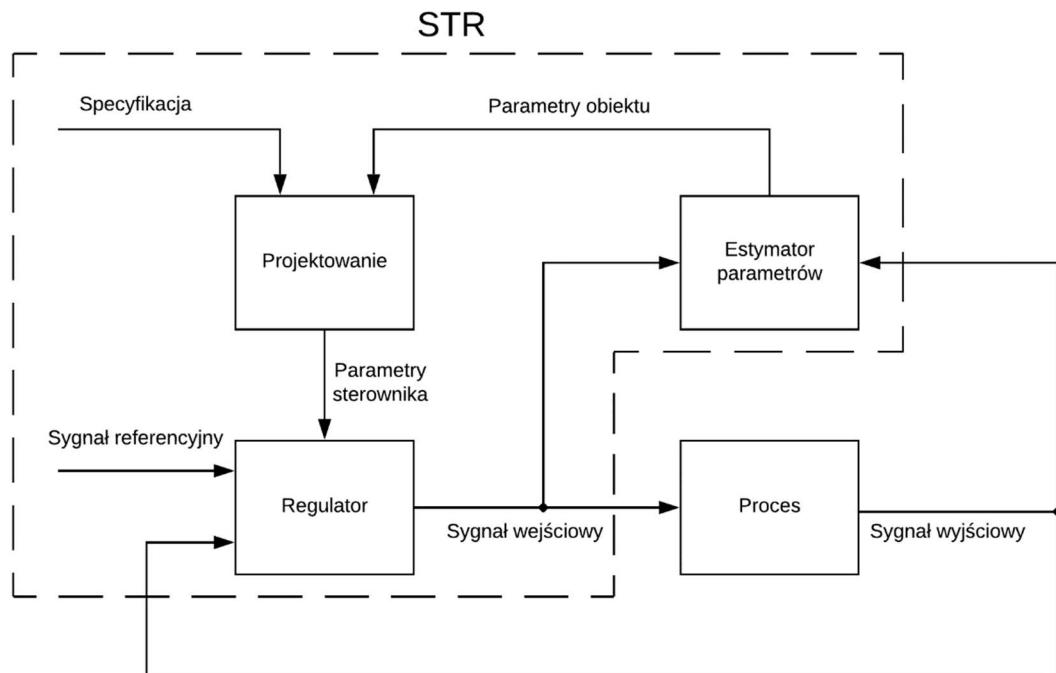
Sterowniki adaptacyjne w wersji MRAS używają modeli referencyjnych, jako źródła specyfikacji regulacji. Modele te wykazują jak powinien wyglądać idealny sygnał wyjściowy, pobudzony odpowiednim sygnałem wejściowym oraz determinującą takie cechy układu zamkniętego, jak np.: stała czasowa, pasmo przenoszenia, współczynnik tłumienia. W celu zapewnienia przyczynowości sterownika oraz stabilności układu zamkniętego stosuje się następujące założenia:

- odwrotność transmitancji obiektu sterowania oraz transmitancja modelu referencyjnego są stabilne,
- względny stopień transmitancji modelu referencyjnego musi być co najmniej równy względнемu stopniowi transmitancji obiektu sterowania [3].

2.2.3. Regulator samonastrajalny

Regulatory samonastrajalne STR (ang. Self-Tuning Regulator) są jednymi z najbardziej popularnych sterowników stosowanych w systemach adaptacyjnych. Dotychczas rozważane formy regulacji adaptacyjnej miały charakter bezpośredni. Mechanizm strojenia parametrów jasno

określał w jaki sposób aktualizowane są parametry obiektu sterowania. Zasada działania regulatorów z grupy STR jest nieco inna, co obrazuje poniższy schemat (rys. 2.5).



Rys. 2.5. Idea regulatora samonastrajalnego [1]

Podobnie jak we wcześniejszych przykładach, występują tu dwie pętle sterowania: wewnętrzna i zewnętrzna. Pętla wewnętrzna realizuje sterowanie ze sprzężeniem zwrotnym, natomiast pętla zewnętrzna odpowiedzialna jest za estymację parametrów obiektu na podstawie aktualnych pomiarów sygnałów wejściowych i wyjściowych. Projektowanie sterownika odbywa się z wykorzystaniem estymowanych wartości parametrów obiektu sterowania. Innymi słowy, regulator samonastrajalny automatycznie wyznacza parametry obiektu i wykorzystuje je do zrealizowania pożąданej specyfikacji. Oznacza to, że otrzymane współczynniki modelu w wyniku procesu identyfikacji, traktowane są jako prawdziwe parametry obiektu (ang. Certainly Equivalent Principle – CEP) [1].

Istnieje wiele odmian sterowników z rodziny STR, dlatego też schemat z rys. 2.5 należy traktować poglądowo. Jest mnóstwo możliwych sposobów realizacji STR z uwagi na elastyczność w doborze algorytmu estymacji parametrów oraz rodzaju sterownika.

2.2.4. Sterowanie dualne

Omawiane wcześniej rodzaje regulacji adaptacyjnej oparte są w dużej mierze na podejściu heurystycznym. Same założenia początkowe sprawiają, że implementowane sterowniki mają pewne ograniczenia. Przykładem może być to, że identyfikowany obiekt traktujemy jako prawdziwy. Należy się zastanowić nad tym, czy możliwe jest lepsze podejście niż skorzystanie z zasady CEP. Teoria sterowania dualnego pozwala odpowiedzieć na to pytanie.

Sterownik optymalny posiada bardzo interesującą właściwość. Podczas, gdy parametry modelu obiektu odbiegają od rzeczywistych, podążanie za trajektorią zadaną staje się mniej dokładne. To z kolei poprawia jakość estymacji parametrów i polepsza jakość regulacji. Sterowanie optymalne wprowadza więc odpowiedni balans pomiędzy utrzymaniem dobrej jakości regulacji a małymi błędami estymacji. Termin „sterowanie dualne” powstał po to, aby móc wyjaśnić powyższe zjawisko.

Istnieje wiele analogicznych przykładów wyjaśniających na czym polega sterowanie dualne. Jednym z nich jest odwołanie się do sterowania samochodem. Chcąc się udać do pewnego miejsca docelowego można się poruszać autem po prostej drodze z jednostajną prędkością tak, aby dotrzeć do celu najszybciej oraz najtaniej. Prowadząc samochód po raz pierwszy, można podczas podróży spróbować nim zahamować oraz skręcić po to, aby nauczyć się tych manewrów i móc je wykorzystać w przyszłości. Ta analogia pokazuje, że można pobudzić obiekt sygnałem o większej mocy i pogorszyć chwilowo jakość sterowania, aby uzyskać więcej informacji o obiekcie i wykorzystać je polepszając sterowanie układem w przyszłości.

Sterowanie dualne jest zagadnieniem czysto teoretycznym i dość skomplikowanym. Oznacza to, że teoria ta jest niemożliwa do wykorzystania w praktyce, jednak pokazuje ograniczenia wynikające ze stosowania metody CEP oraz sugeruje w jaki sposób można usprawnić proces regulacji [1].

3. IDentyfikacja systemu

Identyfikacja systemu (bądź obiektu lub też procesu) definiowana jest jako modelowanie systemów na podstawie danych doświadczalnych. Obok modelowania fizycznego, identyfikacja systemu jest innym sposobem na uzyskanie modelu.

W modelowaniu fizycznym (fenomenologicznym) model wyprowadzany jest z ogólnie znanych praw fizycznych, chemicznych itp., wykorzystując znane zasady matematyczne. Wymagana jest tutaj obszerna znajomość zjawisk zachodzących w przyrodzie oraz samego modelowanego obiektu. Główną zaletą tych modeli jest to, że są one ważne dla wszystkich procesów tego typu i dla różnych warunków operacyjnych. Podejście fizyczne stosuje się w wielu gałęziach przemysłu, tworząc głównie modele: samolotów, statków, silników czy pieców. Takie modele są bardzo dokładne i mogą posłużyć jako zamiennik rzeczywistego prototypu do testowania, co znacznie redukuje koszty związane z wytwarzaniem końcowego produktu.

Identyfikacja systemu pozwala na zbudowanie modelu w oparciu o dane pomiarowe. Często, jeśli to możliwe, wykorzystuje się wstępную wiedzę o modelowanym procesie, która umożliwia szybsze i dokładniejsze dopasowanie modelu. To podejście nazywane jest też podejściem eksperymentalnym (instrumentalnym) bądź po prostu identyfikacją. Obiekt traktowany jest tutaj jako czarna skrzynka, na temat której nie jest wymagana szersza wiedza. Jest to niewątpliwie duża zaleta w stosunku do podejścia fizycznego, co powoduje, że metody eksperymentalne są bardzo praktyczne. W identyfikacji systemu stosuje się modele dynamiczne o znanych trajektoriach:

- wejściowo-wyjściowych, czyli takich, w których mierzalne są zarówno sygnały wejściowe jak i wyjściowe,
- wyjściowych – przebieg sygnału wejściowego jest w tym przypadku niemierzalny (nieznany).

Na podstawie danych pomiarowych i przyjętej struktury wyznaczany jest model matematyczny, który jest ważny tylko dla badanego obiektu lub systemu i tych samych warunków operacyjnych, w jakich przeprowadzano eksperyment. Określenie struktury jest bardzo ważnym zagadnieniem i oznacza wybranie: typu modelu z danej klasy oraz odpowiedniego rzędu modelu. Jeśli liczba parametrów będzie zbyt duża, może to prowadzić do niepotrzebnych problemów w obliczeniach przy wyznaczaniu ocen parametrów. Z kolei za mały rząd modelu skutkuje mniejszą dokładnością. W przeciwnieństwie do modelowania fizycznego, modelowanie instrumentalne może być wykonalne tylko dla istniejących systemów.

Identyfikacja systemu obejmuje szeroki zakres zastosowań w różnych dziedzinach. W automatyce i teorii sterowania służy w głównej mierze do syntezy regulatorów klasycznych, konstrukcji sterowników predykcyjnych i sterowników adaptacyjnych oraz symulacji. Metody identyfikacyjne z powodzeniem stosuje się również w przetwarzaniu sygnałów m.in. do: filtracji adaptacyjnej, analizy widmowej, korekcji kanałów telekomunikacyjnych, kodowania sygnałów, aktywnego tłumienia hałasu i drgań, detekcji uszkodzeń, rozpoznawania obrazów, predykcji. Potrzeba użycia systemów identyfikujących znaczaco rośnie w ekonomii, w której używa się modeli matematycznych do analizy, prognozowania i sterowania [4].

3.1. Podział metod identyfikacji

Ze względu na obecność parametrów, metody identyfikowania systemów można podzielić na: nieparametryczne i parametryczne. Modele nieparametryczne nie posiadają struktury. Najczęściej identyfikacja nieparametryczna jest zagadnieniem trudniejszym. Wynikiem jej działania jest model w postaci funkcji lub krzywej, niekoniecznie opisanej wektorem parametrów o skończonym wymiarze. Można wyróżnić następujące metody nieparametryczne:

- analiza przebiegów przejściowych – model w postaci sygnału wyjściowego, wywołany sygnałem impulsowym lub sygnałem skokowym,
- analiza częstotliwościowa – sinusoidalny sygnał wejściowy, w stanie ustalonym w modelu liniowym sygnał wyjściowy jest również sinusoidalny,
- analiza korelacyjna – biały szum jako sygnał wejściowy, estymator funkcji wagowej dany jest przez funkcję korelacji wzajemnej sygnału wejściowego,
- analiza widmowa – podzielenie wzajemnej gęstości widmowej sygnałów: wyjściowego i wejściowego przez gęstość widmową sygnału wejściowego pozwala estymować transmitancję w dziedzinie częstotliwości.

W podejściu parametrycznym proces rozpoczyna się od utworzenia struktury modelu, ponieważ to od niej zależy ściśle liczba identyfikowanych parametrów, które są rezultatem procesu identyfikacji. Etap wyznaczania parametrów modelu nazywa się ich estymacją. Do parametrycznych metod zaliczają się między innymi:

- Metoda Najmniejszych Kwadratów (ang. LS – Least Squares),
- Metoda Najmniejszych Średnich Kwadratów (ang. LMS – Least Mean Squares),
- Rekursywna Metoda Najmniejszych Kwadratów (ang. RLS – Recursive Least Squares),
- Metoda Zmiennych Instrumentalnych (ang. IV – Instrumental Variables) [4].

3.2. Metoda Najmniejszych Kwadratów

Metoda Najmniejszych Kwadratów jest jedną z najstarszych metod identyfikacji i należy do grupy metod bazujących na wyznaczaniu błędu predykcji. Stosuje się ją dla dyskretnych modeli stochastycznych:

- AR (ang. Autoregressive),
- ARX (ang. Autoregressive with Exogenous Input),
- ARMA (ang. Autoregressive Moving Average),
- ARMAX (ang. Autoregressive Moving Average with Exogenous Input).

W niniejszym podrozdziale zaprezentowano opis metody LS dla modelu ARX. Rozważając przypadek skalarny, model ten opisuje się następującą zależnością:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t) + \eta(t), \quad (3.1)$$

gdzie t jest czasem dyskretnym oraz:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-n}, \quad (3.2)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}, \quad (3.3)$$

$\eta(t)$ oznacza szum biały, $y(t)$ jest sygnałem wyjściowym, $u(t)$ to sygnał wejściowy a d jest opóźnieniem procesu. Przyjmując, że d wynosi 1, z modelu (3.1) można wyznaczyć próbki sygnału wyjściowego na chwilę t , czyli przekształcić je do następującej postaci:

$$y(t) = \boldsymbol{\varphi}^T(t)\boldsymbol{\theta} + \eta(t), \quad (3.4)$$

gdzie:

$$\boldsymbol{\varphi}^T(t) = [-y(t-1), \dots, -y(t-na), u(t-1), \dots, u(t-nb)], \quad (3.5)$$

$$\boldsymbol{\theta} = [a_1, \dots, a_{na}, b_0, \dots, b_{nb}]^T. \quad (3.6)$$

Podstawowym zadaniem metody LS jest wyznaczenie oszacowania $\hat{\boldsymbol{\theta}}$ wektora parametrów $\boldsymbol{\theta}$ na podstawie zebranych pomiarów $\boldsymbol{\varphi}^T(1), \dots, \boldsymbol{\varphi}^T(N)$, gdzie N jest liczbą próbek. Wartość $\hat{\boldsymbol{\theta}}$ wyznaczana jest z kryterium jakości (minimalizującego sumę kwadratów błędów estymacji $e(t)$) opisanego wzorem:

$$J_N(\boldsymbol{\theta}) = \sum_{t=1}^N [e(t)]^2, \quad (3.7)$$

gdzie:

$$e(t) = y(t) - \boldsymbol{\varphi}^T(t)\boldsymbol{\theta}. \quad (3.8)$$

Równanie (3.7) można zapisać również w postaci:

$$J_N(\boldsymbol{\theta}) = \sum_{t=1}^N y^2(t) - 2 \left[\sum_{t=1}^N y(t)\boldsymbol{\varphi}^T(t) \right] \boldsymbol{\theta} + \boldsymbol{\theta}^T \left[\sum_{t=1}^N \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t) \right] \boldsymbol{\theta}. \quad (3.9)$$

Minimum funkcji $J_N(\boldsymbol{\theta})$ wynosi:

$$\nabla_{\boldsymbol{\theta}} J_N(\boldsymbol{\theta}) = -2 \sum_{t=1}^N y(t)\boldsymbol{\varphi}^T(t) + 2 \left[\sum_{t=1}^N \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t) \right] \boldsymbol{\theta}, \quad (3.10)$$

pod warunkiem, że:

$$\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}^T} J_N(\boldsymbol{\theta}) = 2 \left[\sum_{t=1}^N \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t) \right] > 0. \quad (3.11)$$

Wyznaczenie estymatora parametrów realizowane jest przez przyrównanie gradientu funkcji kryterium jakości (3.10) do zera:

$$\hat{\boldsymbol{\theta}}(N) = \mathbf{R}^{-1}(N) \sum_{t=1}^N y(t)\boldsymbol{\varphi}^T(t), \quad (3.12)$$

gdzie:

$$\mathbf{R}(N) = \sum_{t=1}^N \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t) \quad (3.13)$$

jest nieujemnie określona macierzą regresji. Nierówność (3.11) jest spełniona wtedy i tylko wtedy, gdy macierz regresji jest nieosobliwa. W przeciwnym przypadku wyznaczone parametry będą nieprawidłowe. Równanie (3.12) jest podstawą do wyprowadzenia estymatora rekurencyjnego, opisanego w kolejnym podrozdziale [4].

3.3. Rekurencyjna Metoda Najmniejszych Kwadratów

Rekurencyjna Metoda Najmniejszych Kwadratów (RLS) jest uogólnieniem wsadowej Metody Najmniejszych Kwadratów (LS). Estymacja parametrów w rekurencyjnych metodach identyfikacji systemów polega na wyznaczaniu wartości oszacowania na chwilę obecną poprzez modyfikację estymatora z poprzedniej chwili czasu. Jeśli znamy estymator $\hat{\theta}(t-1)$, który jest wyznaczony na chwilę $t-1$, to po uwzględnieniu „drobnej poprawki” można wyznaczyć estymator $\hat{\theta}(t)$.

W metodach rekurencyjnych nie ma potrzeby używania wszystkich zapisanych danych jednocześnie (w przeciwieństwie do metod wsadowych), co sprawia, że nie zużywają one dużo pamięci oraz mogą być stosowane w czasie rzeczywistym m.in. do: filtracji adaptacyjnej, identyfikacji w systemach adaptacyjnych oraz algorytmów śledzących zmieniające się parametry [4].

3.3.1. Algorytm

Rozważając ponownie układ skalarny, czyli $\dim[y] = 1$, możemy wprowadzić poniższe oznaczenie:

$$\mathbf{P}(t) = \left[\sum_{s=1}^N \boldsymbol{\varphi}(s)\boldsymbol{\varphi}^T(s) \right]^{-1}. \quad (3.14)$$

Ponieważ:

$$\mathbf{P}^{-1}(t) = \mathbf{P}^{-1}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t), \quad (3.15)$$

otrzymujemy następującą zależność (argument t pozwala podkreślić zależność wartości estymatora $\hat{\theta}$ od czasu):

$$\begin{aligned} \hat{\theta}(t) &= \mathbf{P}(t) \left[\sum_{s=1}^{t-1} \boldsymbol{\varphi}(s)y(s) + \boldsymbol{\varphi}(t)y(t) \right] = \mathbf{P}(t)[\mathbf{P}^{-1}(t-1)\hat{\theta}(t-1) + \boldsymbol{\varphi}(t)y(t)] = \\ &= \hat{\theta}(t-1) + \mathbf{P}(t)\boldsymbol{\varphi}(t)[y(t) - \boldsymbol{\varphi}^T(t)\hat{\theta}(t-1)]. \end{aligned} \quad (3.16)$$

W ostateczności:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mathbf{k}(t)e(t), \quad (3.17)$$

$$\mathbf{k}(t) = \mathbf{P}(t)\boldsymbol{\varphi}(t), \quad (3.18)$$

$$e(t) = y(t) - \boldsymbol{\varphi}^T(t)\hat{\theta}(t-1). \quad (3.19)$$

W równaniu (3.19) wielkość $e(t)$ należy traktować jako błąd predykcji, ponieważ jest to różnica pomiędzy mierzonym sygnałem $y(t)$ a predykcją jednokrokową $\hat{y}(t|t-1)$ sygnału wyjściowego dokonaną w chwili t i opartą na modelu z estymatorem $\hat{\theta}(t-1)$. Wektor $k(t)$ jest interpretowany jako wzmacnienie, które odzwierciedla jak bardzo wielkość błędu $e(t)$ wpływa na wektor parametrów $\hat{\theta}$. Im większy błąd, tym bardziej estymowane parametry odbiegają od prawidłowej wartości.

Podczas implementacji algorytmu RLS można zauważyc, że równanie (3.15) jest bardzo niewygodne z uwagi na konieczność odwracania macierzy P w każdej iteracji programu. Korzystając z lematu o odwracaniu macierzy, zależność (3.15) można przekształcić do postaci:

$$P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1 + \varphi^T(t)P(t-1)\varphi(t)}, \quad (3.20)$$

w której zamiast odwracania macierzy występuje jedynie dzielenie przez wartość skalarną. Dodatkowo mnożąc obie strony równania (3.20) przez $\varphi(t)$ otrzymujemy:

$$k(t) = \frac{P(t-1)\varphi(t)}{1 + \varphi^T(t)P(t-1)\varphi(t)}. \quad (3.21)$$

Algorytm RLS składa się więc z równań (3.17), (3.19), (3.20) i (3.21). Do poprawnej jego pracy wymagane jest przyjęcie wartości początkowych $\hat{\theta}(0)$ i $P(0)$. $P(t)$ jest macierzą kowariancji. Wynika z tego, że $P(0)$ jest informacją o zaufaniu, jaką obarczona jest wartość parametrów $\hat{\theta}(0)$. Jeżeli nie posiadamy żadnej wiedzy o identyfikowanym obiekcie, początkowa wartość $\hat{\theta}(0)$ powinna wynosić 0. Natomiast przyjęcie dużych wartości na elementach diagonalnych macierzy $P(0)$ spowoduje, że oceny parametrów zaczynają szybko dążyć do prawdziwych wartości [4].

3.3.2. Identyfikacja w czasie rzeczywistym

Rekurencyjna Metoda Najmniejszych Kwadratów może być implementowana z powodzeniem do identyfikacji w czasie rzeczywistym (online) poprzez zastosowanie współczynnika zapominania λ jako mnożnika w postaci okna wykładniczego (EF-RLS) w minimalizowanej funkcji strat:

$$V_t(\theta) = \sum_{s=1}^t \lambda^{t-s} e^2(s). \quad (3.22)$$

Wartość współczynnika zapominania jest liczbą mniejszą od 1 i przyjmuje zazwyczaj wartości wyższe niż 0,9 np. 0,97 lub 0,99. Jak sama nazwa wskazuje, stosując λ , wcześniejsze pomiary wraz ze wzrostem t stają się coraz mniej ważne. Innymi słowy, im mniejsza jest wartość współczynnika zapominania, tym szybciej „zapominamy” o wcześniejszych informacjach.

Algorytm EF-RLS uwzględniający współczynnik zapominania składa się z następujących równań:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + k(t)e(t), \quad (3.23)$$

$$e(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1), \quad (3.24)$$

$$\mathbf{k}(t) = \mathbf{P}(t)\boldsymbol{\varphi}(t) = \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{\lambda + \boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}, \quad (3.25)$$

$$\mathbf{P}(t) = \frac{1}{\lambda} \left[\frac{\mathbf{P}(t-1) - \mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)}{\lambda + \boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} \right]. \quad (3.26)$$

Warto zauważyć, że jeżeli przyjmiemy $\lambda = 1$, to uzyskamy zbiór równań odpowiadający standardowej metodzie RLS, czyli równaniom (3.17), (3.19), (3.20) i (3.21) [4].

3.4. Identyfikowalność

Identyfikowalność oznacza możliwość wykonania procesu identyfikacji. W rozważanych modelach ARX warunkiem identyfikalności jest:

$$\det[\mathbf{R}(N)] = \det \left[\sum_{t=1}^N \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t) \right] \neq 0. \quad (3.27)$$

Innymi słowy, warunek ten nie jest spełniony, gdy elementy wektora regresji $\boldsymbol{\varphi}(t)$ są liniowo zależne od siebie.

Identyfikowalność nie jest własnością procesu, lecz skutkiem złego uwarunkowania eksperymentu identyfikacji. Oznacza to, że zależy ono od konkretnej postaci modelu i wektora regresji $\boldsymbol{\varphi}(t)$. Istnieją dwa główne źródła nieidentyfikalności:

- sygnał pobudzający zbyt niskiego rzędu – im słabiej pobudzony zostanie układ, tym mniej dokładnie można go zidentyfikować a w skrajnym przypadku może dojść do nieidentyfikalności,
- istnienie liniowego sprzężenia zwrotnego pomiędzy sygnałem wejściowym a sygnałem wyjściowym.

Przywrócenie identyfikowalności jest możliwe poprzez zastosowanie sygnału wejściowego wysokiego rzędu np. PRBS (ang. Pseudo Random Binary Sequence), dodając szum biały do sygnału wyjściowego lub przełączając pomiędzy kilkoma sygnałami sterującymi [4].

3.5. Weryfikacja modelu

Po wykonaniu eksperymentu identyfikacyjnego należy ocenić czy otrzymany model dobrze odwzorowuje dynamikę obiektu rzeczywistego. Powinien też być dostatecznie elastyczny oraz możliwie jak najprostszy.

Na samym początku zaleca się sprawdzić, czy wyjście naszego modelu jest zbliżone do wartości wyjściowej obiektu. W tym celu należy wyznaczyć wartość wyjściową modelu korzystając z tego samego sygnału wejściowego, jakim pobudzono obiekt rzeczywisty na drodze eksperymentu identyfikacji. W praktyce wyjście obiektu zawsze zawiera szum pomiarowy, więc ważne jest to, aby wiedzieć, że sygnały te muszą się nieznacznie od siebie różnić, ponieważ model opisuje tylko tę część sygnału wyjściowego, która pochodzi od sygnału wejściowego.

Dobór struktury modelu jest jednym z najważniejszych etapów identyfikacji. Jeżeli model będzie przeparametryzowany, może to doprowadzić do niepotrzebnych problemów obliczeniowych przy estymacji parametrów oraz implementacji sterownika. Z kolei, jeśli mamy

do czynienia z niedoparametryzowaniem, otrzymamy mało dokładny model, który pogorszy jakość regulacji.

W praktyce często dokonujemy identyfikacji modeli o wzrastających rzędach i wybieramy jeden, najbardziej pasujący do dynamiki obiektu rzeczywistego. Decyzja o wyborze konkretnego modelu może być poprzedzona różnego rodzaju testami. Zazwyczaj pomagają one podczas ustalania struktury modelu, jednak wskazane jest sporządzanie wykresów i posługiwanie się zdrowym rozsądkiem [4].

3.6. Kryterium informacyjne Akaike'go

Kryterium informacyjne Akaike'go (AIC) jest jedną z metod doboru rzędu modelu, która bazuje na ocenie wartości funkcji strat (lub funkcji kosztów). Oblicza się ją następująco:

$$V_N(\hat{\theta}) = \frac{1}{N} \sum_{t=1}^N [y(t) - \varphi^T(t)\hat{\theta}(N)]^2, \quad (3.28)$$

gdzie N jest liczbą próbek branych pod uwagę podczas procesu walidacji. Jeżeli do obliczania funkcji strat wykorzystamy dane użyte w identyfikacji, wraz ze zwiększaniem się rzędu modelu jej wartość spada, jednak po osiągnięciu najbardziej pasującego modelu ta różnica nie jest już tak duża. Dane niewykorzystane do identyfikacji ale pochodzące z eksperymentu przeprowadzanego w takich samych warunkach, spowodują, że wartości funkcji $V_N(\hat{\theta})$ zaczynają wzrastać dla modeli przeparametryzowanych. Kryterium AIC zdefiniowane jest poniższym wzorem:

$$AIC(p) = N \log[V_N(\hat{\theta})] + 2p, \quad (3.29)$$

gdzie p jest liczbą estymowanych parametrów modelu. Zaleca się wybrać model, dla którego wartość $AIC(p)$ jest najmniejsza. Gwarantuje to uniknięcie wybrania zbyt skomplikowanego modelu, zapewniając tym samym dobre odtworzenie dynamiki obiektu [4].

4. STEROWANIE PREDYKCYJNE

Pierwsze sterowniki oparte na technologii predykcyjnej MPC (ang. Model Predictive Control) powstały w latach siedemdziesiątych dwudziestego wieku i od tamtego czasu intensywnie się rozwijały. Termin Model Predictive Control obejmuje bardzo szeroką gamę metod sterowania predykcyjnego, które posiadają następujące, wspólne cechy:

- stosowanie modelu do predykcji przyszłych odpowiedzi obiektu, określonych na horyzoncie predykcji,
- wyznaczenie wektora sterowań, który minimalizuje funkcję kosztu,
- stosowanie przesuwnego horyzontu, w każdej iteracji algorytmu wyznaczany jest wektor sterowań określony na horyzoncie sterowania, natomiast implementowana jest tylko pierwsza jego wartość.

Sterowniki predykcyjne posiadają wiele zalet w stosunku do tradycyjnych metod sterowania:

- mogą być użyte do regulacji obiektów o bardzo prostej dynamice jak i tych o skomplikowanej dynamice, wliczając w to obiekty nieminimalnofazowe, niestabilne oraz z dużym opóźnieniem transportowym,
- są łatwe w obsłudze, intuicyjne, wystarczy podstawowa wiedza, aby dobrze dostroić regulator,
- kompensują czas martwy obiektu,
- łatwo można rozszerzyć koncepcję sterownika na przypadek wielowymiarowy,
- należą do kategorii sterowania optymalnego, tzn. posiadają kryterium jakości,
- na etapie projektowania regulatora można nałożyć ograniczenia (zarówno na wektor sterowań jak i wektor przyrostów sterowań) [5].

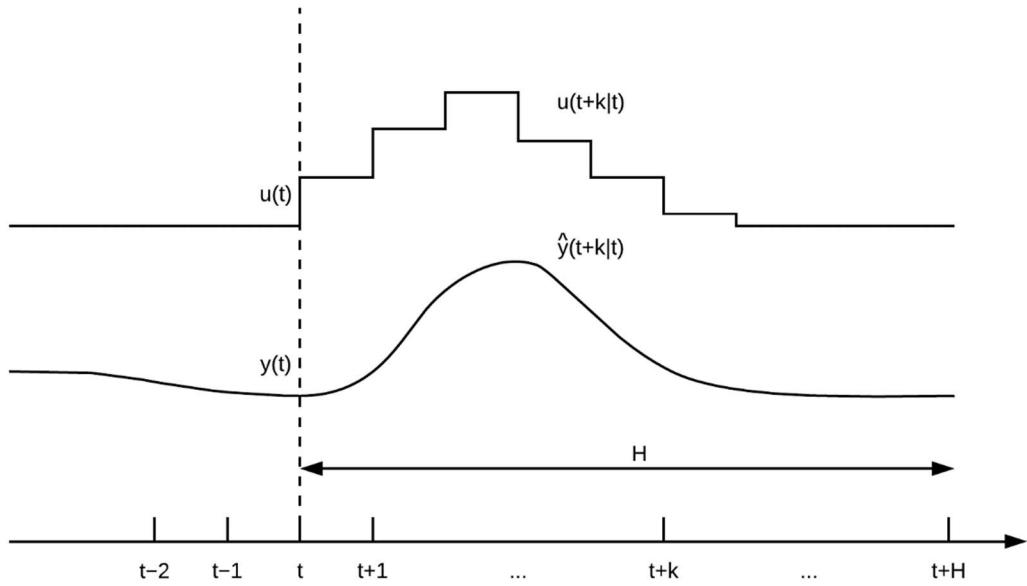
Bardzo szybko znalazły one szerokie zastosowanie m. in. w przemyśle: chemicznym, samochodowym, rafineryjnym, papierniczym, gazowym, kosmicznym, metalurgicznym, farmaceutycznym a także w: elektrowniach, cukrowniach, przetwórstwie spożywczym, górnictwie, hutnictwie, procesach destylacji, mikroelektronice, biotechnologii, polimeryzacji oraz procesach oczyszczania ścieków [8].

4.1. Idea sterowania predykcyjnego

Przyszłe wartości wyjścia określone na horyzoncie H , zwanym horyzontem predykcji, szacowane są w każdym kolejnym kroku iteracji na podstawie modelu procesu. Wyrażenie $\hat{y}(t + k|t)$ oznacza predykcję wyjścia na chwilę $t + k$ w chwili t . Dzieje się to dla wszystkich $k = 0 \dots H - 1$ określonych na horyzoncie predykcji (rys. 4.1).

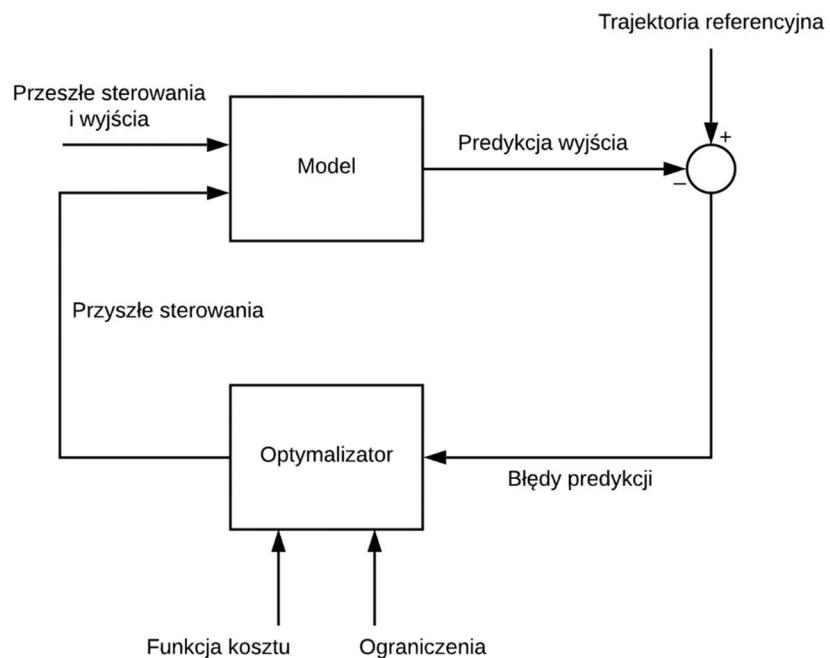
Przyszłe wartości sterowań $u(t + k)$ wyznaczane są poprzez rozwiązanie zadania optymalizującego kryterium jakości w taki sposób, aby predykowane wartości wyjścia były jak najbardziej zbliżone do zadanej trajektorii referencyjnej. Kryterium to najczęściej przyjmuje formę kwadratową i uwzględnia także ograniczenia nałożone na wartości wektora przyrostów sterowań lub wartości wektora sterowań.

Po wyznaczeniu wektora przyszłych sterowań należy zaimplementować pierwszą wartość sygnału sterującego $u(t|t)$ a następnie powtórzyć wykonanie pierwszego kroku.



Rys. 4.1. Idea sterowania predykcyjnego [5]

Proces implementacji sterowników z rodziny MPC ilustruje schemat blokowy z rys. 4.2. Predykcja wyjścia dokonywana jest na podstawie modelu procesu, który korzysta z przeszłych wartości sterowań i przeszłych wartości wyjść oraz przyszłych wartości sterowań.



Rys. 4.2. Struktura sterownika predykcyjnego [5]

Model jest bardzo ważnym czynnikiem, ponieważ od niego zależy jak dokładnie odwzorowana będzie dynamika procesu oraz jak łatwo będzie dało się zaimplementować

sterownik. To właśnie od modelu procesu zależą różne metodologie sterowania predykcyjnego, które zostały przedstawione w następnym podrozdziale. Poniższy schemat przedstawia sytuację, w której parametry modelu procesu są znane, tzn. posiadamy pełną wiedzę o procesie. W przypadku, gdy proces jest nieznany, trzeba dokonać estymacji parametrów modelu metodami identyfikacyjnymi. Jeżeli mamy do czynienia z procesem niestacjonarnym, najlepszym rozwiązaniem będzie implementacja adaptacyjnego sterownika predykcyjnego.

Drugim bardzo istotnym czynnikiem jest optymalizator, który na podstawie błędów predykcji, funkcji kosztu oraz ograniczeń wyznacza ciąg przyszłych sterowań. Stopień skomplikowania optymalizacji zależy ściśle od horyzontu predykcji, ilości zmiennych oraz istnienia ograniczeń. Rozwiążanie problemu sterowania predykcyjnego bez ograniczeń zużyje mniej zasobów obliczeniowych oraz skróci czas trwania algorytmu w porównaniu do tego samego zadania z uwzględnieniem ograniczeń [5].

4.2. Rodzaje sterowników predykcyjnych

Istnieje wiele metod projektowania sterowników opartych na koncepcji MPC. Wybrano kilka najpopularniejszych wersji stosowanych w przemyśle i scharakteryzowano ich podstawowe właściwości. Najwięcej uwagi poświęcono sterownikowi GPC, który opisano szerzej w kolejnym podrozdziale.

4.2.1. Dynamic Matrix Control

Sterownik Dynamic Matrix Control (DMC) jest jednym z pierwszych algorytmów należących do rodziny MPC. W 1979 roku zaprezentowane zostały szczegóły tego algorytmu działającego na zmiennych wielowymiarowych bez ograniczeń. Algorytm ten bazuje na odpowiedzi skokowej modelu procesu i jest implementowany dla obiektów liniowych w pętli otwartej. Stosowany jest z powodzeniem w przemyśle petrochemicznym, gdzie wymagana jest duża liczba zmiennych [6, 7].

4.2.2. Model Algorithmic Control

W 1978 roku Richalet zaproponował swoje rozwiązanie, które początkowo nazywano Model Predictive Heuristic Control (MPHC). Nazwa szybko przekształciła się w Model Algorithmic Control (MAC). Metoda projektowania regulatora MAC jest bardzo zbliżona do regulatora DMC, jednak posiada następujące różnice:

- zamiast odpowiedzi skokowej modelu procesu, używa się jego odpowiedzi impulsowej,
- zamiast wektora przyrostów sterowań Δu , stosuje się wektor sterowań u ,
- wektor estymowanych wartości szumu poddawany jest filtracji.

Sterowniki MAC najczęściej stosuje się w: elektrowniach, piecach szklanych, generatorach pary i procesach destylacji ropy naftowej [6, 7].

4.2.3. Predictive Functional Control

Algorytm Predictive Functional Control (PFC) dzięki swojej elastyczności i szybkiej zbieżności do wartości zadanej nadaje się do sterowania procesami o dużej dynamice. Można użyć w nim dowolnego modelu procesu, jednak najczęściej stosowanym jest model zmiennych stanu, który gwarantuje właściwość sterowania odpornego. Wadą tego rozwiązania jest to, że dla procesów niestacjonarnych oraz z nieznanym opóźnieniem transportowym, uzyskanie dokładnego modelu matematycznego w postaci zmiennych stanu jest skomplikowanym zadaniem [6].

4.2.4. Extended Prediction Self-Adaptive Control

Implementacja sterownika Extended Prediction Self-Adaptive Control (EPSAC) wymaga użycia modelu transmitancyjnego z uwzględnieniem modelu zakłóceń. Struktura algorytmu jest prosta, ponieważ założono w niej, że sygnał sterujący po chwili t jest stały, tzn. przyrost sygnału sterującego jest zerowy a horyzont predykcji wynosi 1. W rezultacie wystarczy policzyć wartość skalarną $u(t)$, którą można wyznaczyć analitycznie [5].

4.2.5. Extended Horizon Adaptive Control

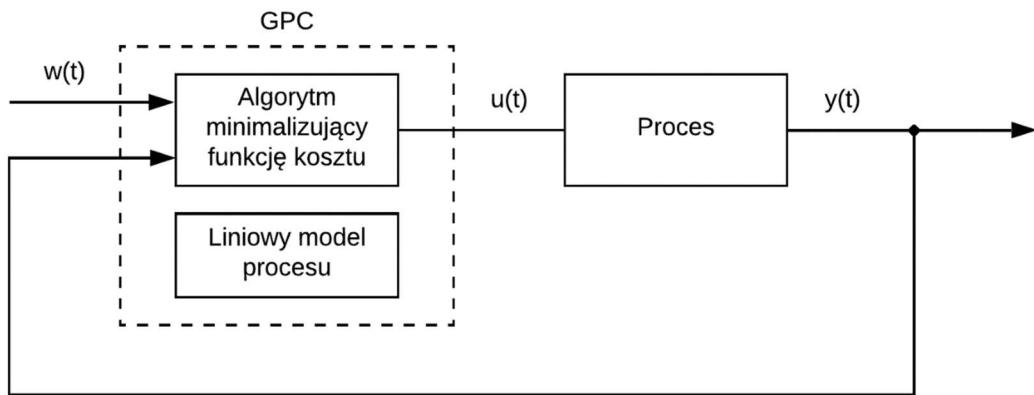
Formuła Extended Horizon Adaptive Control (EHAC) określa model transmitancyjny bez uwzględnienia modelu zakłóceń. Głównym zadaniem algorytmu jest wyznaczenie wektora sterowań, określonego na horyzoncie predykcji H , minimalizującego różnicę pomiędzy przewidywanym wyjściem modelu procesu a trajektorią referencyjną tylko w chwili $t + H$. Właściwy dobór horyzontu predykcji pozwala na sterowanie obiektami nieminimalnofazowymi [5].

4.3. Generalized Predictive Control

Sterownik GPC jest jednym z najpopularniejszych rozwiązań z grupy MPC, stosowany zarówno w przemyśle jak i w środowisku akademickim. Charakteryzuje się dużą wydajnością oraz odpornością.

Uogólniony sterownik predykcyjny ma wiele wspólnych cech z pozostałymi odmianami sterowników z rodziny MPC. Może zostać zaimplementowany z powodzeniem dla obiektów niestabilnych, nieminimalnofazowych oraz ze zmiennym opóźnieniem transportowym.

Algorytm minimalizujący funkcję kosztu wyznacza ciąg sterowań $u(t)$ na podstawie modelu obiektu, zmierzonego wyjścia $y(t)$ i trajektorii referencyjnej $w(t)$. Pomiędzy kolejnymi próbkami, algorytm sterownika dokonuje jednokrokowej predykcji sygnału wyjściowego obiektu regulacji (rekursywnie – tyle razy ile wynosi horyzont predykcji) a następnie, korzystając z liniowego modelu procesu, wyznacza przyszłą próbkę sygnału sterującego. Uproszczony schemat tej operacji został przedstawiony poniżej na rys. 4.3 [5, 6, 7].



Rys. 4.3. Podstawowa struktura sterownika GPC [9]

4.3.1. Model

Najczęściej stosowanym modelem w sterowniku GPC jest model CARIMA (ang. Controlled Autoregressive Integrated Moving Average):

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t) + C(z^{-1})\frac{\eta(t)}{\Delta}, \quad (4.1)$$

$$\Delta = 1 - z^{-1}, \quad (4.2)$$

gdzie A, B i C są wielomianami operatora przesunięcia z^{-1} :

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na}, \quad (4.3)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}, \quad (4.4)$$

$$C(z^{-1}) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc} \quad (4.5)$$

oraz d stanowi opóźnienie transportowe obiektu, $y(t)$ jest wyjściem obiektu, $u(t)$ to sygnał sterujący obiektem, $\eta(t)$ oznacza szum biały o zerowej średniej.

Często równanie (4.1) można zapisać w sposób następujący, mnożąc oby dwie strony równania przez Δ :

$$\Delta A(z^{-1})y(t) = z^{-d}B(z^{-1})\Delta u(t) + C(z^{-1})\eta(t). \quad (4.6)$$

Obecność operatora Δ zapewnia właściwości całkujące regulatora GPC, dzięki czemu możliwe jest modelowanie zakłóceń niestacjonarnych [5].

Założenie, że dyskretny czas opóźnienia $d = 1$ sprawi, że algorytm sterownika GPC w wersji adaptacyjnej będzie mniej wrażliwy na zmiany czasu opóźnienia [10].

4.3.2. Funkcja kosztu

Algorytm GPC wyznacza wektor sterowania, który minimalizuje funkcję kosztu podaną równaniem (4.7). Kryterium jakości J odnosi się do modelu CARIMA z równania (4.6). Założono w modelu, że dyskretny czas opóźnienia d wynosi 1.

$$J(H, L) = E \left\{ \sum_{j=1}^H [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^L \varrho(j)[\Delta u(t+j-1)]^2 \right\}, \quad (4.7)$$

gdzie $\hat{y}(t+j|t)$ jest optymalną j -krokową predykcją wyjścia obiektu na chwilę t , H i L są odpowiednio horyzontem predykci i horyzontem sterowania, $\varrho(j)$ to współczynnik wagowy, natomiast $w(t+j)$ to przyszła trajektoria referencyjna.

Celem regulacji jest wyznaczenie takiego ciągu sterowań $u(t), u(t+1) \dots u(t+L)$, z uwzględnieniem wagi na przyrost sterowania, aby przyszłe (predykowane) wartości wyjściowe obiektu $\hat{y}(t+j)$ były możliwie jak najbliżej trajektorii referencyjnej $w(t+j)$ [5].

4.3.3. Algorytm sterowania

Algorytm minimalizujący kwadratową funkcję kosztu (5.7) podany jest następującym wyrażeniem:

$$u(t) = u(t-1) + \mathbf{q}^T [\mathbf{w} - \hat{\mathbf{y}}], \quad (4.8)$$

w którym:

$$\mathbf{q}^T = [q_1, q_2, \dots, q_H] = \mathbf{i}^T [\mathbf{Q}^T \mathbf{Q} + \varrho \mathbf{I}]^{-1} \mathbf{Q}^T, \quad (4.9)$$

$$\mathbf{i}^T = [1, 0, 0, \dots, 0], \quad (4.10)$$

$$\mathbf{Q} = \begin{bmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{L-1} & h_{L-2} & \dots & h_0 \\ \vdots & \vdots & & \vdots \\ h_{H-1} & h_{H-2} & \dots & h_{H-L} \end{bmatrix}, \quad (4.11)$$

$$\mathbf{w} = [w(t+1), w(t+2), \dots, w(t+H)]^T, \quad (4.12)$$

$$\hat{\mathbf{y}} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+H)]^T, \quad (4.13)$$

h_j – wartości odpowiedzi skokowej procesu $B(z^{-1})/A(z^{-1})$,

\mathbf{w} – wektor przyszłych wartości trajektorii referencyjnej,

$\hat{\mathbf{y}}$ – wektor optymalnych w sensie średniokwadratowym, wykonanych w chwili t predykci wielkości wyjściowej obiektu regulacji przy założeniu zerowych przyrostów sygnału sterującego tj. przy założeniu, że sygnał sterujący jest stale równy $u(t-1)$.

Bardzo dużą zaletą tego algorytmu jest brak konieczności rozwiązywania równań diofantycznych, które przy dużych horyzontach predykci sprawiają, że wymagana jest o wiele większa moc obliczeniowa oraz wydłużony czas pojedynczej iteracji programu.

Zwiększyły horyzont predykci H (lub/i horyzont sterowania L) powoduje zwiększenie wymiarów macierzy \mathbf{Q} . W praktyce warto przyjąć horyzont sterowania równy horyzontowi predykci (nie może być on większy od horyzontu predykci). Wtedy macierz \mathbf{Q} jest macierzą kwadratową i ma postać odwracalną, co znacznie ułatwia proces implementacji sterownika [10].

4.3.4. Dostrajanie regulatora

Sterownik predykcyjny posiada kilka charakterystycznych dla niego parametrów, które wpływają na jakość regulacji. Do takich czynników należą: horyzont predykcji, horyzont sterowania, czas próbkowania oraz współczynnik wagowy przyrostów sterowania.

Dla systemów dyskretnych, których odpowiedź skokowa jest ustalona, horyzont predykcji można wyznaczyć w następujący sposób:

$$H = \frac{\tau}{T_s}, \quad (4.14)$$

gdzie:

H – horyzont predykcji,

τ – czas ustalania się odpowiedzi skokowej obiektu,

T_s – okres próbkowania.

W praktyce, często postępuje się inaczej. Rozpoczęcie eksperymentu strojenia regulatora powinno się rozpoczynać od wybrania możliwie jak największego horyzontu predykcji i horyzontu sterowania. Następnie, znając dynamikę obiektu, należy dokonać wyboru okresu próbkowania biorąc pod uwagę dobrą jakość sterowania oraz możliwość implementacji algorytmu. Na samym końcu, dla wyznaczonej wartości okresu próbkowania, dobrać wartości horyzontów w taki sposób, aby zapewnić najlepszą jakość regulacji jak najmniejszym kosztem obliczeniowym [11].

Powyższe równanie (4.14) można interpretować tak, że model w każdej kolejnej iteracji algorytmu, musi znać zachowanie obiektu aż do momentu ustalenia się jego odpowiedzi skokowej. Zbyt mały horyzont predykcji spowoduje oscylacje wokół wartości referencyjnej oraz wydłuży czas osiągnięcia wartości zadanej. Należy również pamiętać, że przyjęcie zbyt dużej jego wartości również nie jest wskazane z uwagi na zwiększenie złożoności obliczeniowej algorytmu sterownika [2].

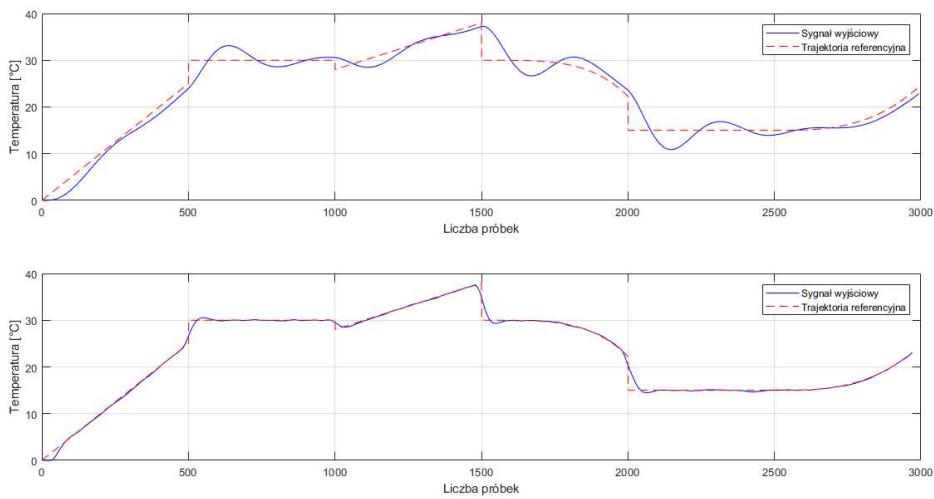
Współczynnik wagowy ϱ , pojawiający się w (4.7) oraz (4.9), wpływa na szybkość zmian sygnału sterującego oraz jego wariancję. Im większa jest jego wartość, regulacja jest bardziej płynna. Oznacza to, że odpowiedź obiektu zostanie osiągnięta w dłuższym wymiarze czasu, stosując łagodne sterowanie. Z kolei jeśli współczynnik ϱ będzie mały, odpowiedź obiektu szybko zbiegnie do wartości zadanej ale sygnał sterujący będzie posiadał dużą wariancję [10].

Przeprowadzono symulacje sygnałów: wejściowych i wyjściowych dla modelu CARIMA o wielomianach:

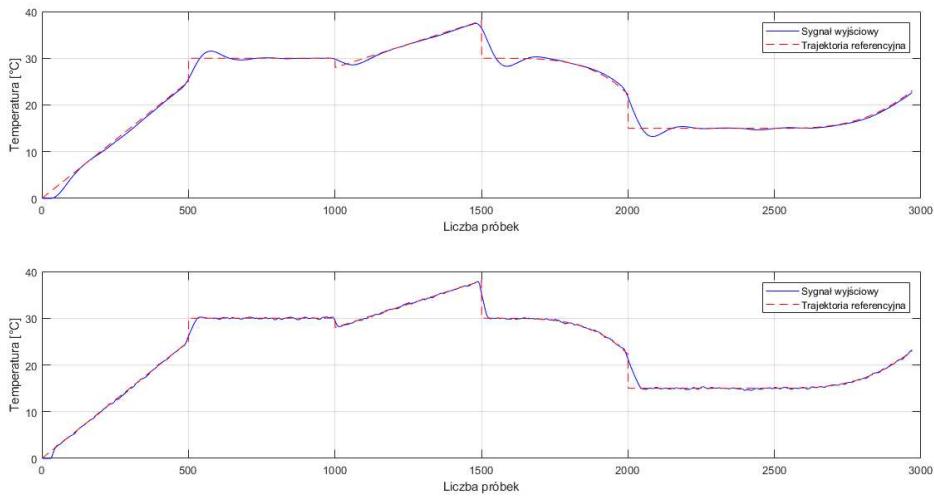
$$A(z^{-1}) = 1 - 1,37z^{-1} + 0,375z^{-2}, \quad (4.15)$$

$$B(z^{-1}) = 0,0025 \quad (4.16)$$

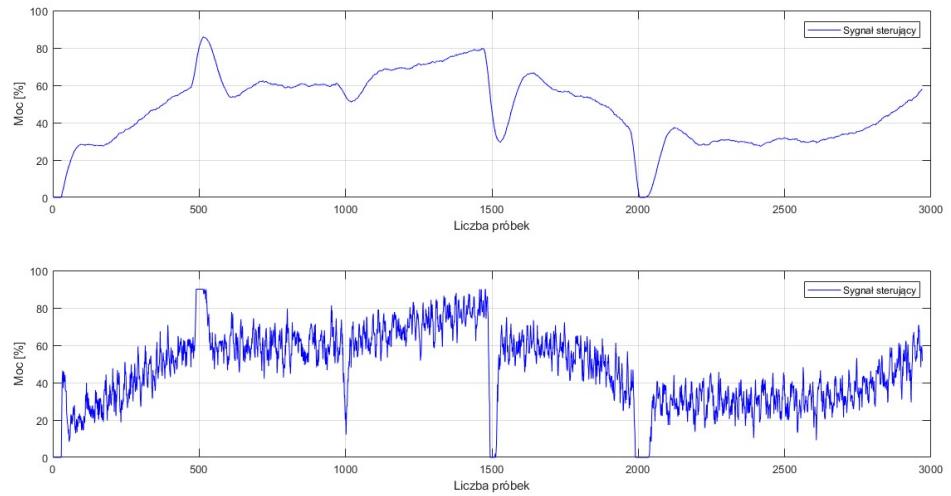
i opóźnieniu transportowym $d = 1$. W modelu uwzględniono szum o wariancji $\sigma_\eta = 0,25$. Symulacje przedstawiają sygnały wyjściowe modelu w zależności od nastaw: horyzontu predykcji (rys. 4.4) oraz współczynnika ϱ (rys. 4.5). Dodatkowo zaprezentowano, wyznaczone przez algorytm sterownika, sygnały sterujące dla różnych wartości parametru ϱ (rys. 4.6).



Rys. 4.4. Symulacja sygnałów wyjściowych modelu dla algorytmu sterownika o parametrach:
 $H = 7$ (górnny wykres) i $H = 30$ (dolny wykres). Współczynnik $\varrho = 1$.



Rys. 4.5. Symulacja sygnałów wyjściowych modelu dla algorytmu sterownika o parametrach:
 $\varrho = 5$ (górnny wykres) i $\varrho = 0.01$ (dolny wykres). Horyzont predykcji $H = 30$.



Rys. 4.6. Symulacja sygnałów sterujących modelu dla algorytmu sterownika o parametrach: $\varrho = 5$ (górnny wykres) i $\varrho = 0.01$ (dolny wykres). Horyzont predykcji $H = 30$.

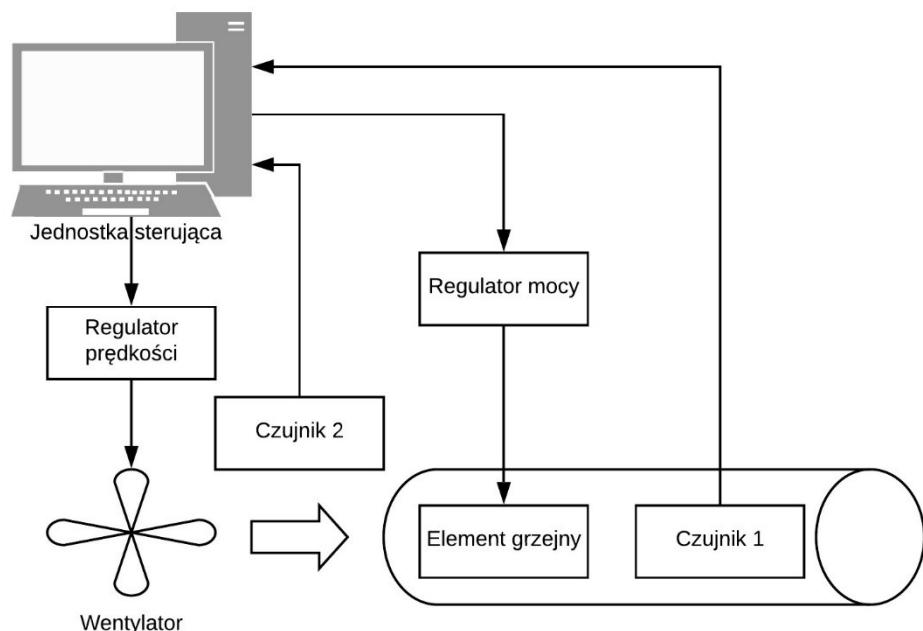
5. BUDOWA STANOWISKA

Budowa stanowiska jest pierwszym zagadnieniem praktycznym niniejszej pracy. Na samym początku określono założenia projektowe, które pozwoliły wyznaczać kolejne cele konstrukcyjne. W późniejszej części rozdziału zostały przedstawione i szerzej omówione poszczególne elementy całego systemu.

5.1. Założenia projektowe

Główym celem projektowym jest budowa systemu, na którym istnieje możliwość implementacji algorytmu sterowania temperaturą powietrza oraz jego uruchomienie i zebranie danych pomiarowych.

Obiektem regulacji jest nagrzewnica powietrza, składająca się z: kanału, w którym odbywa się transport ogrzewanego powietrza, wentylatora, regulatora mocy i czujnika temperatury. Zadaniem wentylatora jest wymuszenie transportu powietrza skierowanego w stronę elementu grzejnego. Na końcu kanału znajduje się czujnik 1, który dokonuje pomiaru wartości wyjściowych obiektu regulacji. Czujnik 2 służy do pomiaru aktualnej temperatury powietrza. Informacje z czujników temperatury przesyłane są do jednostki sterującej. Schemat całego stanowiska został przedstawiony na rys. 5.1.



Rys. 5.1. Schemat stanowiska

W jednostce sterującej zaimplementowano algorytm sterownika klimatyzacji, który na podstawie różnicy pomiędzy aktualnymi pomiarami temperatury z obu czujników, będzie sterować mocą elementu grzejnego. Wyeliminuje to wpływ zmieniającej się temperatury otoczenia na wyniki przeprowadzanych eksperymentów.

Sterowanie prędkością wentylatora powoduje zmianę opóźnienia transportowego obiektu. Wymuszenie zmiany parametrów obiektu jest bardzo istotne, ponieważ umożliwia sprawdzenie działania algorytmu sterownika predykcyjnego w wersji adaptacyjnej.

Podstawowym założeniem przy budowie stanowiska jest jego prostota. Skonstruowany obiekt ma spełniać wszystkie powyższe funkcjonalności i być możliwie jak najprostszy, ponieważ przynosi to duże oszczędności finansowe. Ważne jest również to, aby stanowisko było bezpieczne. Sterowanie elementami grzejnymi jest bardzo niebezpieczne z uwagi na osiąganą przez nie wysoką temperaturę oraz duży pobór prądu.

5.2. Nagrzewnica powietrza

Nagrzewnica jest jednym z najważniejszych elementów stanowiska, to ona ma decydujący wpływ na charakterystykę obiektu sterowania.

Krytycznym etapem okazał się wybór elementu grzejnego, którego zadaniem jest ogrzewanie transportowanego powietrza. Zdecydowano się na grzałkę ceramiczną o niskiej mocy, która wykorzystywana jest do druku 3D (rys. 5.2). Jej parametry techniczne zostały przedstawione w tabeli 5.1.



Rys. 5.2. Grzałka ceramiczna

Tabela 5.1. Parametry techniczne grzałki ceramicznej

Parametr	Wartość
Wymiary (średnica/długość)	6 mm/20 mm
Napięcie zasilania	12 V
Moc znamionowa	50 W
Temperatura maksymalna	350 °C
Długość przewodu	1 m
Materiał	Stal nierdzewna

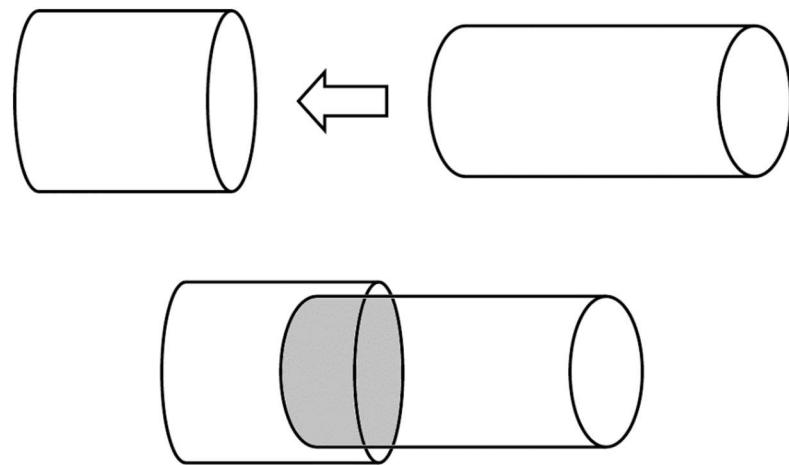
Grzałka osiąga temperaturę maksymalną w zupełności wystarczającą, aby podgrzać strumień powietrza o nawet 30 °C. Jedyną wadą jest jej mały rozmiar. Zbyt mała powierzchnia grzejna może spowodować trudności z ogrzaniem transportowanego powietrza.

Idealnym rozwiązaniem okazuje się użycie bloku grzejnego, do którego można przymocować grzałkę, zwiększając tym samym jej powierzchnię grzejną. Na rys. 5.3 został zaprezentowany blok grzejny, który wykorzystano do budowy stanowiska.



Rys. 5.3. Blok grzejny E3D V6 o wymiarach 20x16x12 mm

Grzałka ceramiczna wraz z blokiem grzejnym została umieszczona w rurze wykonanej z polipropylenu. Na jej końcu umieszczono również termometr do pomiaru temperatury wyjściowej obiektu. O ile z montażem termometru nie wiąże się żadne ryzyko, to nad mocowaniem elementu grzejnego w rurze z tworzywa sztucznego należy się zastanowić. Polipropylen bez problemu wytrzyma ciągły przepływ cieczy bądź powietrza w temperaturze około 100 °C. Kanał transmisyjny tworzą dwa elementy o różnej średnicy, włożone jeden w drugi, aby zwiększyć grubość ścianki kanału tam, gdzie znajduje się grzałka. Miejsce, w którym dwie części nachodzą się na siebie, zaznaczono kolorem szarym (rys. 5.4).



Rys. 5.4. Schemat kanału do przepływu cieplego powietrza.

Na wlocie kanału zamontowano wentylator (rys. 5.5) oraz czujnik do pomiaru temperatury powietrza nawiewanego. Dane techniczne wentylatora zamieszczone w tabeli 5.2.

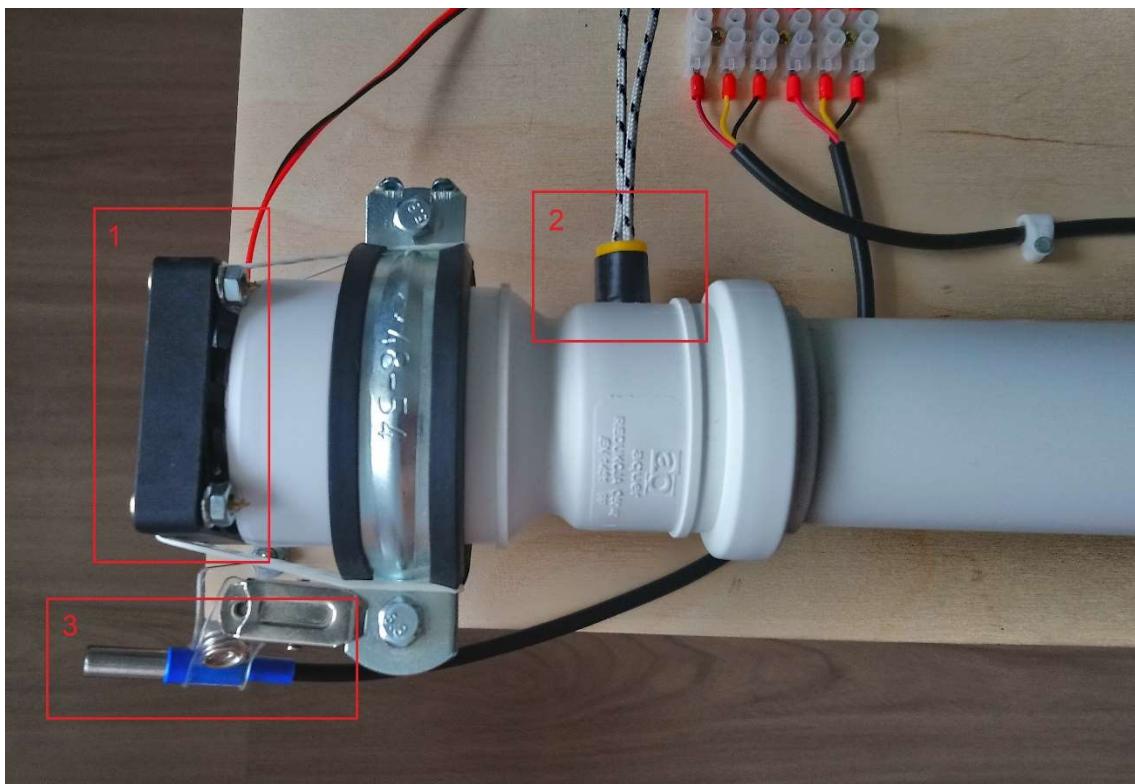


Rys. 5.5. Wentylator

Tabela 5.2. Dane techniczne wentylatora

Parametr	Wartość
Wymiary	50x50x10 mm
Napięcie zasilania	12 V
Maksymalna prędkość obrotowa	7500 obr./min
Przepływ powietrza	26,28 m ³ /h
Natężenie dźwięku	40 dB

Montaż nagrzewnicy rozpoczęto od wywiercenia otworów w płycie oraz zainstalowania metalowych obejm, w których zaciśnięto kanał. Następnie zrobiono dwa otwory w rurze do montażu grzałki i czujnika temperatury. Na grzałkę nałożono blok grzejny a jej przewód w okolicach elementu grzejnego owinięto bawełnianą izolacją termiczną i całość zaklejono taśmą naprawczą o podwyższonej odporności na wysoką temperaturę. Dzięki temu zabiegowi, kanał w miejscu, w którym znajduje się grzałka, nie ulegnie nadtopieniu. Zdjęcie fragmentu nagrzewnicy, zawierającego wszystkie wymienione wyżej elementy, przedstawiono na rys. 5.6. Na rys. 5.7 zaprezentowano sposób montażu czujnika temperatury.



Rys. 5.6. Nagrzewnica powietrza: 1 – wentylator, 2 – mocowanie grzałki, 3 – czujnik temperatury



Rys. 5.7. Mocowanie czujnika temperatury

5.3. Zasilanie układu

Nagrzewnica powietrza potrzebuje swojego zewnętrznego źródła zasilania, ponieważ element grzejny pobiera zbyt dużo mocy, aby zasilano go bezpośrednio z jednostki sterującej. Zdecydowano się na zakup zasilacza modułowego z zabezpieczeniami przeciążeniowymi i przeciwwspięciowymi (rys. 5.8). Szczegółowe dane techniczne zasilacza przedstawiono w tabeli 5.3.

Tabela 5.3. Dane techniczne zasilacza modułowego

Parametr	Wartość
Wymiary	36x80x110 mm
Napięcie wejściowe	110/220 V ±15%
Napięcie wyjściowe	12 V
Regulacja napięcia	±0,5 V
Moc znamionowa	50 W
Stopień ochrony	IP20

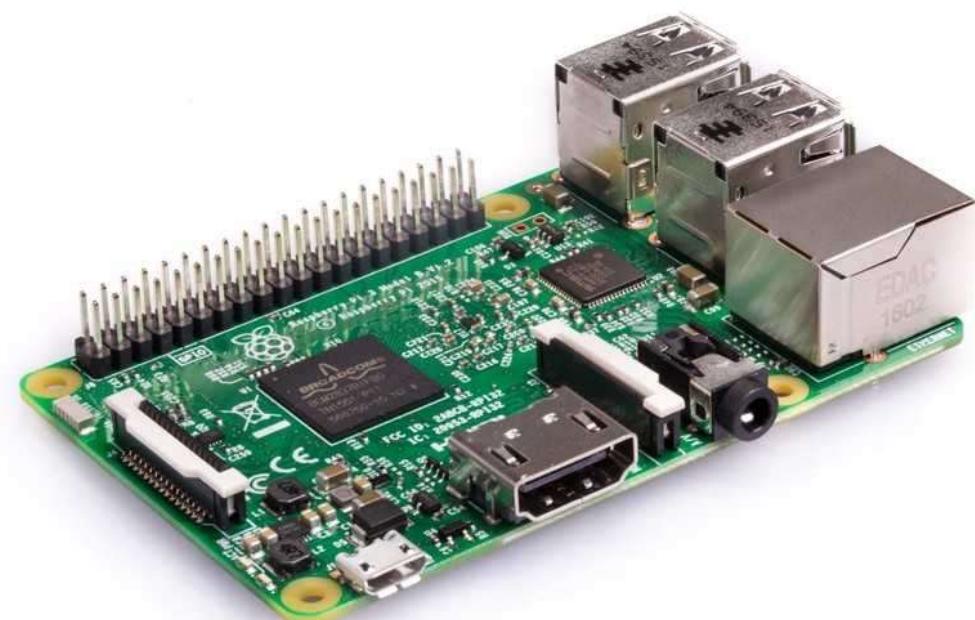


Rys. 5.8. Zasilacz modułowy

Ten model zasilacza posiada wbudowany regulator napięcia zasilania. Jego wartość można zmieniać za pomocą potencjometru obrotowego, znajdującego się obok listwy zaciskowej. Jednak różnica pomiędzy poziomami regulacji jest zbyt duża i nie pozwoli ona na bardzo płynne sterowanie mocą grzałki. Do tego celu wykorzystano specjalny moduł sterownika, oparty na metodzie PWM (ang. Pulse Width Modulation), który opisano szerzej w dalszej części rozdziału.

5.4 Jednostka sterująca

Wybór jednostki sterującej stanowi bardzo ważny aspekt w kontekście pracy całego systemu kontrolno–pomiarowego. Determinuje on odpowiedni dobór klasy (analogowe, cyfrowe) innych podzespołów, w tym przypadku czujników temperatury. Na potrzeby projektu wybrano komputer jednoplatformowy Raspberry Pi 3 Model B (rys. 5.9), kierując się jego niską ceną, dobrymi parametrami technicznymi, kompatybilnością z wieloma modułami rozszerzającymi oraz stosunkowo małymi wymiarami.



Rys. 5.9. Komputer Raspberry Pi 3 Model B

Wyposażono go w czterordzeniowy procesor ARM-8 Cortex-A53 z częstotliwością taktowania 1,2 GHz oraz 1 GB pamięci RAM. Cały układ scalony (ang. SoC – System on a Chip) posiada wiele urządzeń peryferyjnych:

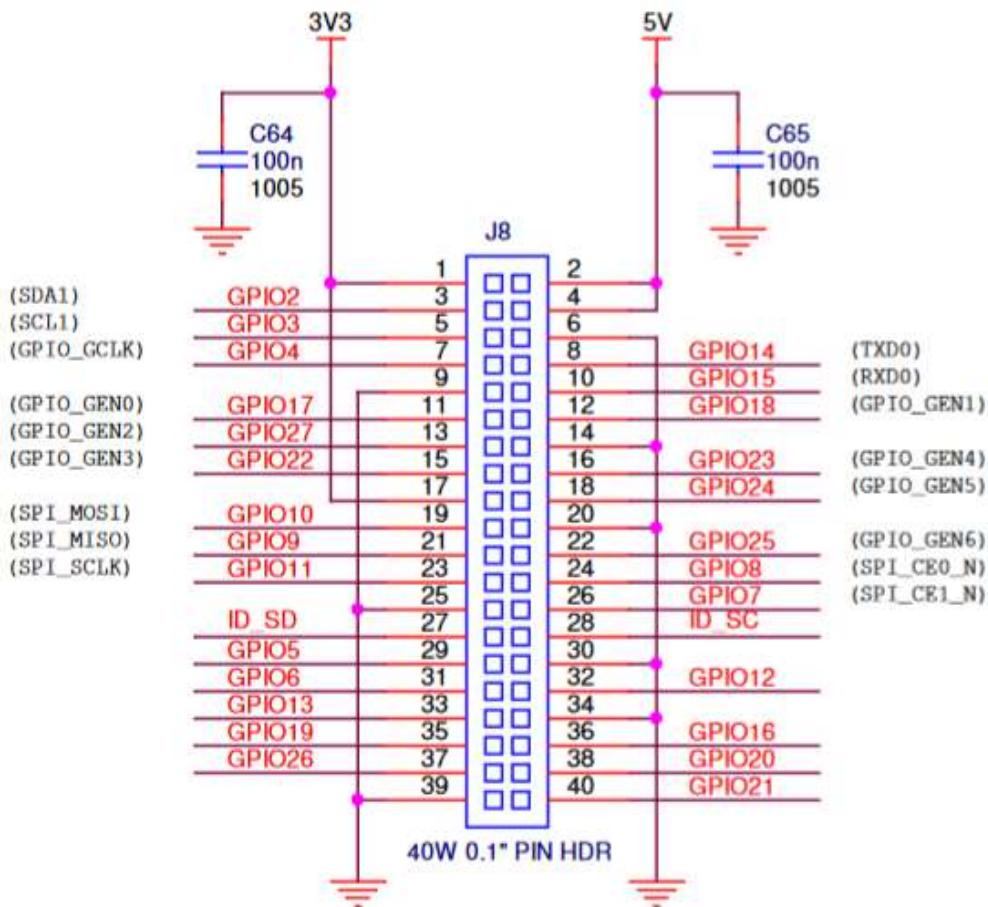
- magistrale: SPI (ang. Serial Peripheral Interface), I2C (ang. Inter-Integrated Circuit), UART (ang. Universal Asynchronous Receiver–Transmitter), I2S (ang. Inter-IC Sound),
- 40-pinowy moduł GPIO (ang. General Purpose Input Output),
- 4 porty USB 2.0 (ang. Universal Serial Bus),
- 2 generatory PWM,
- kontrolery przerwań.

Raspberry Pi 3 Model B charakteryzuje się również tym, że posiada dodatkowe moduły, które są obsługiwane po zainstalowaniu rekomendowanego przez producenta systemu operacyjnego Raspbian:

- 1x gniazdo HDMI video/audio,
- 1x gniazdo RCA,

- 1x port Ethernet 10/100 Mb/s,
- moduł Bluetooth 4.1,
- moduł WiFi 802.11 b/g/n,
- złącze CSI (ang. Camera Serial Interface) do podłączenia kamery oraz złącze DSI (ang. Digital Serial Interface) do podłączenia wyświetlacza.

Niewątpliwie dużą zaletą Raspberry Pi jest jego rozbudowany moduł GPIO, czyli złącze, na którym znajdują się wyprowadzenia ogólnego przeznaczenia konfigurowalne jako wejścia lub wyjścia. Dodatkowo moduł posiada cztery piny zasilające (2 z potencjałem 5 V i 2 z potencjałem 3,3 V) oraz osiem pinów z potencjałem zerowym. Umożliwiają one zasilanie urządzeń o łącznym poborze natężenia prądu elektrycznego przy pracy ciągłej o wartości 50 mA. Poniżej przedstawiono schemat elektryczny modułu GPIO (rys. 5.10).



Rys. 5.10. Schemat elektryczny modułu GPIO [12]

Korzystanie z komputera typu Raspberry Pi (podobnie jak w komputerze klasy PC) wymaga podłączenia niezbędnych urządzeń takich jak np. klawiatura, mysz i monitor. Jeżeli jednak nie dysponujemy tego typu sprzętem, można połączyć się z jednostką za pomocą protokołu SSH (ang. Secure Shell) używając do tego kabla ethernetowego.

5.5. Czujniki temperatury

Komputer Raspberry Pi 3 Model B nie posiada wejść i wyjść analogowych ani przetworników analogowo-cyfrowych i cyfrowo-analogowych. W związku z tym uzasadniony jest wybór cyfrowej klasy czujników temperatury, do jakiej należy DS18B20 firmy Dallas Semiconductor (rys. 5.11).



Rys. 5.11. Czujnik temperatury DS18B20

DS18B20 jest bardzo chętnie wykorzystywany w projektach studenckich jak i w prostych aplikacjach komercyjnych. Umieszczenie termometru w sondzie pomiarowej pozwoli na wykorzystanie go do pomiaru temperatury powietrza (rys. 5.12). Jego podstawowe właściwości i parametry techniczne zostały przedstawione w tabeli 5.4.



Rys. 5.12. Sonda pomiarowa z czujnikiem temperatury DS18B20

Tabela 5.4. Dane techniczne czujnika temperatury DS18B20

Parametr	Wartość
Napięcie zasilania	3,3 V – 5,0 V
Zakres pomiarowy	-55 °C – 125 °C
Dokładność *	±0,5 °C
Rozdzielcość	9 – 12 bitów

* dokładność maksymalna osiągana w zakresie temperatur od -10 °C do 85 °C, poza tym zakresem dokładność może wynosić nawet ponad ±2 °C

Cyfrowe czujniki temperatury pełnią rolę podrzędną, więc muszą posiadać interfejs komunikacyjny, aby istniała możliwość odczytu temperatury. DS18B20 wyposażono w protokół

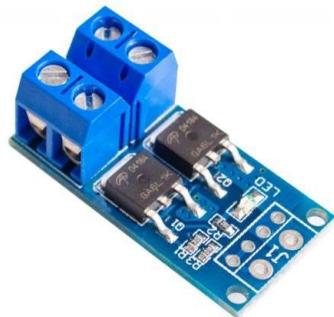
1-Wire opracowany przez firmę Dallas Semiconductor. W tym protokole komunikacja odbywa się w trybie master-slave. Funkcję mastera pełni mikrokontroler, który zbiera informacje z urządzeń typu slave (czujników) za pomocą odpytywania.

Interfejs ten charakteryzuje się stosunkowo małą przepustowością, jednak stosowany w niewielkich urządzeniach takich jak: instrumenty metrologiczne, zamki elektroniczne, cyfrowe czujniki temperatury, radzi sobie bardzo dobrze.

Główną zaletą DS18B20 jest możliwość podłączenia wielu urządzeń w trybie pasożytniczym. Oznacza to, że te urządzenia nie wymagają osobnej linii zasilającej a zatem interfejs korzysta tylko z jednej linii, przeznaczonej do przesyłania danych. W praktyce do prawidłowego działania potrzebne jest jeszcze podłączenie linii z potencjałem zerowym (tzw. masy). Termometr wyposażono w kondensator, który jest ładowany podczas transmisji danych a następnie energia w nim zgromadzona służy do zasilania układu.

5.6. Moduł sterownika PWM

Rolę regulatorów: mocy grzałki i prędkości obrotowej wentylatora pełnią moduły sterowników PWM (rys. 5.12).



Rys. 5.13. Moduł sterownika PWM

Moduł przedstawiony na rys 5.13 wyposażono w dwa tranzystory typu MOSFET (ang. Metal–Oxide Semiconductor Field–Effect Transistor) o maksymalnej mocy wyjściowej 400 W. Układ zasilany napięciem stałym 5 – 36 V generuje na wyjściu maksymalne natężenie prądu elektrycznego o wartości 15 A. Prawidłowe działanie modułu uwarunkowane jest podaniem sygnału sterującego o amplitudzie w zakresie od 2 V do 20 V, więc jest on kompatybilny z logiką procesora Raspberry Pi 3 Model B.

Zasada działania sterownika opiera się na regulacji sygnału napięciowego o stałej amplitudzie i częstotliwości poprzez zmianę wypełnienia sygnału. Modulacja realizowana jest przez cykliczne załączanie i wyłączanie tranzystorów bądź tyristorów. Czynność nazywana jest często kluczowaniem.

Moduły sterowników PWM są stosowane najczęściej w zasilaczach impulsowych lub wzmacniaczach impulsowych. Dedykowane są do sterowania urządzeniami o dużym poborze mocy tj. silnikami prądu stałego, grzałkami elektrycznymi.

6. IMPLEMENTACJA ALGORYTMU IDENTYFIKACJI

Zaimplementowano algorytmy: RLS do identyfikacji obiektu off-line oraz EF-RLS do identyfikacji obiektu w czasie rzeczywistym (on-line). W niniejszym rozdziale znajduje się opis: badań charakterystyki obiektu, doboru struktury modelu oraz przyjętych założeń i parametrów startowych algorytmów identyfikacji. W każdym z podrozdziałów przedstawiono wyniki eksperymentów wykonanych na zbudowanym stanowisku.

6.1. Badanie charakterystyki obiektu

Prace rozpoczęto od określenia charakterystyki obiektu i wyznaczenia jego parametrów dynamicznych, które są niezbędne do przeprowadzenia dalszych badań. Wyznaczenie stałej czasowej procesu pozwoli na wyznaczenie właściwego okresu próbkowania, determinującego prawidłowe działania algorytmów.

Obiekt pobudzono skokiem jednostkowym. Na jego wejście podano maksymalną, bezpieczną wartość procentową mocy zasilacza grzałki, która wynosi 90%. Ograniczona jest ona sprawnością zasilacza. Przy większych nastawach regulatora PWM jego moc już nie wzrasta a nadmiar energii zamieniany jest w ciepło, co może doprowadzić do przegrzania zasilacza. Okres próbkowania przyjęto z góry. Odczytu temperatury na wyjściu obiektu dokonywano co 10 s. Sygnał sterujący prędkością obrotową wentylatora nastawiono na 50%. Zebrane wyniki pomiarów przedstawiono na wykresie (rys. 6.1).

Analizując wykres odpowiedzi skokowej obiektu, zauważono, że jest ona zbliżona do charakterystyki obiektu inercyjnego pierwszego rzędu z opóźnieniem. Skorzystano więc z metod wyznaczania charakterystyk dynamicznych właśnie dla takich obiektów i wyznaczono stałą czasową procesu w sposób analityczny [13].

Za wartość ustaloną odpowiedzi obiektu została uznana próbka, dla której względna wartość zmian wyjścia obiektu była mniejsza bądź równa 2% przez co najmniej cztery kolejne próbki, tzn.:

$$\frac{y(t) - y(t-1)}{y(t)} \leq 2\%, \quad (6.1)$$

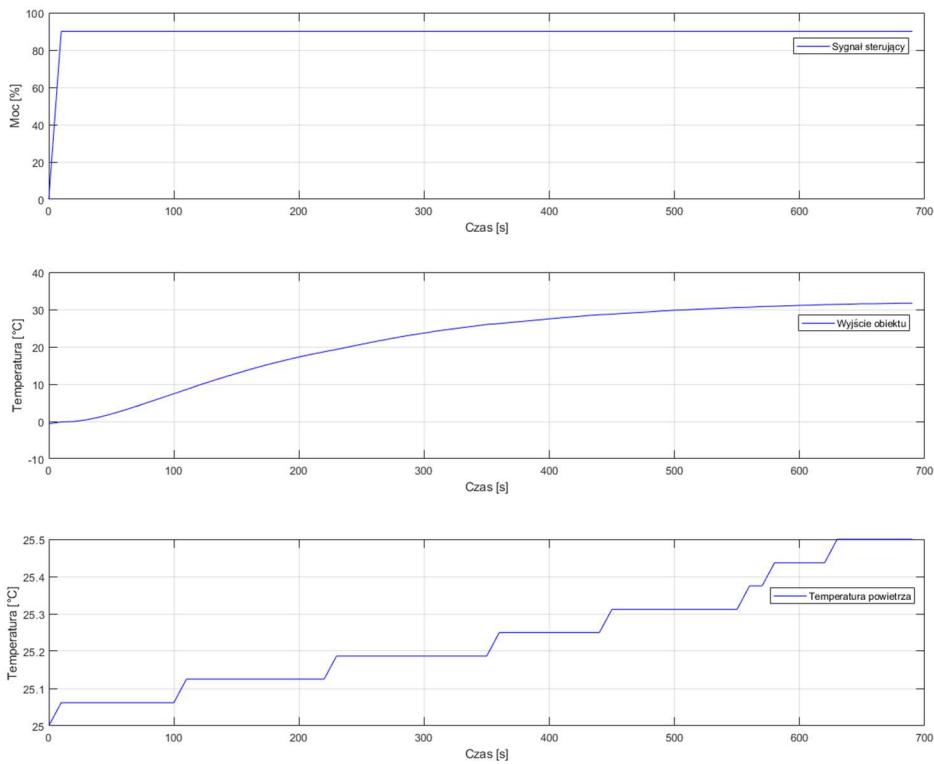
$$y_{ust} = 24,688 \text{ } ^\circ\text{C}. \quad (6.2)$$

Wartość, dla której sygnał wyjściowy obiektu osiąga 63% ustalonej wartości odpowiedzi skokowej obiektu wynosi:

$$y_{63\%} = 0,63y_{ust} = 15,553 \text{ } ^\circ\text{C} \quad (6.3)$$

i osiągana jest dla próbki numer 18. Stała czasowa obiektu wynosi więc:

$$\tau = 180 \text{ s.} \quad (6.4)$$



Rys. 6.1. Charakterystyka obiektu, od góry: sygnał sterujący, sygnał wyjściowy, temperatura powietrza w pokoju

Opóźnienie transportowe obiektu, dla tak dobranej nastawy regulatora prędkości obrotowej wentylatora jak w tym eksperymentie, jest mniejsze od okresu próbkowania:

$$t_d < 10 \text{ s.} \quad (6.5)$$

Okres próbkowania powinien być dobrany tak, aby stanowił maksymalnie 10% stałej czasowej obiektu, tzn.:

$$T_s \leq 0,1\tau. \quad (6.6)$$

Zgodnie z podaną zależnością (6.6) okres próbkowany został przyjęty prawidłowo i taka jego wartość będzie wykorzystywana przy implementacji algorytmów: identyfikacji oraz sterownika.

6.2. Wybór struktury modelu

Po zbadaniu charakterystyki obiektu wykonano pierwszą jego identyfikację, stosując algorytm RLS z modelem ARX. Wysterowanie prędkości obrotowej wentylatora wynosi 50%. Obiekt pobudzono sygnałem przedstawionym na rys. 6.2. Sygnał wyjściowy obiektu stanowi różnicę:

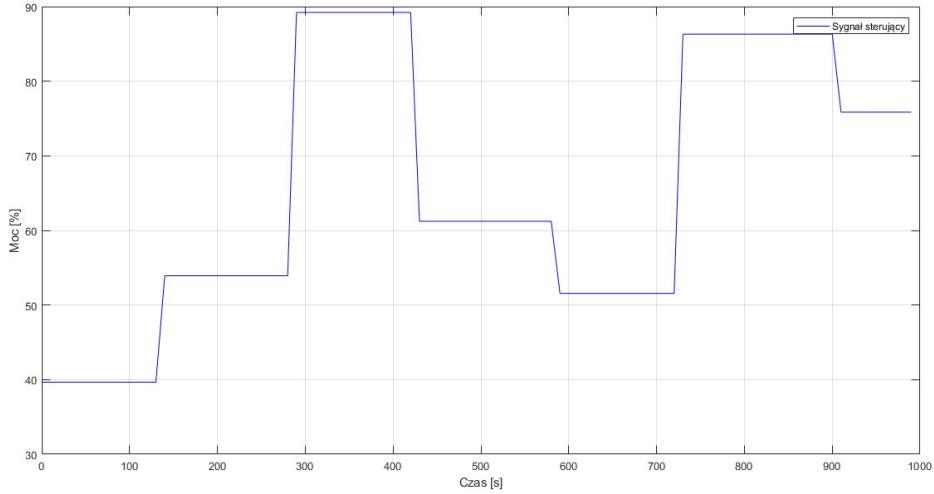
$$y(t) = Temp_1(t) - Temp_2(t), \quad (6.7)$$

gdzie:

$Temp_1(t)$ – temperatura z czujnika umieszczonego na wylocie kanału,

$Temp_2(t)$ – temperatura powietrza nawiewanego.

Mimo, że badany obiekt wykazuje właściwości obiektu inercyjnego pierwszego rzędu, zdecydowano się zwiększyć stopień wielomianu $A(z^{-1})$ o jeden (a następnie w kolejnym eksperymencie o dwa) z uwagi na obecność opóźnienia transportowego. Sprawdzono, który model lepiej odzwierciedla dynamikę badanego obiektu a także posłużono się kryterium informacyjnym Akaike'go do oceny struktury modelu.



Rys. 6.2. Sygnał sterujący

6.2.1. Identyfikacja obiektu modelem ARX pierwszego rzędu

Zebranie wyników pomiarowych i zapisanie ich do pliku pozwoliło przeprowadzić eksperyment identyfikacji dla modelu $ARX(1,1)$:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t), \quad (6.8)$$

$$A(z^{-1}) = 1 + a_1z^{-1}, \quad (6.9)$$

$$B(z^{-1}) = b_0, \quad (6.10)$$

gdzie opóźnienie $d = 1$, czyli wektor regresji $\varphi^T(t)$ i wektor parametrów θ można zapisać:

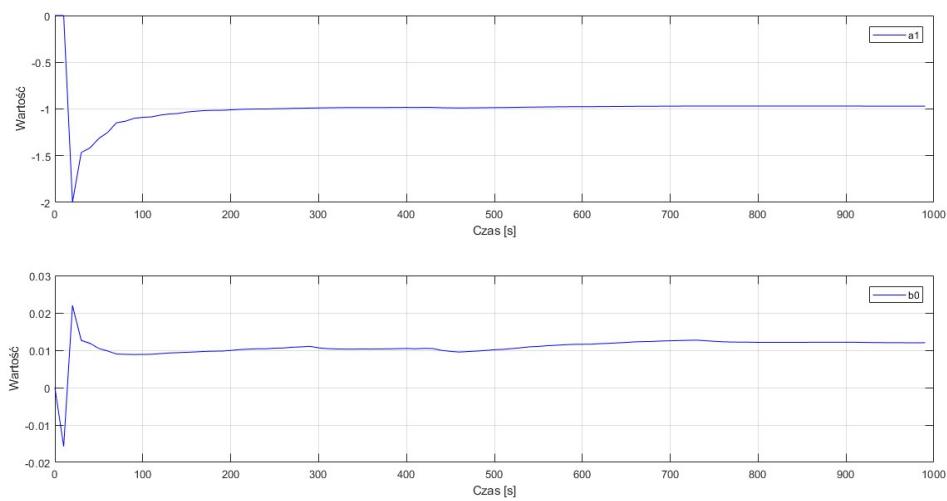
$$\varphi^T(t) = [-y(t-1), u(t-1)], \quad (6.11)$$

$$\theta = [a_1, b_0]^T. \quad (6.12)$$

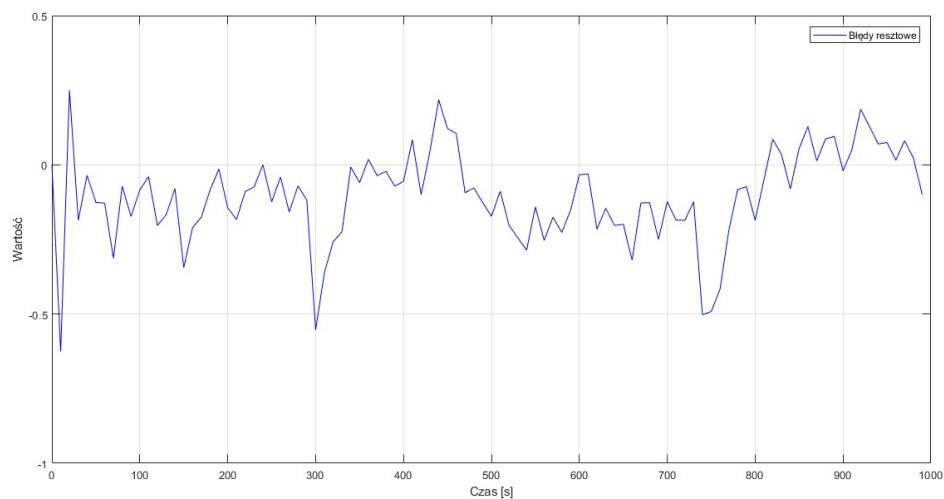
Uzyskane wyniki w postaci estymowanych parametrów modelu $ARX(1,1)$ zostały przedstawione na rys. 6.3. Wykres błędów resztowych modelu znajduje się na rys. 6.4. Ustalone wartości parametrów modelu uzyskane podczas eksperymentu identyfikacji wykorzystano do utworzenia transmitancji zastępczej dyskretnej obiektu regulacji:

$$G_{ARX(1,1)}(z^{-1}) = z^{-1} \frac{b_0}{1 + a_1 z^{-1}}. \quad (6.13)$$

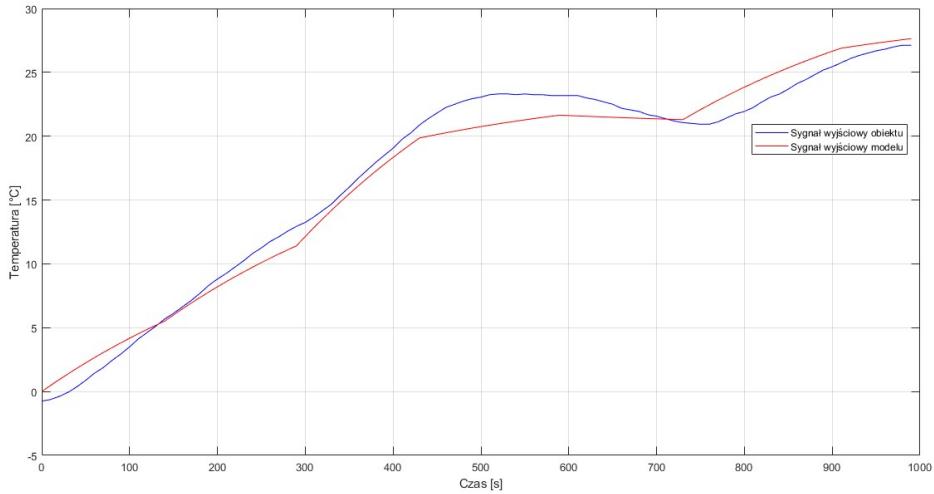
Wyznaczono odpowiedź modelu obiektu rzeczywistego na zadany sygnał sterujący i porównano ją z odpowiedzią obiektu rzeczywistego (rys. 6.5).



Rys. 6.3. Estymowane parametry modelu $ARX(1,1)$



Rys. 6.4. Błędy resztowe modelu $ARX(1,1)$



Rys. 6.5. Porównanie sygnałów: wyjściowego obiektu rzeczywistego i wyjściowego modelu $ARX(1,1)$

6.2.2. Identyfikacja obiektu modelem ARX drugiego rzędu

Następnie dla tych samych danych pomiarowych (sygnału sterującego i sygnału wyjściowego) dokonano ponownej identyfikacji obiektu metodą RLS dla modelu $ARX(2,1)$. Wartość opóźnienia transportowego wynosi $d = 1$. Zatem następuje zmiana w wielomianie:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2}, \quad (6.14)$$

oraz wektorze regresji:

$$\boldsymbol{\varphi}^T(t) = [-y(t-1), -y(t-2), u(t-1)], \quad (6.15)$$

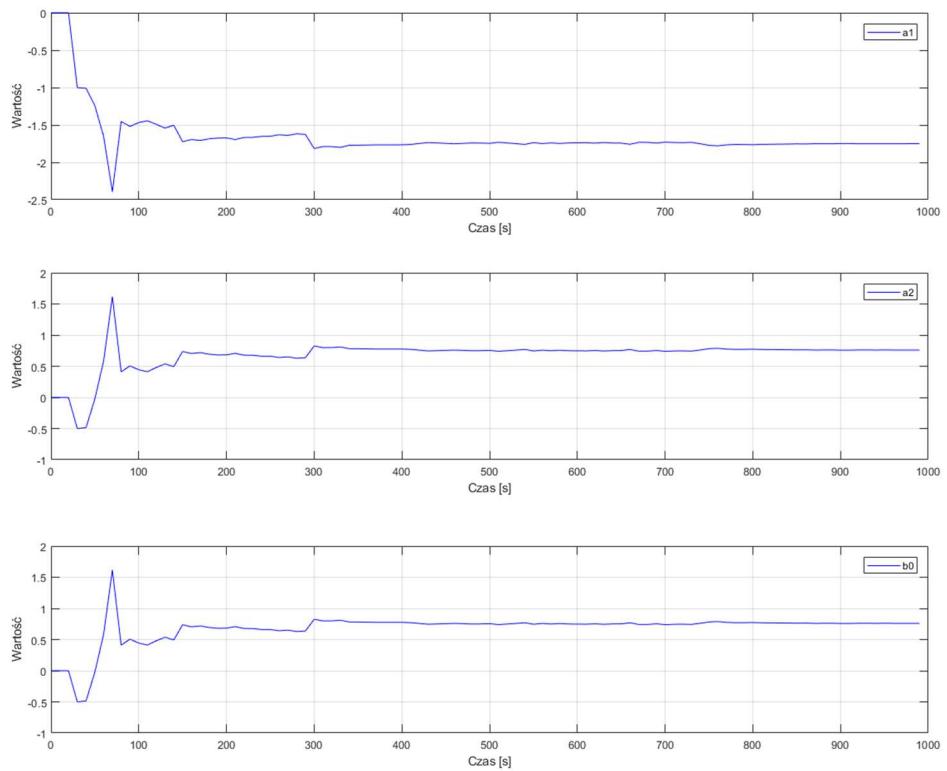
a wektor parametrów wygląda następująco:

$$\boldsymbol{\theta} = [a_1, a_2, b_0]^T. \quad (6.16)$$

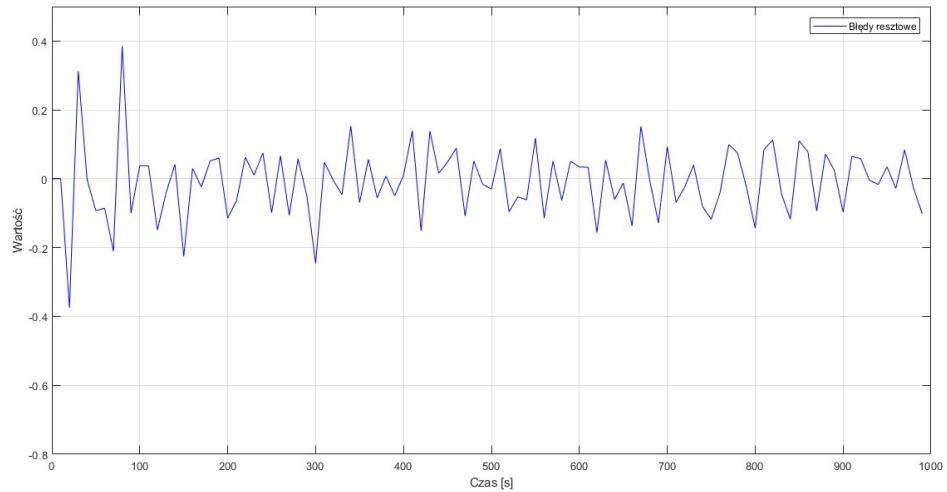
Uzyskane wyniki w postaci estymowanych parametrów modelu $ARX(2,1)$ zostały przedstawione na rys. 6.6. Wykres błędów resztowych modelu znajduje się na rys. 6.7. Ustalone wartości parametrów modelu uzyskane podczas eksperymentu identyfikacji wykorzystano do utworzenia transmitancji zastępczej dyskretnej obiektu regulacji:

$$G_{ARX(2,1)}(z^{-1}) = z^{-1} \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}}. \quad (6.17)$$

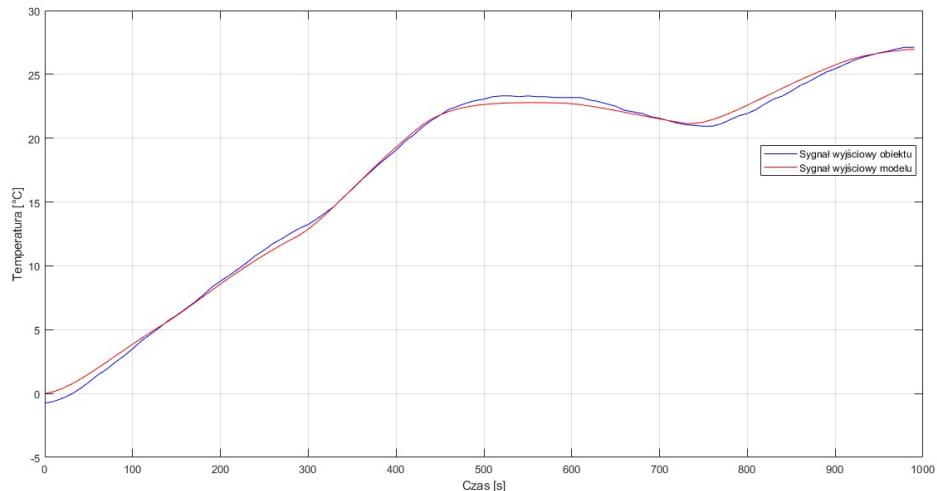
Wyznaczono odpowiedź modelu obiektu rzeczywistego na zadany sygnał sterujący i porównano ją z odpowiedzią obiektu rzeczywistego (rys. 6.8).



Rys. 6.6. Estymowane parametry modelu $ARX(2,1)$



Rys. 6.7. Błędy resztowe modelu $ARX(2,1)$



Rys. 6.8. Porównanie sygnałów: wyjściowego obiektu rzeczywistego i wyjściowego modelu $ARX(2,1)$

6.2.3. Identyfikacja obiektu modelem ARX trzeciego rzędu

Następnie dla tych samych danych pomiarowych (sygnału sterującego i sygnału wyjściowego) dokonano ponownej identyfikacji obiektu metodą RLS dla modelu $ARX(3,1)$. Wartość opóźnienia transportowego wynosi $d = 1$. Zatem następuje zmiana w wielomianie:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}, \quad (6.18)$$

oraz wektorze regresji:

$$\varphi^T(t) = [-y(t-1), -y(t-2), -y(t-3), u(t-1)], \quad (6.19)$$

a wektor parametrów wygląda następująco:

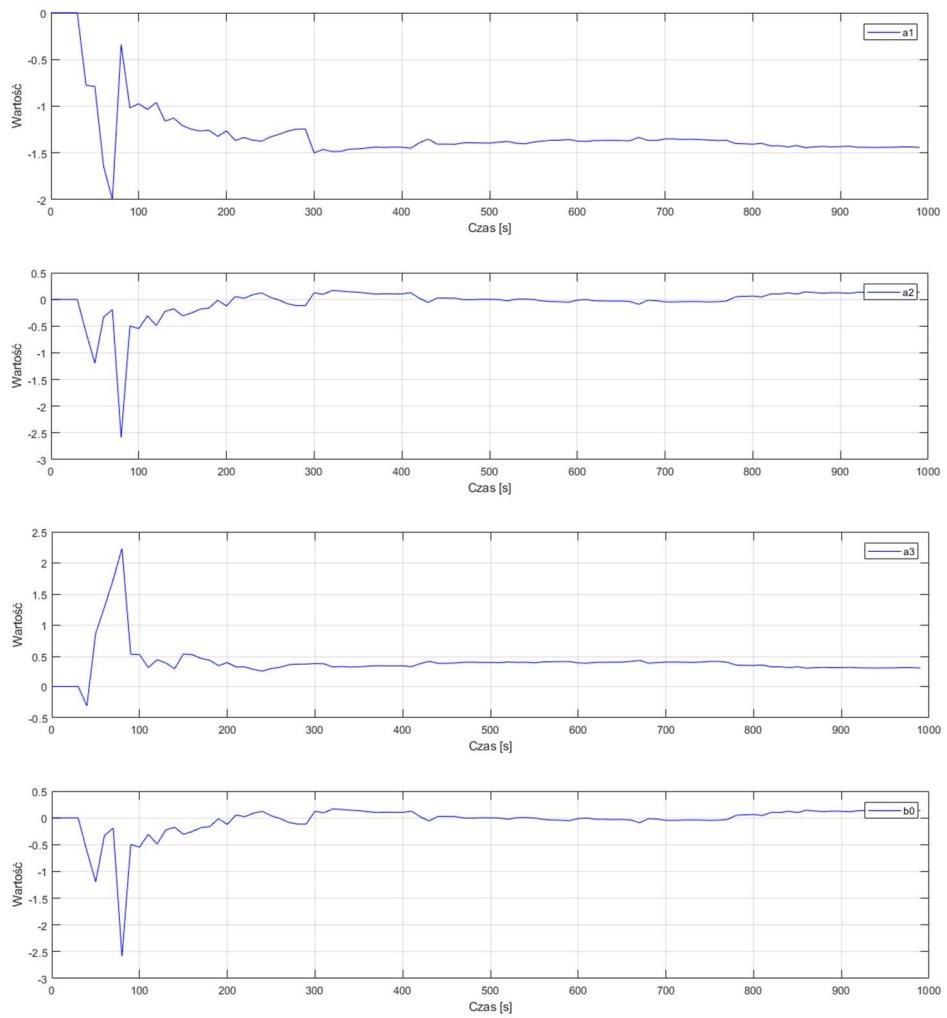
$$\boldsymbol{\theta} = [a_1, a_2, a_3, b_0]^T. \quad (6.20)$$

Uzyskane wyniki w postaci estymowanych parametrów modelu $ARX(3,1)$ zostały przedstawione na rys. 6.9. Wykres błędów resztowych modelu znajduje się na rys. 6.10.

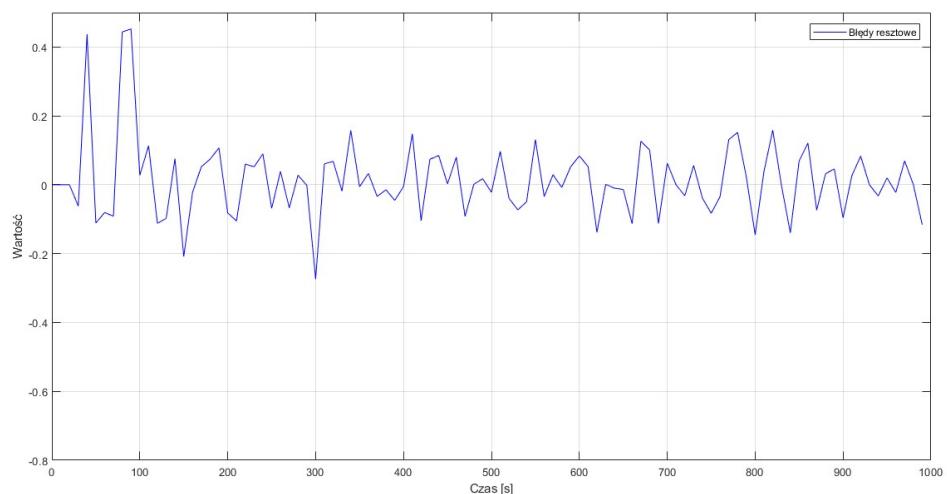
Ustalone wartości parametrów modelu uzyskane podczas eksperymentu identyfikacji wykorzystano do utworzenia transmitancji zastępczej dyskretniej obiektu regulacji:

$$G_{ARX(3,1)}(z^{-1}) = z^{-1} \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}, \quad (6.21)$$

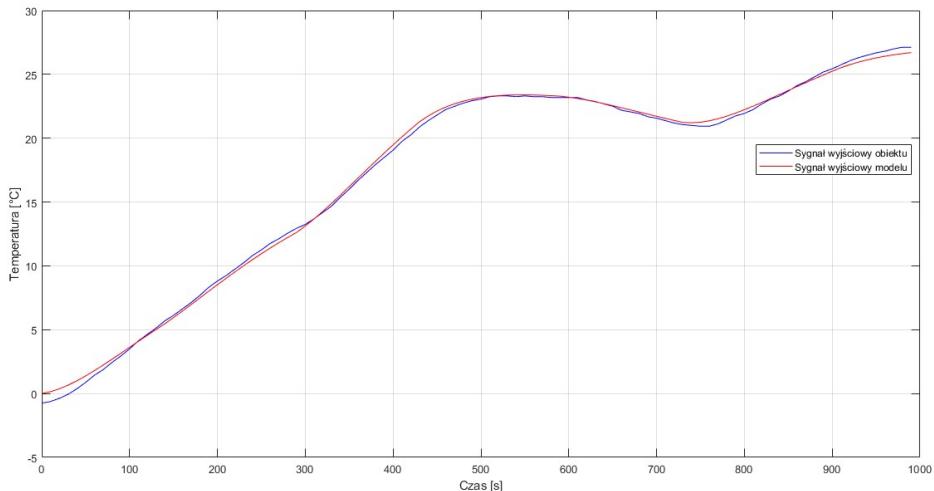
wyznaczono odpowiedź modelu obiektu rzeczywistego na zadany sygnał sterujący i porównano ją z odpowiedzią obiektu rzeczywistego (rys. 6.11).



Rys. 6.9. Estymowane parametry modelu $ARX(3,1)$



Rys. 6.10. Błędy resztowe modelu $ARX(3,1)$



Rys. 6.11. Porównanie sygnałów: wyjściowego obiektu rzeczywistego i wyjściowego modelu $ARX(3,1)$

6.3. Zestawienie wyników

Podsumowano wyniki wszystkich eksperymentów identyfikacji. W tym celu zapisano ustalone wartości parametrów uzyskane dla różnych modeli i zaprezentowano je w tabeli 6.1.

Tabela 6.1. Ustalone wartości parametrów modeli ARX obiektu rzeczywistego

Parametr\Model	$ARX(1,1)$	$ARX(2,1)$	$ARX(3,1)$
b_0	0,012	0,0036	0,0038
a_1	-0,97	-1,75	-1,4412
a_2	-	0,7599	0,1388
a_3	-	-	0,313

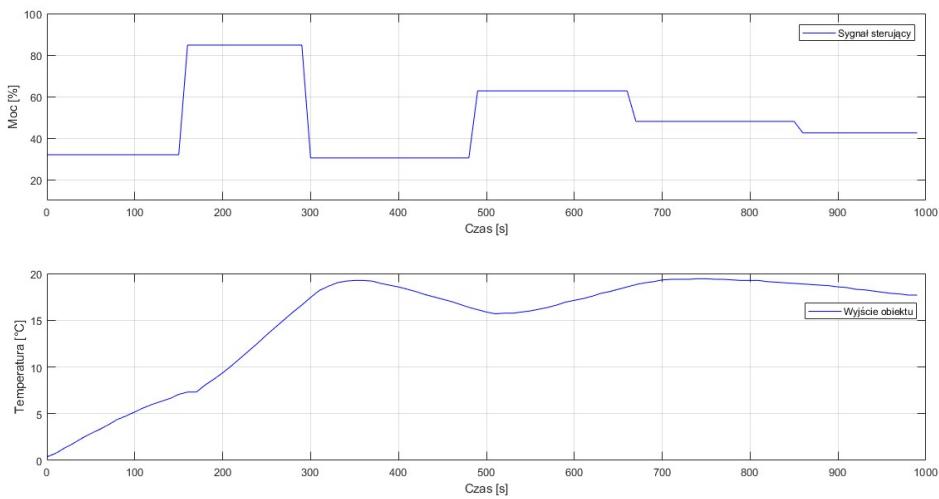
Na rys. 6.5 widać, że krzywe: sygnału wyjściowego modelu i sygnału wyjściowego obiektu rzeczywistego nieco odbiegają od siebie. Wydaje się, że model pierwszego rzędu jest zbyt ubogi na ten obiekt. Wykres błędów resztowych tego modelu (rys. 6.4) posiada największą wariancję.

Sygnał wyjściowy drugiego modelu zdecydowanie lepiej „naśladuje” zachowanie obiektu rzeczywistego, co sprawia, że model ten może być brany pod uwagę w algorytmie sterowania. Dodanie jednego parametru spowodowało znaczną redukcję wariancji błędów resztowych.

W modelu najwyższego rzędu wariancja błędów resztowych jest najmniejsza, więc w najlepszy sposób odwzorowuje on dynamikę obiektu. Jednak różnica pomiędzy odpowiedzią tego modelu a odpowiedzią modelu drugiego rzędu nie jest już tak duża jak pomiędzy modelami $ARX(1,1)$ i $ARX(2,1)$. Oznacza to, że nie poprawi on w znaczący sposób jakości regulacji a z pewnością doprowadzi do zwiększenia złożoności obliczeniowej algorytmu.

6.4. Kryterium informacyjne Akaike'go

Wykonanie serii eksperymentów identyfikacyjnych pozwoliło na uzyskanie trzech różnych struktur modelu ARX. Posłużono się kryterium informacyjnym Akaike'go do ich oceny. W tym celu zebrano kolejne dane pomiarowe, które użyto do walidacji (rys. 6.12). Wyniki obliczeń przedstawiono w tabeli 6.2.



Rys. 6.12. Zestaw danych do walidacji systemu. Okres próbkowania wynosi 10 s.

Tabela 6.2. Wyniki zastosowania kryterium informacyjnego Akaike'go do walidacji danych

$AIC(2)$	$AIC(3)$	$AIC(4)$
1,3508	3,3588	5,3594

gdzie:

$AIC(p)$ – wartość kryterium Akaike'go dla modelu o p wolnych parametrach.

Z przeprowadzonego testu statystycznego wynika, że model $ARX(1,1)$ najbardziej nadaje się do reprezentowania obiektu rzeczywistego, ponieważ wartość $AIC(2)$ w tabeli 6.2 jest najmniejsza. Model drugiego rzędu wypada nieco słabiej i na podstawie badań można stwierdzić, że jest modelem przeparametryzowanym. Zdecydowanie najsłabiej w tym zestawieniu prezentuje się model o największej ilości parametrów, czyli $ARX(3,1)$.

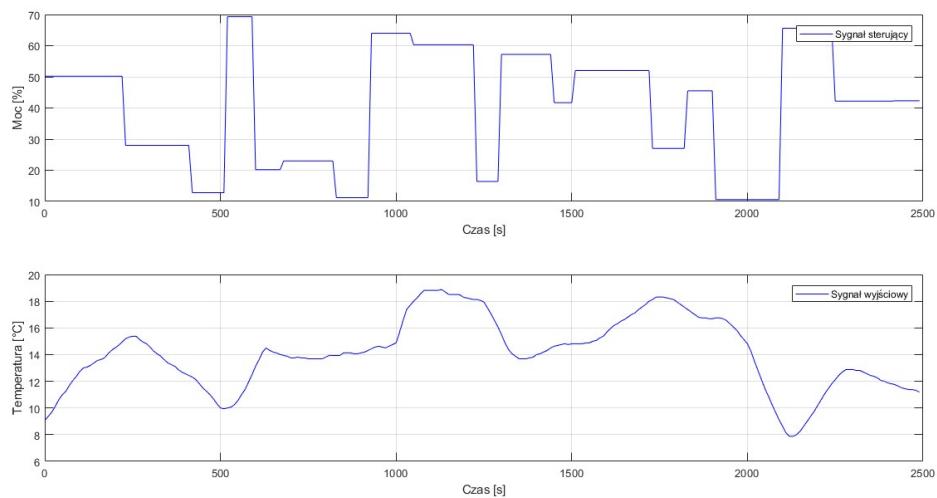
6.5. Identyfikacja obiektu w czasie rzeczywistym

Kolejny etap badań stanowi wykonanie identyfikacji obiektu w czasie rzeczywistym dla zmieniających się parametrów obiektu. W tym celu wykorzystano algorytm EF-RLS (dodatek A). Współczynnik zapominania λ ustawiono na wartość 0,99. Wykorzystano model $ARX(2,1)$. Obiekt pobudzono sygnałem sterującym o generowanej losowo amplitudzie w różnych chwilach czasu po to, aby osiągnąć szybką zbieżność estymowanych parametrów modelu (rys. 6.13).

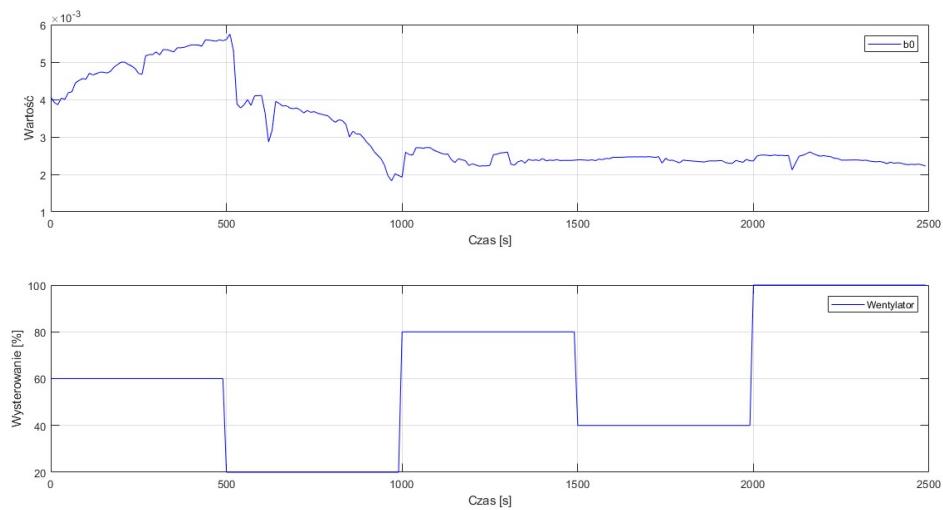
W trakcie eksperymentu dokonywano zmian opóźnienia transportowego. Po upływie 400 s podawano nową wartość na wejście regulatora PWM odpowiedzialnego za sterowanie prędkością wentylatora. Uzyskane wyniki w postaci ustalonych wartości parametrów modelu dla poszczególnych nastaw regulatora prędkości obrotowej wentylatora zebrano w tabeli 6.3 i przedstawiono w sposób graficzny (rys. 6.14 – 6.15). Z uwagi na proces ustalania się parametrów modelu obiektu na wykresach (rys. 6.13 – 6.15) pominięto 50 pierwszych próbek.

Tabela 6.3. Zmieniające się parametry modelu obiektu w zależności od nastaw regulatora prędkości obrotowej wentylatora. Przedstawione w tabeli wartości są ostatnimi współczynnikami modelu tuż przed zmianą prędkości. Współczynnik zapominania $\lambda = 0,99$.

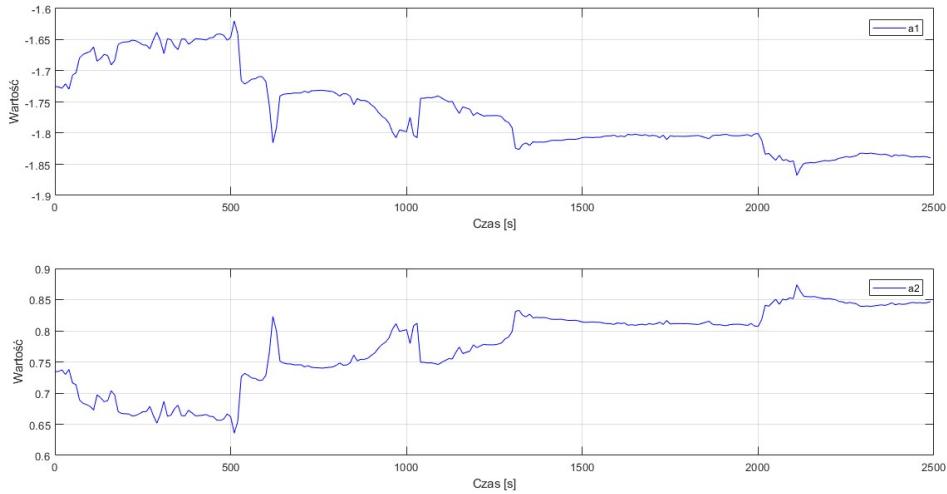
Wysterowanie	b_0	a_1	a_2
40%	0,0041	-1,7197	0,7293
60%	0,0056	-1,6509	0,6664
20%	0,002	-1,7963	0,8005
80%	0,0024	-1,8089	0,8159
40%	0,0024	-1,8013	0,8076
100%	0,0022	-1,8399	0,8469



Rys. 6.13. Sygnały: wejściowy i wyjściowy obiektu. Okres próbkowania wynosi 10 s.



Rys. 6.14. Zmieniający się parametr modelu b_0 obiektu oraz wysterowanie prędkości obrotowej wentylatora. Współczynnik zapominania $\lambda = 0,99$.

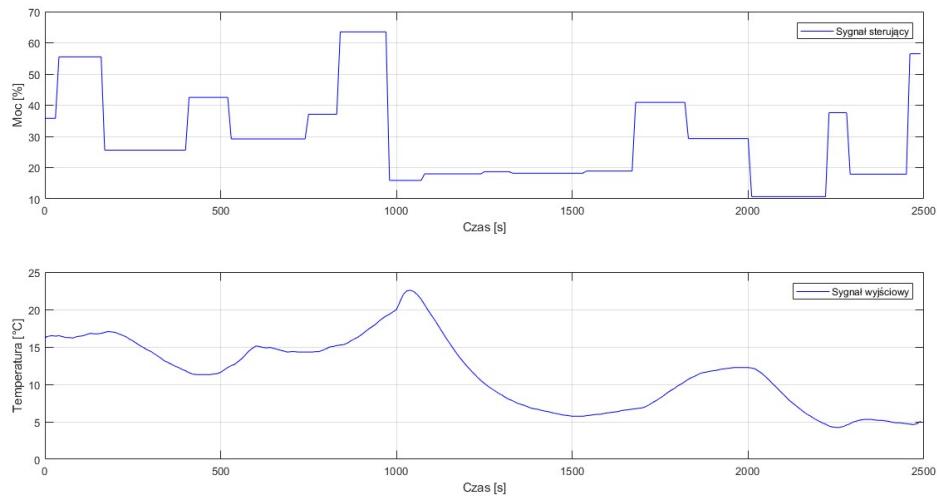


Rys. 6.15. Zmieniające się parametry modelu (a_1 i a_2) obiektu. Współczynnik zapominania $\lambda = 0,99$.

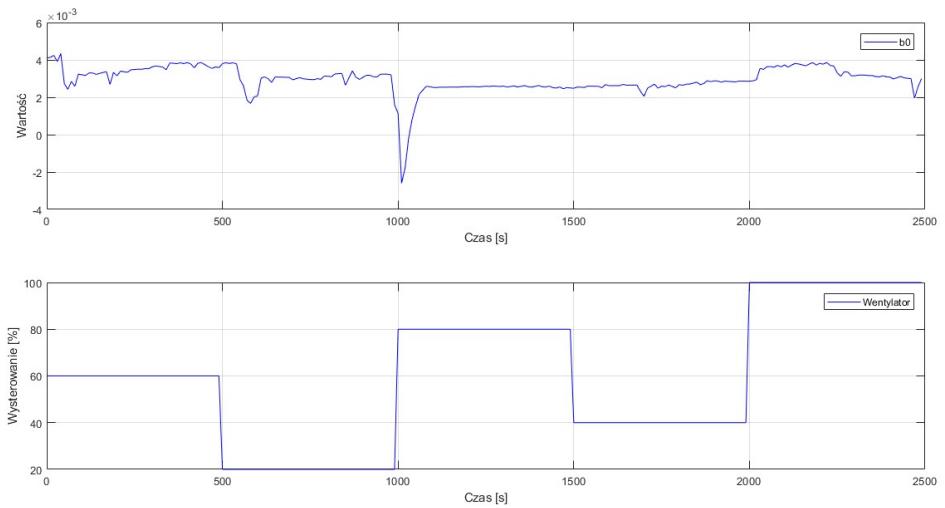
Następnie eksperyment powtórzono wykorzystując te same nastawy prędkości obrotowej wentylatora, jedynie zmieniając współczynnik zapominania na wartość $\lambda = 0,97$. Wyniki zaprezentowano w taki sam sposób w tabeli 6.4 oraz na wykresach (rys. 6.16 – 6.18).

Tabela 6.4 Zmieniające się parametry modelu obiektu w zależności od nastaw regulatora prędkości obrotowej wentylatora. Przedstawione w tabeli wartości są ostatnimi współczynnikami modelu tuż przed zmianą prędkości. Współczynnik zapominania $\lambda = 0,97$.

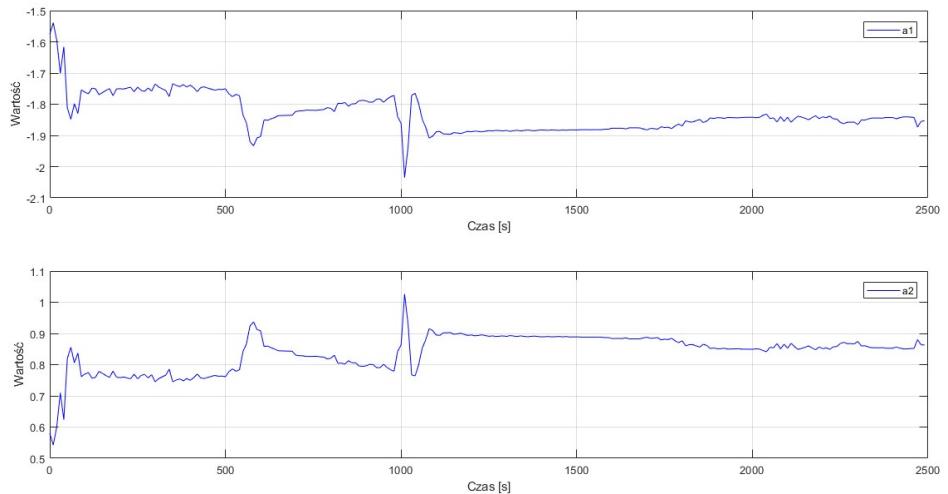
Wysterowanie	b_0	a_1	a_2
40%	0,0027	-1,5989	0,5971
60%	0,0036	-1,7528	0,7633
20%	0,0016	-1,8394	0,8423
80%	0,0025	-1,8824	0,8889
40%	0,0029	-1,8419	0,8494
100%	0,0030	-1,8531	0,8629



Rys. 6.16. Sygnały: wejściowy i wyjściowy obiektu. Okres próbkowania wynosi 10 s.



Rys. 6.17. Zmieniający się parametr modelu b_0 obiektu oraz wysterowanie prędkości obrotowej wentylatora. Współczynnik zapominania $\lambda = 0,97$.



Rys. 6.18. Zmieniające się parametry modelu (a_1 i a_2) obiektu. Współczynnik zapominania $\lambda = 0,97$.

6.6. Wnioski

Przy zmiennym opóźnieniu transportowym parametry obiektu ulegają zmianie. Im współczynnik zapominania jest mniejszy, tym mniej przeszłych próbek algorytm identyfikacji bierze pod uwagę przy wyznaczaniu aktualnych parametrów. Z przeprowadzonych badań wynika, że skutkuje to większymi oscylacjami estymat współczynników modelu obiektu (należy porównać wykresy rys. 6.14 – 6.15 z rys. 6.17 – 6.18) ale dzięki temu powoduje szybszą zbieżność, gdy obiekt zmieni swoje właściwości.

W momencie, gdy następuje zmiana wartości sygnału sterującego, można zauważać, że parametry modelu nieco się zmieniają. Sygnał ten na nowo pobudza obiekt i dzięki temu można uzyskać jeszcze więcej informacji o nim w eksperymencie identyfikacji.

Większe zmiany parametrów zaobserwowano podczas, gdy amplituda sygnału wejściowego w kolejnych próbkach przyjmuje mniejsze wartości. Wyraźnie widać to na rys. 6.16,

gdzie wejście obiektu znacząco spada w chwili nieco wcześniejszej od $t = 1000$ s. Parametry obiektu, dokładnie w tej samej chwili t , zaczynają odbiegać od wcześniejszej wartości ustalonej. Spowodowane jest to tym, że podczas ochładzania się obiektu jego model jest inny niż wtedy, gdy obiekt zwiększa swoją temperaturę.

7. IMPLEMENTACJA ALGORYTMU STEROWNIKA

Dokonano implementacji sterownika predykcyjnego GPC (dodatek B). Zaprezentowano sposób, w jaki dokonano jego syntezy oraz opisano dobór horyzontu predykcji. W tym celu przeprowadzono serię badań dla różnych wartości horyzontu predykcji, zebrane wyniki pomiarów i przedstawiono na wykresach. Dalszą część prac stanowi implementacja dwóch różnych typów trajektorii referencyjnej i przeprowadzenie analizy jakości sterowania dla dobranych wcześniej nastaw regulatora GPC. Badania podzielono na dwie grupy w zależności od tego, czy prędkość wentylatora (a tym samym opóźnienie transportowe obiektu) ulegała zmianie podczas trwania eksperymentu. Wyniki zaprezentowano na wykresach i przedstawiono wnioski.

Następnie dokonano syntezy regulatora ST-GPC. Na podstawie [5] przedstawiono sposób jego implementacji oraz wykonano serię badań dla zmieniającego się opóźnienia transportowego obiektu. Sporządzono wykresy odpowiedzi obiektu, sygnału sterującego i zmieniających

się parametrów modelu obiektu w zależności od prędkości obrotowej wentylatora. Na końcu podrozdziału opisano wnioski ogólne.

7.1. Generalized Predictive Control

Wcześniejsze eksperymenty pozwoliły wyznaczyć strukturę modelu obiektu, która może być użyta do sterowania nim. Do budowy modelu CARIMA wykorzystano parametry uzyskane w wyniku identyfikacji modelem ARX. Zastosowano model CARIMA rzędu drugiego, tzn.:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t) + C(z^{-1})\frac{\eta(t)}{\Delta}, \quad (7.1)$$

$$\Delta = 1 - z^{-1}, \quad (7.2)$$

gdzie:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2}, \quad (7.3)$$

$$B(z^{-1}) = b_0, \quad (7.4)$$

$$C(z^{-1}) = 1 \quad (7.5)$$

a opóźnienie transportowe $d = 1$. Współczynniki wielomianów $A(z^{-1})$ i $B(z^{-1})$ pochodzą z modelu $ARX(2,1)$:

$$a_1 = -1,75, \quad (7.6)$$

$$a_2 = 0,7599, \quad (7.7)$$

$$b_0 = 0,0036, \quad (7.8)$$

Poniżej przedstawiono sposób wyznaczania początkowych wartości odpowiedzi skokowej elementu $B(z^{-1})/A(z^{-1})$ oraz wektora \hat{y} , który wykorzystano do implementacji sterownika. Dla ułatwienia przyjęto mały horyzont predykcji $H = 3$. Macierz Q prezentuje się następująco:

$$Q = \begin{bmatrix} h_0 & 0 & 0 \\ h_1 & h_0 & 0 \\ h_2 & h_1 & h_0 \end{bmatrix}, \quad (7.9)$$

gdzie:

$$h_0 = b_0, \quad (7.10)$$

$$h_1 = -a_1 h_0 + b_0, \quad (7.11)$$

$$h_2 = -a_1 h_1 - a_2 h_0 + b_0. \quad (7.12)$$

Wektor $\hat{\mathbf{y}}$ jest przewidywaną odpowiedzią obiektu, zakładając, że przyszłe wartości sterowań są równe $u(t-1)$ oraz, że zakłócenia są stałe, więc rozpisując równanie modelu wyznaczamy wartość sygnału wyjściowego w chwili :

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) + b_0 u(t-1). \quad (7.13)$$

Na podstawie równania (7.13) dokonujemy predykcji jednokrokowej:

$$\hat{y}(t+1) = -a_1 y(t) - a_2 y(t-1) + b_0 u(t). \quad (7.14)$$

Jeżeli od równania (7.14) odejmiemy równanie (7.13), otrzymamy:

$$\hat{y}(t+1) = (1 - a_1)y(t) + (a_1 - a_2)y(t-1) + a_2 y(t-2) + b_0 \Delta u(t), \quad (7.15)$$

gdzie:

$$\Delta u(t) = u(t) - u(t-1). \quad (7.16)$$

Następnie dokonujemy kolejne predykcje jednokrokowe:

$$\hat{y}(t+2) = (1 - a_1)\hat{y}(t+1) + (a_1 - a_2)y(t) + a_2 y(t-1) + b_0 \Delta u(t+1), \quad (7.17)$$

$$\hat{y}(t+3) = (1 - a_1)\hat{y}(t+2) + (a_1 - a_2)\hat{y}(t+1) + a_2 y(t) + b_0 \Delta u(t+2). \quad (7.18)$$

Zauważmy, że wartości $\Delta u(t)$, $\Delta u(t+1)$ i $\Delta u(t+2)$ są równe 0, co wynika z definicji wektora $\hat{\mathbf{y}}$:

$$u(t+2) = u(t+1) = u(t) = u(t-1). \quad (7.19)$$

Wektor $\hat{\mathbf{y}}$ dla horyzontu predykcji $h = 3$ można zapisać następująco:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(t+1) \\ \hat{y}(t+2) \\ \hat{y}(t+3) \end{bmatrix}. \quad (7.20)$$

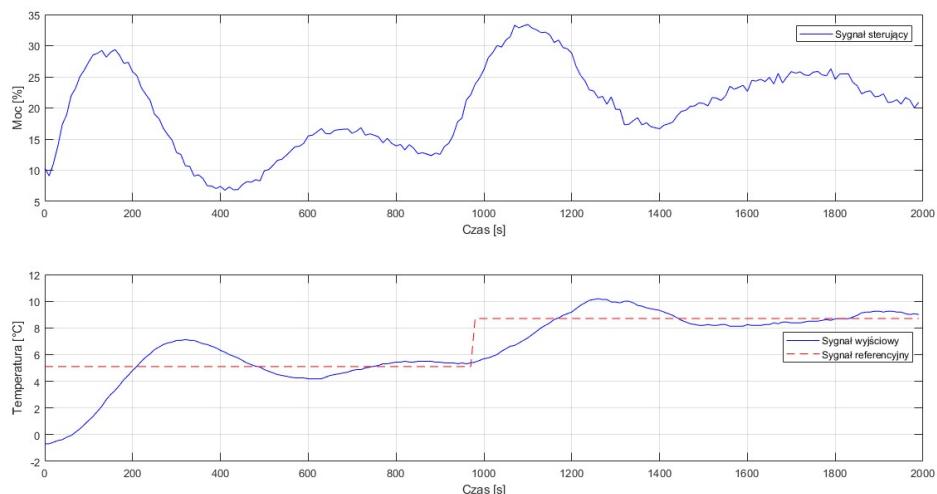
Łatwo można zauważyc, że zarówno wyznaczanie macierzy Q jak i wektora $\hat{\mathbf{y}}$ może odbywać się rekurencyjnie, więc algorytm zaimplementowano tak, aby możliwa była zmiana horyzontu predykcji.

W klasycznym sterowniku GPC, gdzie przyjęte wartości parametrów modelu obiektu są niezmienne, macierz Q i wektor $\hat{\mathbf{y}}$ wyznacza się tylko raz. Natomiast w sterowniku adaptacyjnym, zmienne te należy wyznaczać wtedy, gdy następuje aktualizacja parametrów.

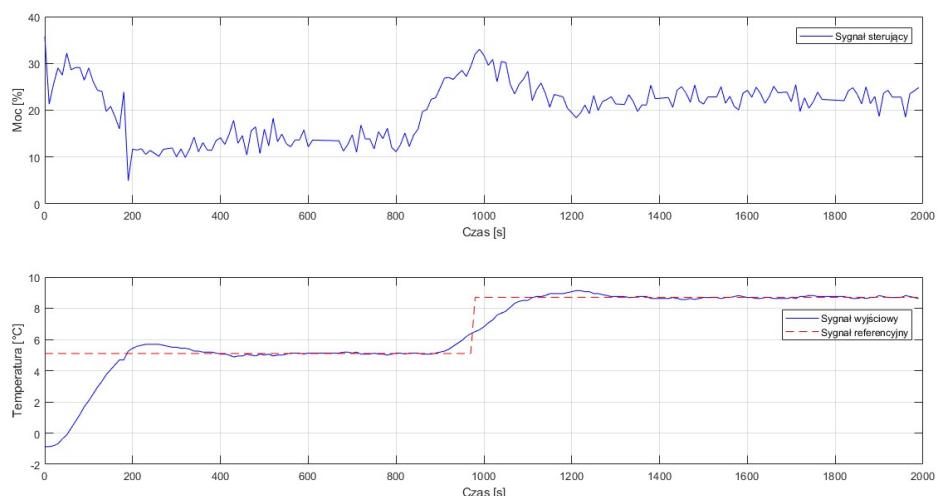
Sygnal sterujący, podawany na wejście regulatora PWM, wyrażany jest w procentach. Jego wartości ograniczono w przedziale od 0% do 90%.

7.1.1. Horyzont predykcji

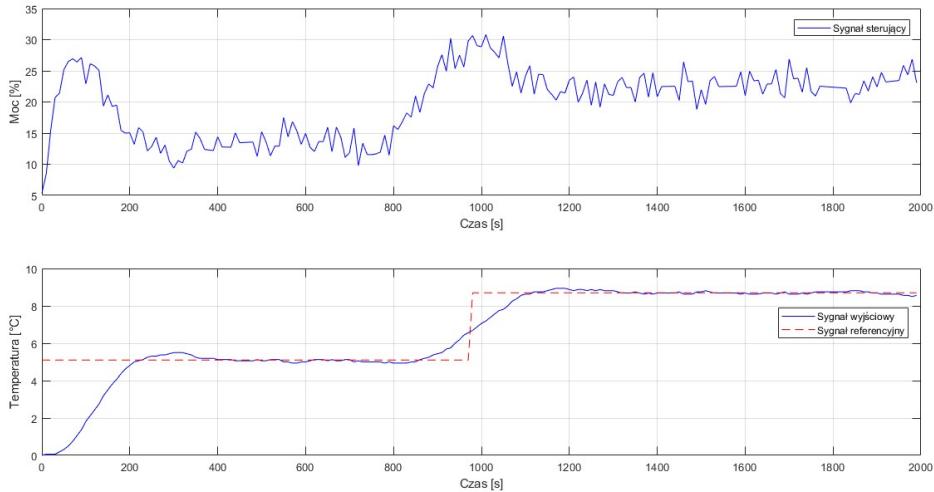
Odpowiedni dobór horyzontu predykcji pozwoli na uzyskanie lepszej jakości regulacji, dlatego też zdecydowano się zbadać, jak zachowuje się sterownik dla jego różnych wartości. Sprawdzono wpływ zwiększania horyzontu predykcji na szybkość ustalania się odpowiedzi skokowej obiektu oraz na wielkość przeregulowania. Wykonano trzy serie badań, dla horyzontu predykcji h wynoszącego: 10, 20 i 30. Okres próbkowania wynosił 10 s. Współczynnik wagowy $\varrho = 0,7$. Wyniki przedstawiono na wykresach (rys. 7.1 – 7.3).



Rys. 7.1. Sygnał sterujący i odpowiedź systemu dla horyzontu sterowania $H = 10$



Rys. 7.2. Sygnał sterujący i odpowiedź systemu dla horyzontu sterowania $H = 20$



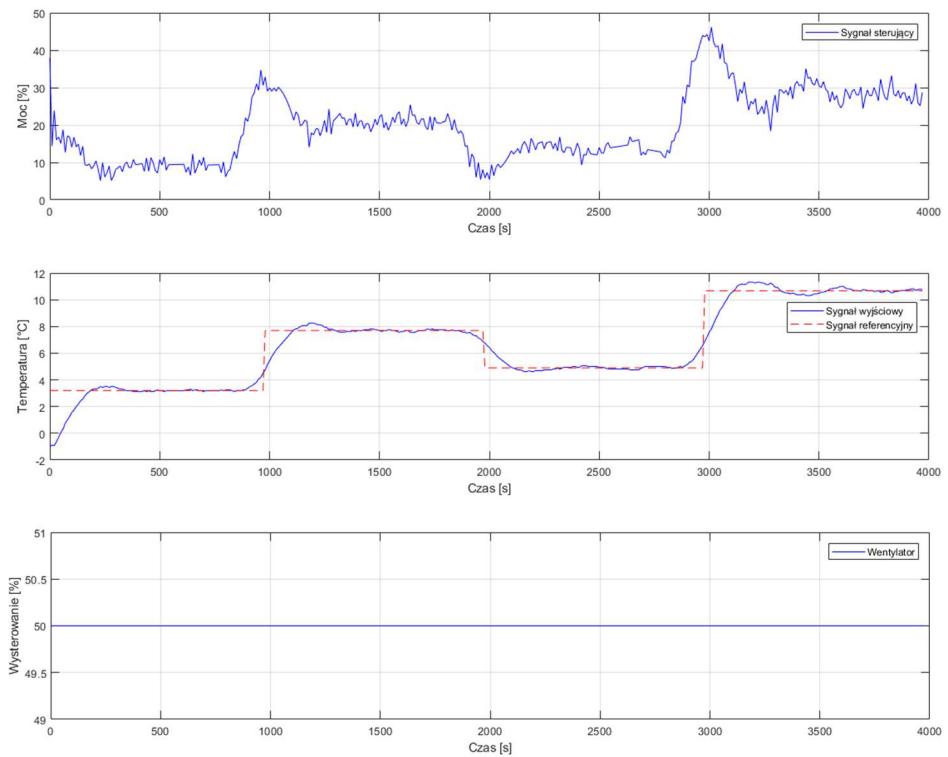
Rys. 7.3. Sygnał sterujący i odpowiedź systemu dla horyzontu sterowania $H = 30$

Na podstawie zebranych danych można stwierdzić, że ogólna jakość regulacji polepsza się wraz ze wzrostem horyzontu predykcji. Dla $H = 10$ widoczne jest przeregulowanie sięgające lekko ponad 39% wartości zadanej. Dla porównania, gdy $H = 20$ przeregulowanie wynosi tylko 11,5% a dla $H = 30$ jest na poziomie 7,7%. Warto zauważyć też, że obiekt, dla zbyt małego horyzontu predykcji, ma dłuższy czas ustalania się odpowiedzi skokowej, ponieważ występują lekkie (ale zanikające) oscylacje wokół wartości zadanej.

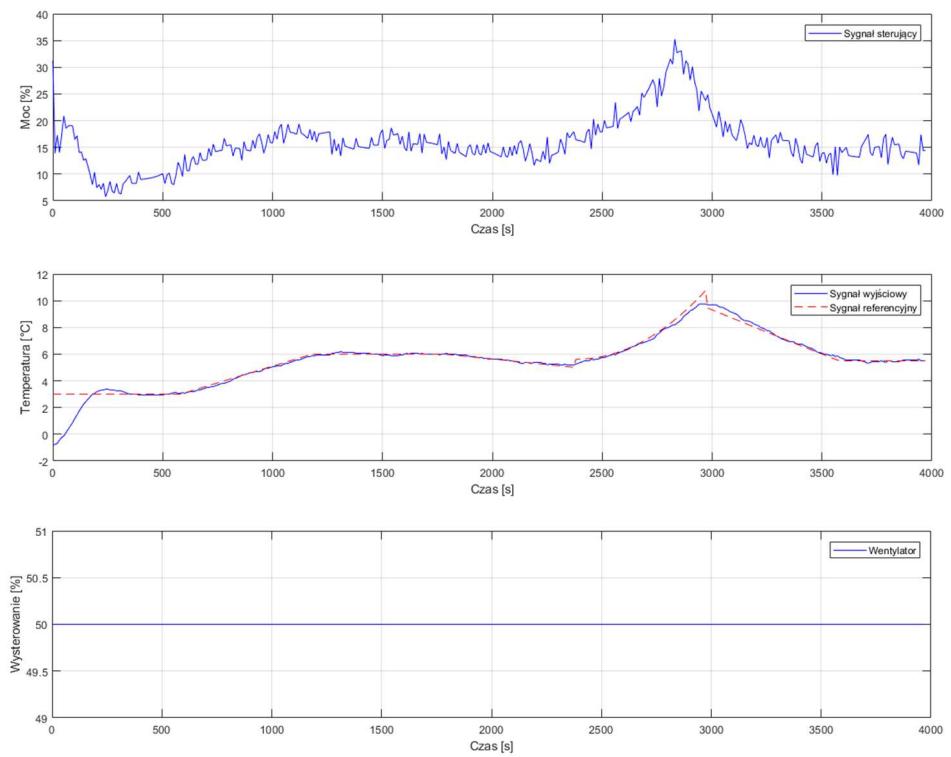
Jedną z zalet małego horyzontu predykcji jest to, że sygnał sterujący jest łagodniejszy, co może wydłużyć pracę elementów wykonawczych systemu. Dodatkowo nie obciąża on tak jednostki sterującej ze względu na małe rozmiary macierzy i wektorów w algorytmie regulatora.

7.1.2. Sterowanie w warunkach stałych

Celem tego eksperymentu jest sprawdzenie jakości regulacji sterownika dla jego określonych, niezmiennych warunków operacyjnych, zadając różną trajektorię referencyjną. Eksperyment wykonano dla takiej prędkości obrotowej wentylatora, dla jakiej uzyskano parametry modelu z równań (7.6) – (7.8). Skorzystano z modelu CARIMA opisanego równaniami (7.1) – (7.8). Horyzont predykcji jest równy horyzontowi sterowania i wynosi $H = L = 20$. Współczynnik wagowy sterowania $\varrho = 0,7$. Okres próbkowania wynosił 10 s. Poniżej zaprezentowano wyniki eksperymentu dla dwóch trajektorii referencyjnych: sygnał zadany przedziałami stały (rys. 7.4), sygnał zadany przedziałami zmienny (rys. 7.5).



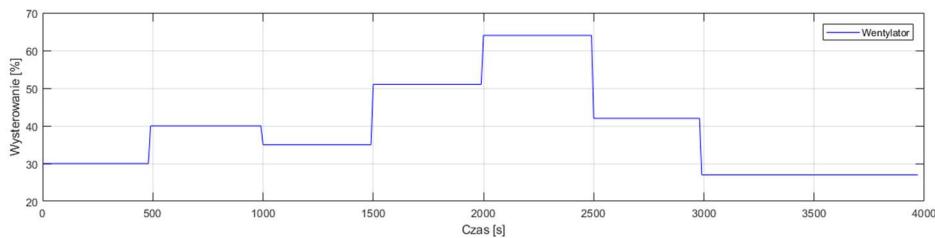
Rys. 7.4. Sygnał sterujący oraz odpowiedź systemu na pierwszą trajektorię zadaną (stała prędkość wentylatora)



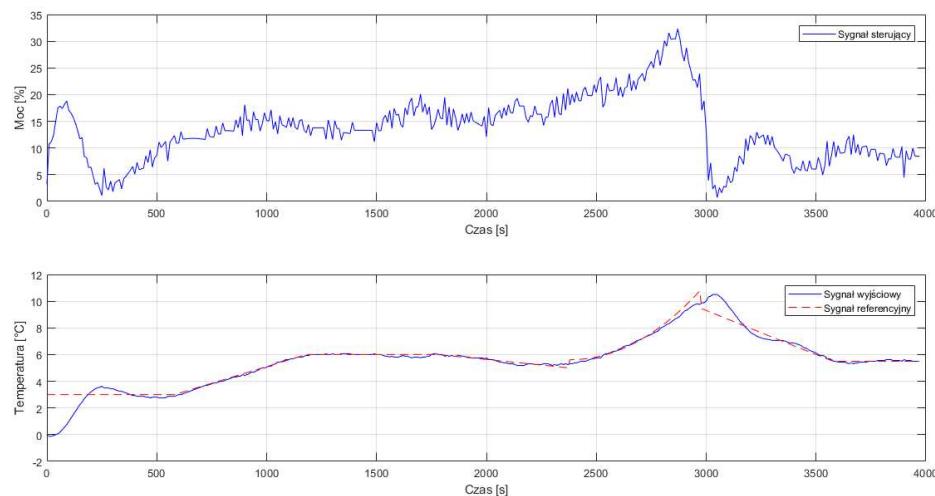
Rys. 7.5. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną (stała prędkość wentylatora)

7.1.3. Sterowanie w warunkach zmiennych

Sterowanie przy zmiennej prędkości obrotowej wentylatora zmienia punkt pracy, ponieważ zmieniają się parametry modelu obiektu. Wykonano te same badania, co w poprzednim podrozdziale z prędkością wentylatora zmieniającą się skokowo co 50 próbek. Parametry startowe algorytmu zostały niezmienione. Wyniki przedstawiono na wykresach (rys. 7.6 – 7.7).

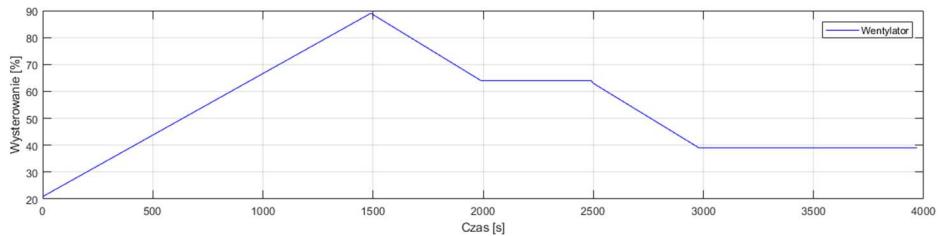


Rys. 7.6. Zmieniający się sygnał sterujący prędkością obrotową wentylatora (skokowa zmiana)

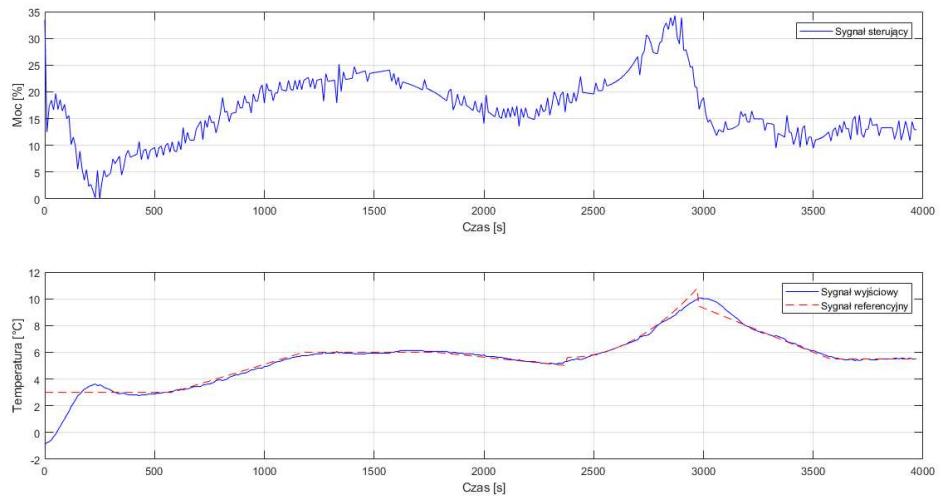


Rys. 7.7. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną dla skokowej zmiany sygnału sterującego prędkością obrotową wentylatora

Badanie powtórzono, wprowadzając modyfikację w sterowaniu prędkością obrotową wentylatora. Zaimplementowano funkcję, która w określonych przedziałach jest rosnąca bądź malejąca (rys. 7.8). Wyniki przedstawiono na wykresach (rys. 7.9).



Rys. 7.8. Zmieniający się sygnał sterujący prędkością obrotową wentylatora (płynna zmiana)



Rys. 7.9. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną dla płynnie zmieniającego się sygnału sterującego prędkością obrotową wentylatora

7.1.4. Wnioski

Nie zaobserwowano znaczącego pogorszenia jakości regulacji przy zmianie prędkości obrotowej wentylatora. Po upływie 3000 s na obu wykresach odpowiedzi systemu (rys. 7.7 i rys. 7.9) widoczne jest przeregulowanie, które jest większe niż w przypadku, gdy prędkość przepływu powietrza się nie zmienia (porównanie z rys. 7.5). Pojawia się ono akurat przy znaczającej obniżce prędkości przepływu powietrza. Mimo, że wartość sygnału sterującego spadła niemalże do zera, temperatura na wyjściu jeszcze przez pewien czas rośnie. Wynika to z konstrukcji obiektu, ponieważ sterownik ma wpływ tylko na podwyższanie jego temperatury, natomiast nie ma możliwości jego schładzania.

7.2. Self-Tuning Generalized Predictive Control

W niniejszej pracy zaimplementowano adaptacyjną wersję sterownika predykcyjnego o nazwie ST-GPC (ang. Self-Tuning Generalized Predictive Control). Składa się on z połączenia algorytmów: identyfikacji obiektu metodą RLS oraz klasycznego sterownika GPC. Sterownik ten należy do klasy regulatorów samonastrajalnych.

Otrzymane parametry w wyniku procesu identyfikacji wykorzystywane są na bieżąco do budowy modelu sterownika GPC. Oznacza to, że w każdej iteracji algorytmu następuje aktualizacja parametrów modelu obiektu, który wykorzystywany jest później do syntezy

sterownika. W każdym kroku należy wyznaczyć na nowo elementy macierzy Q oraz wektor \hat{y} , ponieważ zależą one ściśle od współczynników modelu obiektu.

Inicjalizacja algorytmu sterownika wymaga przyjęcia początkowych wartości wektora parametrów $\theta(0)$ oraz wektora regresji $\varphi^T(0)$. Jako wektor parametrów w chwili $t = 0$ przyjęto wartości współczynników uzyskane z eksperymentu identyfikacji dla modelu $ARX(2,1)$, tj.:

$$\theta(0) = [a_1, a_2, b_0]^T, \quad (7.21)$$

gdzie:

$$a_1 = -1,75, \quad (7.22)$$

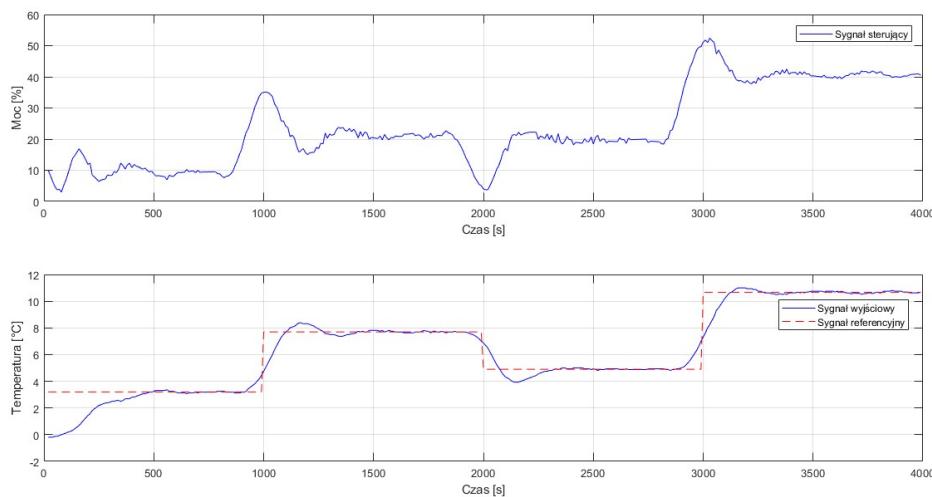
$$a_2 = 0,7599, \quad (7.23)$$

$$b_0 = 0,0036. \quad (7.24)$$

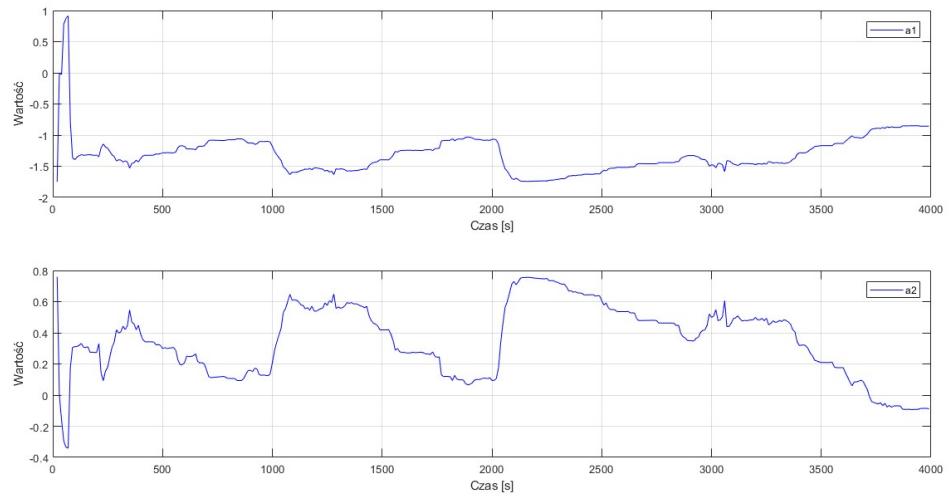
Wypełnienie wektora regresji osiągane jest poprzez opóźnienie załączenia pętli sterownika o dwa okresy próbkowania.

7.2.1. Sterowanie w warunkach zmiennych

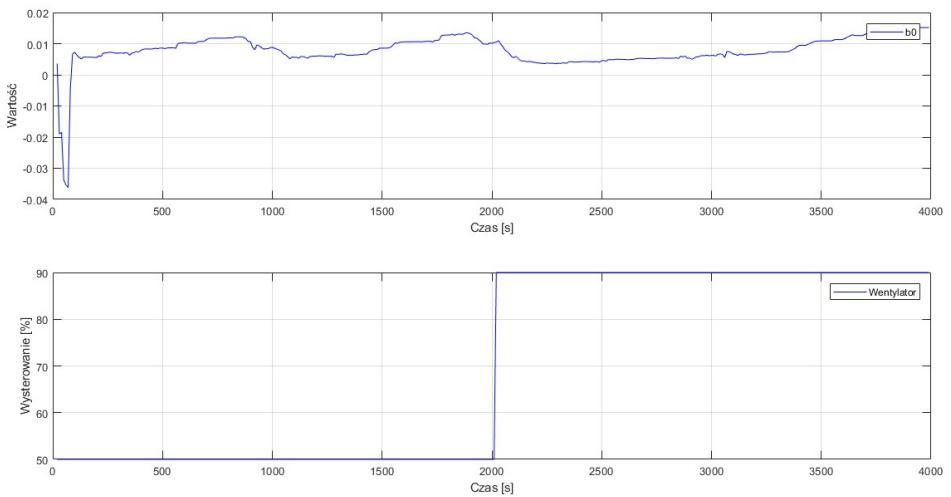
Zbadano działanie sterownika ST-GPC na trajektoriach referencyjnych wykorzystanych w poprzednim podrozdziale. Współczynnik zapominania $\lambda = 0,97$, horyzont predykcji $H = 20$ a współczynnik wagowy $\varrho = 0,7$. Okres próbkowania wynosił 10 s. Wykres sygnału sterującego oraz odpowiedzi układu zamkniętego przedstawiono na rys. 7.10. Rys. 7.11 – 7.12 przedstawiają zmieniające się współczynniki modelu obiektu oraz sygnał sterujący prędkością obrotową wentylatora.



Rys. 7.10. Sygnał sterujący i odpowiedź systemu na pierwszą trajektorię zadaną

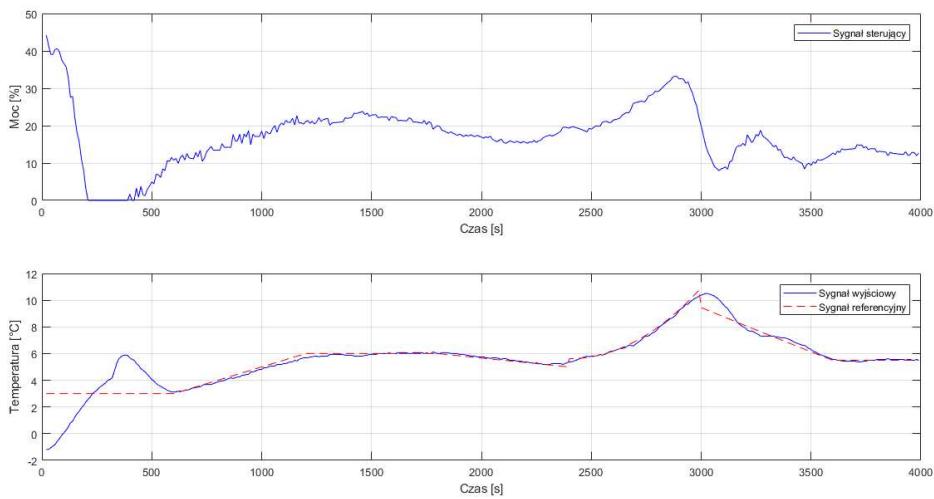


Rys. 7.11. Parametry modelu obiektu (a_1 i a_2)

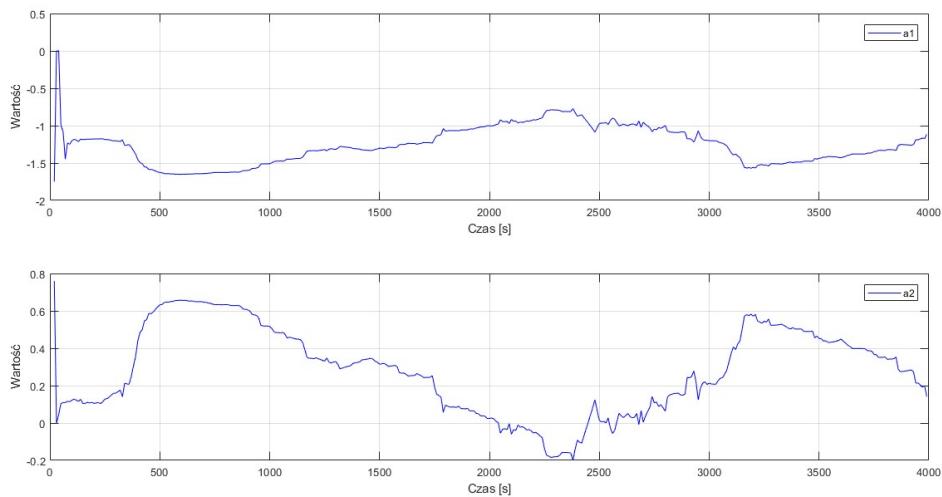


Rys. 7.12. Parametr modelu obiektu b_0 oraz sygnał sterujący prędkością obrotową wentylatora

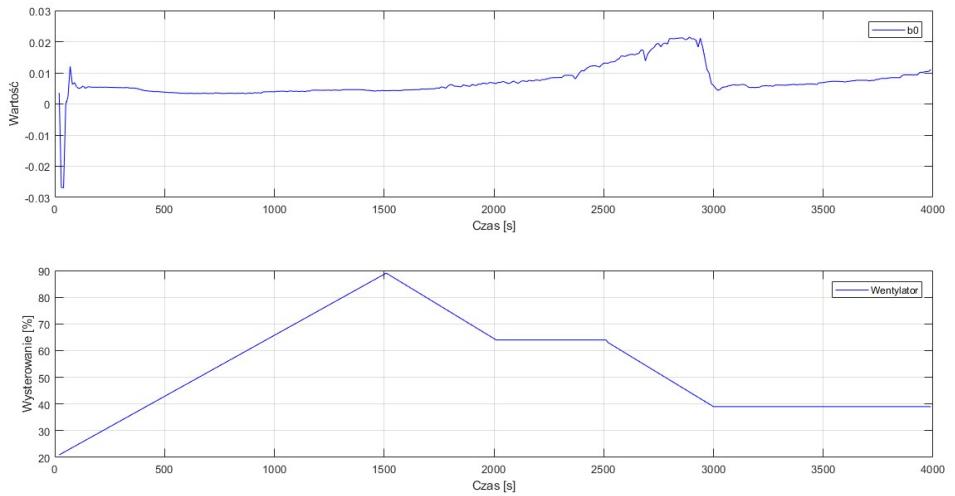
Eksperyment wykonano ponownie, zmieniając trajektorię referencyjną na bardziej płynną. Wprowadzono również modyfikację sygnału sterującego prędkością obrotową wentylatora. Wyniki zaprezentowano na wykresach (rys. 7.13 – 7.15).



Rys. 7.13. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną



Rys. 7.14. Parametry modelu obiektu (a_1 i a_2)



Rys. 7.15. Parametr modelu obiektu b_0 oraz sygnał sterujący prędkością obrotową wentylatora

7.2.2. Wnioski

W adaptacyjnym sterowniku predykcyjnym zaobserwowano nieco zwiększone przeregulowania przy skokowych zmianach sygnału referencyjnego niż w przypadku klasycznego sterownika predykcyjnego. Dodatkowo, na początku regulacji, sygnał wyjściowy bardzo mocno odbiega od sygnału zadanego. Dzieje się to podczas procesu ustalania się parametrów modelu. Do syntezy sterownika dostarczany jest więc jeszcze nieodpowiedni model obiektu regulacji.

Parametry modelu ARX zmieniają się pod wpływem zmian prędkości obrotowej wentylatora. Największe zmiany zachodzą jednak podczas ochładzania się obiektu.

W porównaniu ze sterownikiem GPC, sterownik samonastrajalny wyznacza sygnał sterujący o znacznie mniejszej wariancji. Właściwość zaobserwowano dla obu trajektorii zadanych, niezależnie od zmian sygnału sterującego prędkością obrotową wentylatora.

8. PODSUMOWANIE

Odpowiednia koncepcja projektu umożliwiła zbudowanie stanowiska laboratoryjnego pozwalającego na zbadanie wpływu jakości regulacji algorytmów sterowania w zależności od zmiennego opóźnienia transportowego.

Sterownik, oparty na algorytmie GPC, wykazuje bardzo dobrą jakość regulacji dla obiektów niestabilnych, o zmieniających się parametrach modelu oraz z nieznanym lub ulegającym zmianie opóźnieniu transportowym, co sprawia, że może być wykorzystany w systemach klimatyzacji lub wentylacji. Niniejsza praca magisterska stanowi przykład implementacji sterownika GPC na obiekcie o bardzo małym opóźnieniu transportowym. W systemach zarządzających przepływem powietrza w budynkach opóźnienie jest znacznie większe (a także jego zmiana jest większa) i implementacja sterownika predykcyjnego jest w tych przypadkach uzasadniona. W dalszych etapach związanych z projektem należało wydłużyć kanał transmisji powietrza tak, aby opóźnienie transportowe bardziej odzwierciedlało realne warunki w systemach wentylacji.

Adaptacyjny regulator znakomicie sprawdza się w przypadku, gdy warunki operacyjne obiektu sterowania ulegają ciągłym zmianom. Algorytm identyfikacji pozwala wykryć zmianę punktu pracy i dzięki temu jakość regulacji cały czas utrzymywana jest na zadowalającym poziomie. Podczas eksperymentów identyfikacji obiektu zaobserwowano znaczącą zmianę parametrów modelu podczas zmniejszania się temperatury na wyjściu kanału. Wyłączenie algorytmu identyfikacji w tym momencie spowodowałoby dokładniejszą estymację parametrów modelu obiektu dla procesu nagrzewania, co przełożyłoby się na dokładniejsze śledzenie trajektorii zadanej.

Wszystkie cele oraz założenia projektowe zostały osiągnięte. Realizacja pracy magisterskiej umożliwiła poszerzenie wiedzy zawartej w programie studiów a także wykroczyła poza nią, wykorzystując zainteresowania autora.

WYKAZ LITERATURY

- [1] Åström K. J., Wittenmark B.: *Adaptive Control*, Addison–Weasley, Reading, Massachusetts, 1995.
- [2] Yong K. H.: *Adaptive–Model Based Self–Tuning Generalized Predictive Control of A Biodiesel Transesterification Reactor*, Monash University, Malaysia, 2011.
- [3] Trade Media International, <https://www.controlengineering.pl/sterowanie-adaptacyjne/>, (data dostępu 10.07.2019).
- [4] Söderström T., Stoica P.: *Identyfikacja systemów*, Wydawnictwo Naukowe PWN, Warszawa, 1997.
- [5] Camacho E. F., Bordons C.: *Model Predictive Control*, Springer publication, Londyn, 2007.
- [6] Holkar K. S., Waghmare L. M.: *An Overview of Model Predictive Control*, International Journal of Control and Automation, 2010, s. 53-54.
- [7] Richalet J., Rault A., Testud J. L., Papon J., *Model Predictive Heuristic Control: Applications to Industrial Processes*, Automatica, Vol. 14, pp. 413-428.
- [8] Strona internetowa Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej https://eia.pg.edu.pl/documents/184139/28438497/MPC_podstawy.pdf/, (data dostępu 10.07.2019).
- [9] Chidrawar S., Patre B.: *Generalized Predictive Control and Neural Generalized Predictive Control*, Leonardo Journal of Sciences, 2008.
- [10] Niederliński A., Mościński J., Ogonowski Z.: *Regulacja adaptacyjna*, Wydawnictwo Naukowe PWN, Warszawa, 1995, s. 273.
- [11] Szewczyk R., Zieliński C., Kaliczyńska M.: *Recent Advances in Automation, Robotics and Measuring Techniques*, Springer, s. 332.
- [12] The Raspberry Pi Foundation, <https://www.raspberrypi.org/documentation/hardware/>, (data dostępu 10.07.2019).
- [13] Sample Time is a Fundamental Design and Tuning Specification. Control Guru: Practical Process Control, [https://controlguru.com/sample-time-is-a-fundamental-design-and-tuning specification/](https://controlguru.com/sample-time-is-a-fundamental-design-and-tuning-specification/), (data dostępu 10.07.2019).

WYKAZ RYSUNKÓW

2.1. Schemat blokowy sterownika adaptacyjnego.....	11
2.2. Schemat blokowy systemu harmonogramowania wzmacnienia.....	12
2.3. Zbiornik o niejednolitym przekroju poprzecznym.....	13
2.4. Schemat blokowy przedstawiający strukturę MRAS.....	14
2.5. Idea regulatora samonastrajalnego.....	15
4.1. Idea sterowania predykcyjnego.....	26
4.2. Struktura sterownika predykcyjnego.....	26
4.3. Podstawowa struktura sterownika GPC.....	29
4.4. Symulacja sygnałów wyjściowych modelu dla algorytmu sterownika o parametrach: $H = 7$ (górnny wykres) i $H = 30$ (dolny wykres). Współczynnik $\varrho = 1$	32
4.5. Symulacja sygnałów wyjściowych modelu dla algorytmu sterownika o parametrach: $\varrho = 5$ (górnny wykres) i $\varrho = 0,01$ (dolny wykres). Horyzont predykcji $H = 30$	32
4.6. Symulacja sygnałów sterujących modelu dla algorytmu sterownika o parametrach: $\varrho = 5$ (górnny wykres) i $\varrho = 0,01$ (dolny wykres). Horyzont predykcji $H = 30$	33
5.1. Schemat stanowiska.....	35
5.2. Grzałka ceramiczna.....	36
5.3. Blok grzejny E3D V6 o wymiarach 20x16x12 mm.....	37
5.4. Schemat kanału do przepływu ciepłego powietrza.....	37
5.5. Wentylator.....	38
5.6. Nagrzewnica powietrza: 1 – wentylator, 2 – mocowanie grzałki, 3 – czujnik temperatury.....	39
5.7. Mocowanie czujnika temperatury.....	39
5.8. Zasilacz modułowy.....	40
5.9. Komputer Raspberry Pi 3 Model B.....	41
5.10. Schemat elektryczny modułu GPIO.....	42
5.11. Czujnik temperatury DS18B20.....	43
5.12. Sonda pomiarowa z czujnikiem temperatury DS18B20.....	43
5.13. Moduł sterownika PWM.....	44
6.1. Charakterystyka obiektu, od góry: sygnał sterujący, sygnał wyjściowy, temperatura powietrza w pokoju.....	46
6.2. Sygnał sterujący.....	47
6.3. Estymowane parametry modelu $ARX(1,1)$	48
6.4. Błędy resztowe modelu $ARX(1,1)$	48
6.5. Porównanie sygnałów: wyjściowego obiektu rzeczywistego i wyjściowego modelu $ARX(1,1)$	49
6.6. Estymowane parametry modelu $ARX(2,1)$	50
6.7. Błędy resztowe modelu $ARX(2,1)$	50
6.8. Porównanie sygnałów: wyjściowego obiektu rzeczywistego i wyjściowego modelu $ARX(2,1)$	51
6.9. Estymowane parametry modelu $ARX(3,1)$	52
6.10. Błędy resztowe modelu $ARX(3,1)$	52
6.11. Porównanie sygnałów: wyjściowego obiektu rzeczywistego i wyjściowego modelu $ARX(3,1)$..	53
6.12. Zestaw danych do walidacji systemu. Okres próbkowania wynosi 10 s.....	54

6.13. Sygnały: wejściowy i wyjściowy obiektu. Okres próbkowania wynosi 10 s.....	55
6.14. Zmieniający się parametr modelu b_o obiektu oraz wysterowanie prędkości obrotowej wentylatora. Współczynnik zapominania $\lambda = 0,99$	55
6.15. Zmieniające się parametry modelu (a_1 i a_2) obiektu. Współczynnik zapominania $\lambda = 0,99$	56
6.16. Sygnały: wejściowy i wyjściowy obiektu. Okres próbkowania wynosi 10 s.....	56
6.17. Zmieniający się parametr modelu b_o obiektu oraz wysterowanie prędkości obrotowej wentylatora. Współczynnik zapominania $\lambda = 0,97$	57
6.18. Zmieniające się parametry modelu (a_1 i a_2) obiektu. Współczynnik zapominania $\lambda = 0,99$	57
7.1. Sygnał sterujący i odpowiedź systemu dla horyzontu sterowania $H = 10$	61
7.2. Sygnał sterujący i odpowiedź systemu dla horyzontu sterowania $H = 20$	61
7.3. Sygnał sterujący i odpowiedź systemu dla horyzontu sterowania $H = 30$	62
7.4. Sygnał sterujący oraz odpowiedź systemu na pierwszą trajektorię zadaną (stała prędkość wentylatora).....	63
7.5. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną (stała prędkość wentylatora).....	63
7.6. Zmieniający się sygnał sterujący prędkością obrotową wentylatora (skokowa zmiana).....	64
7.7. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną dla skokowej zmiany sygnału sterującego prędkością obrotową wentylatora.....	64
7.8. Zmieniający się sygnał sterujący prędkością obrotową wentylatora (płynna zmiana).....	65
7.9. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną dla płynnie zmieniającego się sygnału sterującego prędkością obrotową wentylatora.....	65
7.10. Sygnał sterujący i odpowiedź systemu na pierwszą trajektorię zadaną.....	66
7.11. Parametry modelu obiektu (a_1 i a_2).....	67
7.12. Parametr modelu obiektu b_o oraz sygnał sterujący prędkością obrotową wentylatora.....	67
7.13. Sygnał sterujący i odpowiedź systemu na drugą trajektorię zadaną.....	68
7.14. Parametry modelu obiektu (a_1 i a_2).....	68
7.15. Parametr modelu obiektu b_o oraz sygnał sterujący prędkością obrotową wentylatora.....	69

WYKAZ TABEL

5.1. Parametry techniczne grzałki ceramicznej.....	36
5.2. Dane techniczne wentylatora.....	38
5.3. Dane techniczne zasilacza modułowego.....	40
5.4. Dane techniczne czujnika temperatury DS18B20.....	43
6.1. Ustalone wartości parametrów modeli ARX obiektu rzeczywistego.....	53
6.2. Wyniki zastosowania kryterium informacyjnego Akaike'go do walidacji danych.....	54
6.3. Zmieniające się parametry modelu obiektu w zależności od nastaw regulatora prędkości obrotowej wentylatora. Przedstawione w tabeli wartości są ostatnimi współczynnikami modelu tuż przed zmianą prędkości. Współczynnik zapominania $\lambda = 0,99$	55
6.4. Zmieniające się parametry modelu obiektu w zależności od nastaw regulatora prędkości obrotowej wentylatora. Przedstawione w tabeli wartości są ostatnimi współczynnikami modelu tuż przed zmianą prędkości. Współczynnik zapominania $\lambda = 0,97$	56

Dodatek A: Algorytm identyfikacji obiektu EF-RLS

```
1 import time
2 import csv
3 import sys
4 import numpy as np
5 import RPi.GPIO as GPIO
6 from w1thermsensor import W1ThermSensor
7
8 # Inicjalizacja modułu GPIO
9 GPIO.setwarnings(False)
10 GPIO.setmode(GPIO.BCM)
11 GPIO.setup(12, GPIO.OUT)
12 GPIO.setup(13, GPIO.OUT)
13 Heater = GPIO.PWM(12, 50) # 1 kanał PWM - grzałka
14 Fan = GPIO.PWM(13, 50) # 2 kanał PWM - wentylator
15 Sensor = W1ThermSensor.get_available_sensors() # tablica z ID czujników temperatury
16
17 ma = 2 # liczba estymowanych parametrów wielomianu A
18 mb = 1 # liczba estymowanych parametrów wielomianu B
19 Ts = 10 # okres próbkowania
20 lam = 0.97 # współczynnik zapominania lambda
21 L = 300 # liczba próbek
22 P = 100000*np.eye(ma+mb) # początkowa wartość macierzy P
23 phi = np.zeros((ma+mb, 1), np.float32) # wektor regresji
24 theta = np.zeros((ma+mb, 1), np.float32) # wektor parametrów
25 U = np.zeros((1, L), np.float32) # sygnał sterujący
26 Y = np.zeros((1, L), np.float32) # sygnał wyjściowy
27 x1 = 50 # ilość próbek po jakiej następuje zmiana prędkości wentylatora
28 x2 = 15 # ilość próbek po jakiej następuje zmiana sygnału sterującego
29 e = 0 # początkowa wartość chwilowego błędu resztowego
30
31 # Pomiar temperatury na wyjściu kanału oraz temperatury powietrza
32 def Temperature_Measurement():
33     Air_Temp = Sensor[0].get_temperature() # temperatura powietrza
34     Heater_Temp = Sensor[1].get_temperature() # temperatura na wyjściu kanału
35     Diff_Temp = np.float32(round((Heater_Temp - Air_Temp), 3)) # różnica
36     return Air_Temp, Diff_Temp
37
38 # Inicjalizacja regulatorów PWM
39 def Init():
40     global Power # % wartość mocy grzałki
41     global Speed # % wartość prędkości obrotowej wentylator
42     Power = np.float32(27)
43     Speed = np.float32(40)
44     Heater.start(Power)
45     Fan.start(Speed)
46
47 # Schładzanie obiektu
48 def Heater_Cooling():
49     Heater.ChangeDutyCycle(0)
50     Fan.ChangeDutyCycle(100)
51     while True:
52         [Air_Temp, Diff_Temp] = Temperature_Measurement()
53         if Diff_Temp > 0.2:
54             [Air_Temp, Diff_Temp] = Temperature_Measurement()
55             print("Cooling...\n", Diff_Temp)
56             time.sleep(5)
57         else:
58             GPIO.cleanup()
59             sys.exit()
60
61 try:
62     Init()
63     print(" Fan:", Speed, " Ts:", Ts, " Lambda:", lam)
64     for t in range(0, L):
65         start = time.time()
```

```

66 [Air_Temp, Diff_Temp] = Temperature_Measurement()
67 Y[0, t] = Diff_Temp
68 U[0, t] = Power
69
70 # Zmiana prędkości obrotowej wentylatora i sygnału sterującego
71 if (t == x1):
72     Speed = 60
73     Fan.ChangeDutyCycle(Speed)
74 if (t == 2*x1):
75     Speed = 20
76     Fan.ChangeDutyCycle(Speed)
77 if (t == 3*x1):
78     Speed = 80
79     Fan.ChangeDutyCycle(Speed)
80 if (t == 4*x1):
81     Speed = 40
82     Fan.ChangeDutyCycle(Speed)
83 if (t == 5*x1):
84     Speed = 100
85     Fan.ChangeDutyCycle(Speed)
86 if (t == x2):
87     x2 = np.int(np.random.rand(1)[0] * 20 + 5)
88     x2 = t + x2
89     Power = np.round(np.float32(np.random.rand(1)[0] * 60 + 10), 1)
90     Heater.ChangeDutyCycle(Power)
91 U[0, t] = Power
92
93 # Algorytm VF-RLS
94 while t > ma:
95     phi = np.array([[ -Y[0, t-1]], [-Y[0, t-2]], [U[0, t-1]]])
96     e = Y[0, t] - phi.T @ theta
97     L = P @ phi / (lam + phi.T @ P @ phi)
98     P = (P - L @ phi.T @ P) / lam
99     theta = theta + L * e
100    break
101
102 # Zapis do pliku .csv
103 with open('Online_RLS_097.csv', 'a') as csvFile:
104     data = csv.writer(csvFile)
105     data.writerow([t, U[0, t], Y[0, t], theta[0], theta[1], theta[2], e, Air_Temp, Speed, Ts, lam])
106     csvFile.close()
107 print("t:", t, " U(t):", U[0, t], " Y(t):", Y[0, t], " Speed(t):", Speed, " a1(t):", theta[0],
108 " a2(t):", theta[1], " b0(t):", theta[2])
109 end = time.time()
110 time.sleep(Ts - end + start) # ujednolicenie okresu próbkowania
111 Heater_Cooling()
112
113 except(KeyboardInterrupt):
114     Heater_Cooling()

```

Dodatek B: Algorytm sterownika GPC

```
1 import numpy as np
2 import time
3 import csv
4 import sys
5 import matplotlib.pyplot as plt
6 import RPi.GPIO as GPIO
7 from numpy.linalg import inv
8 from w1thermsensor import W1ThermSensor
9 from Reference import Gentle_Reference
10 from Reference import Rectangle_Reference
11 from Speed import Gentle_Speed
12 from Speed import Rectangle_Speed
13
14 # Inicjalizacja modułu GPIO
15 GPIO.setwarnings(False)
16 GPIO.setmode(GPIO.BCM)
17 GPIO.setup(12, GPIO.OUT)
18 GPIO.setup(13, GPIO.OUT)
19 Heater = GPIO.PWM(12, 50) # 1 kanał PWM - grzałka
20 Fan = GPIO.PWM(13, 50) # 2 kanał PWM - wentylator
21 Sensor = W1ThermSensor.get_available_sensors() # tablica z ID czujników temperatury
22
23 a = np.array([-1.7500, 0.7599], np.float32) # parametry wielomianu A
24 b = np.float32(0.0036) # parametry wielomianu B
25 na = 2 # rząd wielomianu A
26 H = 20 # horyzont predykcji/sterowania
27 L = 300 # liczba próbek
28 Ts = 10 # okres próbkowania
29
30 h = np.zeros((1, H), np.float32) # elementy macierzy Q
31 Q = np.zeros((2*H, 2*H), np.float32) # macierz Q
32 Y = np.zeros((1, L+H), np.float32) # sygnał wyjściowy
33 U = np.zeros((1, L+H), np.float32) # sygnał sterujący
34 w = np.zeros((L+H, 1), np.float32) # trajektoria referencyjna
35 # w = Gentle_Reference(L,H) # gładka trajektoria referencyjna
36 w = Rectangle_Reference(L,H) # skokowa trajektoria referencyjna
37 V = Gentle_Speed(L,H) # płynnie zmieniająca się prędkość obrotowa wentylatora
38 # V = Rectangle_Speed(L,H) # skokowo zmieniająca się prędkość obrotowa wentylatora
39 f = np.zeros((1, H), np.float32) # odpowiedź swobodna obiektu
40 Umin = 0 # minimalna wartość sygnału sterującego
41 Umax = 90 # maksymalna wartość sygnału sterującego
42 ro = 0.7 # współczynnik wagowy przyrostu sterowania
43
44 # Pomiar temperatury na wyjściu kanału oraz temperatury powietrza
45 def Temperature_Measurement():
46     Air_Temp = Sensor[0].get_temperature() # temperatura powietrza
47     Heater_Temp = Sensor[1].get_temperature() # temperatura na wyjściu kanału
48     Diff_Temp = np.float32(np.round((Heater_Temp - Air_Temp), 3)) # różnica
49     return Air_Temp, Diff_Temp
50
51 # Inicjalizacja regulatorów PWM
52 def Init():
53     global Power # % wartość mocy grzałki
54     global Speed # % wartość prędkości obrotowej wentylatora
55     Power = np.float32(0)
56     Speed = np.float32(50)
57     Heater.start(Power)
58     Fan.start(Speed)
59
60 # Schładzanie obiektu
61 def Heater_Cooling():
62     Heater.ChangeDutyCycle(0)
63     Fan.ChangeDutyCycle(100)
64     while True:
65         [Air_Temp, Diff_Temp] = Temperature_Measurement()
```

```

66     if Diff_Temp > 0.2:
67         [Air_Temp, Diff_Temp] = Temperature_Measurement()
68         print("Cooling...\n", Diff_Temp)
69         time.sleep(5)
70     else:
71         GPIO.cleanup()
72         sys.exit()
73
74 # Wyznaczanie macierzy Q
75 h[0, 0] = b
76 h[0, 1] = -a[1] * h[0, 0] + b
77 for i in range(2, H):
78     h[0, i] = -a[1] * h[0, i-1] - a[2] * h[0, i-2] + b
79
80 for j in range(0, H):
81     for i in range(0, H):
82         Q[j+i-1, i] = h[0, j]
83 Q = Q[0:H, 0:H]
84
85 try:
86     Init()
87     print(" Fan:", Speed, " Ts:", Ts, " H:", H, " ro:", ro)
88     for t in range(na, L+na):
89         start = time.time()
90         [Air_Temp, Y[0, t]] = Temperature_Measurement()
91
92         # Wyznaczanie odpowiedzi swobodnej obiektu
93         f0 = Y[0, t]
94         f[0, 0] = (1 - a[1]) * f0 + (a[1] - a[2]) * Y[0, t-1] + a[2] * Y[0, t-2]
95         f[0, 1] = (1 - a[1]) * f[0, 0] + (a[1] - a[2]) * f0 + a[2] * Y[0, t-1]
96         f[0, 2] = (1 - a[1]) * f[0, 1] + (a[1] - a[2]) * f[0, 0] + a[2] * f0
97         for i in range(3,H):
98             f[0, i] = (1 - a[1]) * f[0, i-1] + (a[1] - a[2]) * f[0, i-2] + a[2] * f[0, i-3]
99
100        # Algorytm sterownika GPC
101        dU = inv(Q.T @ Q + ro * np.eye(H)) @ Q.T @ (w[t:t + H] - f.T)
102        U[0, t] = np.round((U[0, t - 1] + dU[0]), 3)
103
104        # Ograniczenia na wartość sygnału sterującego
105        if U[0, t] < Umin:
106            U[0, t] = Umin
107        elif U[0, t] > Umax:
108            U[0, t] = Umax
109        dU[0] = U[0, t] - U[0, t - 1]
110        Yp = np.round((Q @ dU + f.T), 3)
111        Fan.ChangeDutyCycle(V[0, t])
112        Heater.ChangeDutyCycle(U[0, t])
113
114        # Zapis do pliku .csv
115        with open('P_GPC.csv', 'a') as csvFile:
116            data = csv.writer(csvFile)
117            data.writerow([t-na, U[0, t], Y[0, t], Yp[0][0], w[t][0], Air_Temp, V[0, t], Ts, H, ro])
118            csvFile.close()
119        print("t: ", t-na, " U(t):", U[0,t], " Y(t):", Y[0,t], " Yp(t+1|t):", Yp[0], " w(t):", w[t], " V(t):", V[0, t])
120        end = time.time()
121        time.sleep(Ts - end + start) # ujednolicenie okresu próbkowania
122        Heater_Cooling()
123
124 except(KeyboardInterrupt):
125     Heater_Cooling()

```

Dodatek C: Zdjęcie stanowiska laboratoryjnego

