

POLITECHNIKA WARSZAWSKA

D Z I E D Z I N A N A U K
I N Ź Y N I E R Y J N O - T E C H N I C Z N Y C H

A U T O M A T Y K A , E L E K T R O N I K A
I E L E K T R O T E C H N I K A

Rozprawa doktorska

mgr inż. Rafał Szczepański

**Sterowanie adaptacyjne silnika PMSM o zmiennym momencie
bezwładności z wykorzystaniem inspirowanych przyrodą
algorytmów optymalizacyjnych i regulatora bazującego
na sprzężeniu od wektora zmiennych stanu**

Promotor
dr hab. inż. Tomasz Tarczewski

Promotor pomocniczy
dr inż. Łukasz Jan Niewiara

WARSZAWA 2022

Składam serdeczne podziękowania prof. dr. hab. inż. Lechowi M. Grzesiakowi, dr. hab. inż. Tomaszowi Tarczewskiemu oraz dr. inż. Łukaszowi J. Niewiara za cenne uwagi i poświęcony czas.

Streszczenie

Rozprawa doktorska dotyczy zagadnienia sterowania adaptacyjnego napędem elektrycznym z silnikiem PMSM o zmiennym momencie bezwładności z wykorzystaniem inspirowanych przyrodą algorytmów optymalizacyjnych. Opracowano autorską procedurę adaptacji dla algorytmów optymalizacyjnych, która umożliwia zastosowanie większości algorytmów optymalizacyjnych, w tym inspirowanych przyrodą, do adaptacji współczynników regulatora ze sprzężeniem zwrotnym od wektora zmiennych stanu. Zaproponowana procedura pozwala na zapewnienie niezmiennej odpowiedzi układu niezależnie od aktualnych warunków pracy napędu lub momentu bezwładności układu. Przedstawione wyniki badań symulacyjnych oraz eksperymentalnych wykazały, że zastosowanie opracowanej struktury regulacji adaptacyjnej pozwala na zapewnienie niezmiennej odpowiedzi silnika PMSM o zmiennym momencie bezwładności.

Słowa kluczowe: *napęd elektryczny z silnikiem PMSM, sterowanie adaptacyjne, sterowanie ze sprzężeniem od wektora zmiennych stanu, algorytm optymalizacyjny inspirowany przyrodą.*

Abstract

The doctoral dissertation deals with the problem of adaptive control of an electric drive with a PMSM motor with a variable moment of inertia with the use of optimization algorithms inspired by nature. A proprietary adaptation procedure was developed for optimization algorithms, which allows the use of most optimization algorithms, including those inspired by nature, to adapt the regulator coefficients with feedback from the vector of state variables. The proposed procedure allows to ensure a constant response of the system, regardless of the current operating conditions of the drive or the moment of inertia of the system. The presented results of simulation and experimental studies have shown that the application of the developed adaptive control structure allows to ensure a constant response of the PMSM motor with a variable moment of inertia.

Keywords: *PMSM drive, adaptive control, state feedback controller, nature-inspired optimization algorithm.*

Spis treści

Streszczenie	5
Abstract	7
1. Wprowadzenie	11
2. Napęd elektryczny z silnikiem PMSM o zmiennym momencie bezwładności	18
2.1. Model matematyczny	18
2.2. Struktury sterowania napędu z silnikiem PMSM w trybie regulacji prędkości kątovej	22
2.2.1. Kaskadowa struktura regulacji	22
2.2.2. Regulator bazujący na sprzężeniu od wektora zmiennych stanu	23
2.3. Wyniki badań symulacyjnych	26
2.4. Sterowanie adaptacyjne w napędach elektrycznych	31
2.4.1. Sterowanie z harmonogramem wzmocnienia	31
2.4.2. Automatyczne strojenie regulatora	32
2.4.3. Sterowanie ślizgowe	34
2.4.4. Sterowanie adaptacyjne z modelem odniesienia	34
3. Zastosowanie reguły Widrow'a-Hoff'a w mechanizmie adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu	36
3.1. Reguła Widrow'a-Hoff'a	36
3.2. Mechanizm adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu	37
3.2.1. Model odniesienia	38
3.2.2. Prawo adaptacji	40
3.3. Wyniki badań symulacyjnych	43
3.3.1. Implementacja modelu odniesienia	45
3.3.2. Implementacja mechanizmu adaptacji	45
3.3.3. Analiza wpływu współczynnika uczenia na jakość adaptacji	45
3.3.4. Implementacja modelu odniesienia o nieadekwatnym rzędzie	48
3.4. Podsumowanie	51

4. Zastosowanie algorytmów optymalizacyjnych w mechanizmie adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu	52
4.1. Algorytmy optymalizacyjne	52
4.2. Mechanizm adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu	53
4.2.1. Model odniesienia	55
4.2.2. Procedura adaptacji bazująca na algorytmie optymalizacyjnym	56
4.2.3. Globalny algorytm optymalizacyjny - Optymalizacja Rojem Cząstek	59
4.2.4. Lokalny algorytm optymalizacyjny - Przeszukiwanie wzorcem	62
4.3. Wyniki badań symulacyjnych	63
4.3.1. Implementacja mechanizmu adaptacji	65
4.3.2. Analiza działania zaproponowanego mechanizmu adaptacji	66
4.3.3. Analiza wpływu parametrów procedury adaptacji na przebieg procesu adaptacji	70
4.3.4. Analiza wpływu parametrów algorytmu optymalizacyjnego na jakość adaptacji	74
4.3.5. Analiza powtarzalności rozwiązania	76
4.3.6. Implementacja modelu odniesienia o nieadekwatnym rzędzie	76
4.4. Podsumowanie	79
5. Wyniki badań eksperymentalnych	80
5.1. Opis stanowiska badawczego	80
5.2. Mechanizm adaptacji bazujący na regule Widrow'a Hoff'a	83
5.2.1. Rezultaty	86
5.3. Mechanizm adaptacji bazujący na algorytmie optymalizacyjnym	89
5.3.1. Rezultaty - algorytm optymalizacji rojem cząstek	90
5.3.2. Rezultaty - algorytm przeszukiwania wzorcem	91
5.4. Podsumowanie	96
6. Wnioski	98
Bibliografia	102
Spis tabel	109
Spis rysunków	110
Załącznik 1: Kod źródłowy wybranych funkcji	115

1. Wprowadzenie

Silniki synchroniczne o magnesach trwałych (ang. *permanent magnet synchronous motor*, PMSM) zawdzięczają swoją popularność wysokiej sprawności i wydajności, kompaktowej konstrukcji, wysokiej gęstości mocy oraz korzystnemu stosunkowi rozwijanego momentu do masy [1, 2]. Są one powszechnie stosowane w maszynach sterowanych numerycznie [3], manipulatorach przemysłowych [4], samochodach elektrycznych i hybrydowych [5] oraz systemach sanitarnych (tj. wentylacje oraz klimatyzacje) [6]. Regulacja prądu oraz prędkości w napędach elektrycznych z silnikami PMSM jest powszechnie realizowana przy pomocy kaskadowej struktury regulacji z wykorzystaniem regulatorów proporcjonalno-całkujących (ang. *proportional-integral*, PI). Głównymi powodami są: prosta implementacja oraz intuicyjne metody strojenia [7]. Kaskadowa struktura regulacji zawiera kilka pojedynczych pętli regulacji z regulatorami PI. W związku z tym dobór współczynników regulatorów PI musi odbywać się od wewnętrznej do zewnętrznej pętli regulacji. W celu ograniczenia oscylacji i dużych wartości przeregulowania, pasmo przenoszenia kaskadowej struktury regulacji jest często ograniczone. Skutkuje to stosunkowo długim czasem narastania i regulacji prędkości kątowej silnika oraz słabą kompensacją momentu obciążenia [8]. Alternatywną strukturą sterowania dla napędu elektrycznego z silnikiem PMSM jest regulator ze sprzężeniem od wektora zmiennych stanu. Charakteryzuje się on pojedynczą pętlą sprzężenia zwrotnego od wszystkich zmiennych stanu. Zapewnia on lepszą odporność i tolerancję na nieliniowości występujące w obiekcie [9], lepsze właściwości dynamiczne i kompensację zakłóceń [10] w porównaniu do kaskadowej struktury regulacji.

Niezależnie od wybranej struktury regulacji, głównym założeniem podczas doboru jej współczynników jest niezmiennosc parametrów sterowanego obiektu. Przy tak zaprojektowanym regulatorze o stałych współczynnikach wzmocnień, zmiany parametrów obiektu mogą doprowadzić do niepożądanego odpowiedzi układu na zadany sygnał referencyjny, np. wystąpienie przeregulowania lub znaczne wydłużenie czasu regulacji prędkości kątowej silnika. W celu poprawy odporności układów sterowania w szerokim zakresie zmian parametrów obiektu, można zastosować regulator adaptacyjny. Jest to struktura, która

zmienia swoje zachowanie w odpowiedzi na zewnętrzne zakłócenie lub zmianę parametrów obiektu [11]. W sterowaniu adaptacyjnym można wyróżnić dwie skale czasowe procesu: wewnętrzną związaną z tradycyjnym sprzężeniem zwrotnym oraz zewnętrzną odpowiadającą za aktualizację współczynników struktury sterowania. Dzięki zastosowaniu regulacji adaptacyjnej możliwe jest zwiększenie wydajności energetycznej lub produkcyjnej oraz funkcjonalności systemu. W ostatnich latach regulatory adaptacyjne były tematem wielu prac naukowych [12–21]. W zależności od obiektu oraz jego parametrów, które mogą ulegać zmianie, stosowane są różne schematy adaptacyjne. Najczęściej stosowanymi są: sterowanie ślizgowe (ang. *sliding-mode control*), sterowanie z harmonogramem wzmocnienia (ang. *gain-scheduling control*), regulatory samostrojące (ang. *self-tuning controllers*) oraz sterowanie adaptacyjne z modelem odniesienia (ang. *model-reference adaptive system*, MRAS) [11].

Wybór schematu adaptacji oraz struktury regulacji motywowany jest zwiększeniem wydajności lub odporności układu regulacji automatycznej. Zwiększenie wydajności lub zapewnienie odporności układu można rozpatrywać jako zagadnienie optymalizacji struktury regulacji, w którym na przykład minimalizujemy uchyb pomiędzy sygnałem referencyjnym a odpowiedzią układu niezależnie od parametrów obiektu regulacji. Z tego powodu zagadnienie optymalizacji coraz częściej pojawia się w problemach związanych z napędem elektrycznym. W ostatnich latach algorytmy optymalizacyjne inspirowane przyrodą stały się jednym z podstawowych narzędzi do optymalizacji nietrywialnych problemów optymalizacyjnych z ograniczeniami. W przeciwieństwie do klasycznych metod optymalizacyjnych pozwalają one na globalne przeszukiwanie przestrzeni rozwiązań i równocześnie zapewnienie dobrej zbieżności algorytmu w stosunkowo krótkim czasie. Najczęściej stosowanymi algorytmami optymalizacyjnymi inspirowanymi przyrodą są: algorytm genetyczny (ang. *genetic algorithm*) [22], algorytm mrówkowy (ang. *ant colony optimization*) [23] oraz optymalizacja rojem cząstek (ang. *particle swarm optimization*) [24]. Wymienione algorytmy pozwoliły na rozwiązanie wielu skomplikowanych problemów optymalizacyjnych [25–30] oraz były inspiracją dla naukowców do opracowania nowych algorytmów inspirowanych przyrodą, np. sztuczna kolonia pszczół (ang. *artificial bee colony*) [31], algorytm nietoperzy (ang. *bat algorithm*) [32], kukułcze przeszukiwanie (ang. *cuckoo search*) [33], algorytm światełek (ang. *firefly algorithm*) [34], algorytm wilków szarych (ang. *grey wolf optimizer*) [35].

Jednym z dynamicznie rozwijających się w ostatnich latach zastosowaniem algorytmów optymalizacyjnych przyrodą jest automatyczne strojenie regulatorów [36–41]. Głównym powodem jest możliwość zdefiniowania parametrów określających jakość odpowiedzi układu sterowania na sygnał skokowy sygnału referencyjnego, np. oczekiwany czas narastania oraz maksymalne dozwolone przeregulowanie lub minimalizację całkowego wskaźnika jakości w celu maksymalizacji efektywności układu. Ponadto zostały opracowane mechanizmy pozwalające ograniczyć wzmacnianie szumów pomiarowych oraz uwzględnić ograniczenia, np. maksymalny prąd silnika [36]. Warto zaznaczyć, że tak rozbudowanych możliwości nie posiadają klasyczne metody doboru współczynników regulatorów, tj. metody Ziegler'a-Nichols'a dla regulatorów proporcjonalno-całkująco-różniczkujących lub zastosowanie optymalizacji liniowo-kwadratowej dla regulatorów ze sprzężeniem od wektora zmiennych stanu. Zadanie doboru współczynników nie jest trywialne, a staje się jeszcze bardziej złożone w przypadku założenia, że parametry mechaniczne obiektu, np. moment bezwładności, są zmienne w czasie. W takim przypadku, zmiana parametrów mechanicznych prowadzi do fluktuacji odpowiedzi układu w porównaniu do odpowiedzi oczekiwanej, czyli otrzymanej dla nominalnych parametrów obiektu.

Opracowanie mechanizmów kształtowania odpowiedzi układu na skokową zmianę sygnału referencyjnego dzięki zastosowaniu algorytmów optymalizacyjnych inspirowanych przyrodą do procesu automatycznego doboru współczynników, zainspirowały naukowców do zbadania możliwości zastosowania ich w regulacji adaptacyjnej. Automatyczny dobór bazy reguł oraz funkcji przynależności regulatora bazującego na logice rozmytej został zaproponowany w [42]. Pozwoliło to na znaczne przyśpieszenie procedury doboru parametrów struktury regulacji w porównaniu do wyznaczania ich metodą prób i błędów. Ponadto otrzymano optymalną jakość regulacji dla analizowanej struktury. W [43] została zaprezentowana metoda doboru parametrów adaptacyjnego regulatora predykcyjnego bazującego na logice rozmytej przy wykorzystaniu algorytmu optymalizacji rojem cząstek. Pozwoliła ona na dostarczenie optymalnych własności układu regulacji w szerokim zakresie parametrów obiektów regulacji, tj. reaktora zbiornikowego z mieszaniem ciągłym oraz systemu zbiorników. Zastosowanie algorytmu optymalizacyjnego nietoperzy do efektywnej optymalizacji współczynnika adaptacji w regule Widrow'a-Hoff'a dla układu dwumasowego zostało przedstawione w [44]. Przedstawiony mechanizm pozwolił uzyskać współczynnik dostarczający oczekiwaną jakość adaptacji, tj. stosunkowo krótki czas adaptacji i zapewnienie znalezienia globalnego optimum.

Powyższe prace można uznać za rozwinięcie metod automatycznego doboru współczynników regulatora wykorzystując algorytmy optymalizacyjne inspirowane przyrodą - procedura optymalizacji odbywała się w trybie off-line podczas syntezy regulatora, tj. współczynnik adaptacji był optymalizowany na podstawie wyników symulacyjnych, a następnie otrzymana wartość została zaimplementowana w napędzie elektrycznym.

Kolejnym istotnym krokiem w rozwoju zastosowania algorytmów optymalizacyjnych inspirowanych przyrodą w regulacji adaptacyjnej było ich wykorzystanie w trybie on-line, czyli w trakcie pracy całego układu. W [45] autorzy przedstawili zastosowanie algorytmu optymalizacji rojem cząstek do adaptacji regulatora prędkości w kaskadowej strukturze regulacji z regulatorami PI dla napędu z silnikiem PMSM. Warto zaznaczyć, że warunkiem takiego zastosowania algorytmu optymalizacyjnego jest powtarzalność procesu, czyli cykliczność sygnału referencyjnego. Wynika to z działania algorytmów optymalizacyjnych, które porównują nowe rozwiązanie z poprzednim. Żeby funkcja celu zależna od uchybu prędkości była porównywalna pomiędzy kolejnymi rozwiązaniami, to sygnał referencyjny oraz warunki początkowe muszą być zawsze takie same. W omawianej pracy algorytm optymalizacji rojem cząstek jest odpowiedzialny za aktualizację w trybie on-line współczynników regulatora prędkości PI w napędzie z silnikiem PMSM o zmiennym momencie bezwładności. Praca wykazała potencjał zastosowania algorytmów optymalizacyjnych inspirowanych przyrodą jako mechanizm adaptacji w celu zapewnienia optymalnego sterowania, zdefiniowanego wielocłonową funkcją celu. Wyniki opracowanego algorytmu adaptacji przedstawiały tylko badania symulacyjne. Według najlepszej wiedzy autora niniejszej rozprawy doktorskiej, pierwszy raz, gdy za zmianę parametrów regulatora w warunkach laboratoryjnych odpowiadał algorytm optymalizacyjny inspirowany przyrodą, zaprezentowano w pracy [46]. Przedstawiono w niej nowatorskie zastosowanie algorytmu sztucznej kolonii pszczół oraz sterowania adaptacyjnego z modelem odniesienia dla napędu z silnikiem PMSM. Algorytm optymalizacyjny został zastosowany jako mechanizm adaptacji zamiast znanych metod, tj. reguła Widrow'a-Hoff'a, funkcja Lyapunov'a czy teoria pasywności. Zastosowanie algorytmu sztucznej kolonii pszczół do optymalizacji w trybie on-line wymagało wprowadzenia dodatkowej fazy algorytmu, która wykrywała zmiany parametrów obiektu oraz uruchamiała optymalizację. Algorytm sztucznej kolonii pszczół wyznaczał nowe macierze wagowe dla optymalizacji liniowo-kwadratowej w celu minimalizacji różnicy pomiędzy modelem odniesienia a rzeczywistą odpowiedzią układu. Nowatorska kombinacja sterowania

adaptacyjnego z algorytmem optymalizacyjnym inspirowanym przyrodą pozwoliła uzyskać adaptacyjny regulator odporny na zmienne parametry obiektu. Przedstawiony powyżej bieżący stan wiedzy w rozpatrywanym zagadnieniu skłonił Autora niniejszej rozprawy doktorskiej do dalszych badań nad zastosowaniem algorytmów optymalizacyjnych inspirowanych przyrodą w regulacji adaptacyjnej dla napędu elektrycznego z silnikiem PMSM bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu.

Teza: Możliwe jest uzyskanie powtarzalnych właściwości regulacyjnych napędu elektrycznego z silnikiem PMSM o zmiennym momencie bezwładności poprzez zastosowanie sterowania adaptacyjnego z modelem odniesienia bazującego na zaproponowanej procedurze adaptacji dla algorytmów optymalizacyjnych.

Za cel pracy przyjęto opracowanie struktury adaptacyjnej bazującej na algorytmach optymalizacyjnych inspirowanych przyrodą w celu zapewnienia niezmiennej odpowiedzi silnika PMSM o zmiennym momencie bezwładności. Za najważniejsze zadania niezbędne do realizacji przyjętego celu uznano:

- opracowanie modeli symulacyjnych napędu elektrycznego z silnikiem PMSM o zmiennym momencie bezwładności,
- analiza wpływu zmian momentu bezwładności na odpowiedź skokową układu w celu określenia złożoności problemu i doboru odpowiedniego algorytmu optymalizacyjnego,
- opracowanie struktury regulacji adaptacyjnej bazującej na regule Widrow'a-Hoff'a jako klasycznej metody, która będzie punktem odniesienia do opracowanego mechanizmu adaptacji bazującego na algorytmach optymalizacyjnych inspirowanych przyrodą,
- opracowanie struktury regulacji adaptacyjnej wykorzystującej algorytm optymalizacyjny inspirowany przyrodą,
- badania symulacyjne napędu elektrycznego z regulatorem adaptacyjnym,
- uruchomienie stanowiska badawczego oraz implementacja opracowanych algorytmów,
- przeprowadzenie badań eksperymentalnych regulatora adaptacyjnego,
- analiza otrzymanych wyników w porównaniu do klasycznego mechanizmu adaptacji.

Praca składa się z sześciu rozdziałów uzupełnionych wykazem literatury oraz fragmentami kodu źródłowego zaimplementowanego mechanizmu adaptacji bazującego na algorytmach optymalizacyjnych. W pierwszym rozdziale zamieszczono wprowadzenie, sformułowano tezę oraz cele niniejszej rozprawy.

W drugim rozdziale przedstawiony został matematyczny opis napędu elektrycznego z silnikiem PMSM. Został on również zlinearyzowany w celu uproszczenia syntezy struktury regulacji. Omówione zostały struktury sterowania napędem elektrycznym z silnikiem PMSM oraz zawarto badania symulacyjne, z wykorzystaniem środowiska MATLAB/SIMULINK, stacjonarnego regulatora ze sprzężeniem od wektora zmiennych stanu dla silnika PMSM o zmiennym momencie bezwładności. Rozdział zakończono przeglądem struktur regulacji adaptacyjnej stosowanych w napędach elektrycznych.

Rozdział trzeci zawiera opis regulatora adaptacyjnego bazującego na regule Widrow'a-Hoff'a. Zawarto w nim opis ów reguły oraz mechanizm adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu ją wykorzystujący. Wyprowadzono transmitancję modelu odniesienia oraz zaproponowano prawo adaptacji dla analizowanego napędu elektrycznego z silnikiem PMSM, wykorzystujący regułę Widrow'a-Hoff'a. W rozdziale zostały przedstawione opracowane modele symulacyjne, a następnie zawarto wyniki szeregu badań symulacyjnych.

W czwartym rozdziale został przedstawiony opis autorskiej struktury regulacji adaptacyjnej bazującej na algorytmach optymalizacyjnych. Metoda pozwala na zastosowanie zarówno metod optymalizacyjnych klasycznych jak również inspirowanych przyrodą. Omówiono zagadnienie dotyczące wyboru algorytmów optymalizacyjnych oraz procedurę adaptacji wykorzystującą je w celu zapewnienia odporności układu regulacji na zmianę parametrów obiektu. Projektowanie modelu odniesienia zostało również omówione, ponieważ zaproponowana metoda pozwala na znaczące uproszczenie tego problemu. Zawarto również opis algorytmów optymalizacyjnych wybranych do dalszych rozważań: algorytm optymalizacji rojem cząstek oraz algorytm przeszukiwania wzorcem. Rozdział kończą wyniki badań symulacyjnych.

Rozdział piąty zawiera opis stanowiska laboratoryjnego oraz wyniki badań eksperymentalnych. Zawiera opis zespołu maszynowego, głównych parametrów badanego silnika, urządzenia mechatronicznego pozwalającego na zmianę momentu bezwładności układu oraz prototypu trójfazowego dwupoziomowego falownika napięcia. Omówiono potencjalne problemy z implementacją mechanizmu adaptacji bazującego na regule Widrow'a-Hoff'a wynikające

z dokładności liczb zmiennoprzecinkowych o pojedynczej precyzji w prototypowym falowniku napięcia. Przetawiono wyniki badań eksperymentalnych dla mechanizmu adaptacji bazującego na regule Widrow'a-Hoff'a oraz bazującego na algorytmach optymalizacyjnych.

Wnioski końcowe, opis osiągnięć własnych i spostrzeżenia autora zamieszczono w rozdziale blueszóstym.

2. Napęd elektryczny z silnikiem PMSM o zmiennym momencie bezwładności

W niniejszym rozdziale zawarto opis matematyczny napędu elektrycznego z silnikiem PMSM niezbędny do przeprowadzenia badań symulacyjnych. Przedstawiono również struktury regulacji prędkości kątowej z uwzględnieniem rozwiązań adaptacyjnych stosowanych w napędach elektrycznych.

2.1. Model matematyczny

Model matematyczny napędu elektrycznego opisuje podstawowe oddziaływania elektromagnetyczne i mechaniczne występujące w modelowanej maszynie elektrycznej. W tym celu konieczne jest przygotowanie opisu układu w możliwie prostej postaci, uwzględniającej opis występujących zjawisk istotnych z punktu widzenia syntezy układu regulacji. W celu uzyskania matematycznego opisu napędu z silnikiem PMSM, zostały przyjęte następujące założenia [36, 46, 47]:

- wykorzystano silnik z magnesami rozłożonymi na powierzchni wirnika, w którym występuje symetria obwodu magnetycznego,
- dynamika przekształtnika energoelektronicznego została pominięta,
- silnik ma sinusoidalny przebieg sił elektromotorycznych przy stałej prędkości kątowej oraz sinusoidalny rozkład strumienia od magnesów trwałych,
- pominięto wpływ prądów wirowych,
- tarcie występujące w układzie zostaje uproszczone do tarcia wiskotycznego, tj. pominięto tarcie statyczne, Newtonowskie oraz efekt Stribeck'a,
- układ mechaniczny jest idealnie sztywny, tj. nie występują w układzie sprężystości ani luzy,
- wykorzystany zostanie model silnika w ortogonalnym, wirującym z prędkością wirnika układzie odniesienia d - q .

Powyższe założenia pozwalają na uproszczenia, które wpływają na dokładność odwzorowania układu rzeczywistego w bardzo niewielkim stopniu. Dzięki doprecyzowaniu konstrukcji silnika, tj. magnesy rozłożone na powierzchni wirnika, można przyjąć, że indukcyjności w osi d oraz q są sobie równe ($L_s = L_d = L_q$) [48]. Wynika to z symetrii obwodu magnetycznego. Postęp w rozwoju półprzewodnikowych przyrządów mocy pozwolił na konstrukcję przekształtników cechujących się krótkimi czasami przełączeń skutkującymi redukcją strat mocy. Dzięki temu możliwe stało się zwiększenie częstotliwości pracy łączników, co w konfrontacji z kilka rzędów większą stałą elektryczną silnika pozwoliło uzyskać liniowy charakter pracy układu w szerokim zakresie napięć i prądów. W praktyce przełożyło się to na możliwość modelowania przekształtnika w postaci członu proporcjonalnego [10]. Otrzymana dokładność jest wystarczająco dobra, aby z powodzeniem zastosować taki model do syntezy regulatorów. W takim przypadku przekształtnik energoelektroniczny jest traktowany jako wzmacnienie statyczne, a napięcia w osi d i q przyjmują postać:

$$u_d(t) = K_p u_{dn}(t) \quad (2.1)$$

$$u_q(t) = K_p u_{qn}(t) \quad (2.2)$$

gdzie: sygnały $u_{dn}(t)$ i $u_{qn}(t)$ są znormalizowane do wartości $\langle -1, +1 \rangle$, K_p oznacza wzmacnienie przekształtnika. Wykorzystując powyższe założenia, model matematyczny napędu z silnikiem PMSM uzupełniony o równanie ruchu przyjmuje postać [47]:

$$K_p u_{dn}(t) = R_s i_d(t) + L_s \frac{d}{dt} i_d(t) - L_s p \omega_m(t) i_q(t) \quad (2.3)$$

$$K_p u_{qn}(t) = R_s i_q(t) + L_s \frac{d}{dt} i_q(t) + L_s p \omega_m(t) i_d(t) + p \omega_m(t) \psi_f \quad (2.4)$$

$$m_e(t) = \frac{3}{2} p \psi_f i_q(t) \quad (2.5)$$

$$\frac{d\omega_m(t)}{dt} = \frac{1}{J} (m_e(t) - m_o(t) - B\omega_m(t)) \quad (2.6)$$

gdzie: $u_{dn}(t)$, $u_{qn}(t)$, $i_d(t)$, $i_q(t)$ - sygnały sterujące oraz prądy odpowiednio w osiach d i q , R_s , L_s - rezystancja i indukcyjność uzwojeń stojana, ψ_f - strumień pochodzący od magnesów trwałych, p - liczba par biegunów, $\omega_m(t)$ - prędkość kątowna wału silnika, $m_e(t)$ - moment elektromagnetyczny wygenerowany przez silnik, $m_o(t)$ - moment obciążenia, J - moment bezwładności, B - współczynnik tarcia wiskotycznego. Przedstawiony model jest nieliniowy

ze względu na występujące iloczyny składowych prądu w osi q i d oraz prędkości wału silnika w równaniach (2.3)-(2.4). Ponadto, w modelu występują również tzw. sprzężenia skrośne.

Synteza regulatora dla obiektu nieliniowego bywa bardziej skomplikowana i wymusza stosowanie, np. przestrajania regulatora w zależności od punktu pracy układu [49, 50]. W celu uproszczenia procedury projektowania oraz doboru współczynników układu sterowania dla obiektu nieliniowego powszechnym podejściem jest zastosowanie linearyzacji poprzez sprzężenie zwrotne (tzw. odsprężanie) [47]. Modyfikując równania (2.3)-(2.4) uzyskuje się następujące zależności:

$$K_p u_{dn}(t) + L_s p \omega_m(t) i_q(t) = R_s i_d(t) + L_s \frac{d}{dt} i_d(t) \quad (2.7)$$

$$K_p u_{qn}(t) - (L_s p \omega_m(t) i_d(t) + p \omega_m(t) \psi_f) = R_s i_q(t) + L_s \frac{d}{dt} i_q(t) \quad (2.8)$$

Definiując zmienne pomocnicze, tak aby prędkość kątowna nie występowała w sposób jawny w równaniach części elektrycznej napędu otrzymuje się:

$$K_p (u_{dn}(t) - u_{do}(t)) = R_s i_d(t) + L_s \frac{d}{dt} i_d(t) \quad (2.9)$$

$$K_p (u_{qn}(t) - u_{qo}(t)) = R_s i_q(t) + L_s \frac{d}{dt} i_q(t) \quad (2.10)$$

przy czym:

$$u_{do}(t) = -\frac{L_s p}{K_p} \omega_m(t) i_q(t), \quad (2.11)$$

$$u_{qo}(t) = \frac{p}{K_p} \omega_m(t) [L_s i_d(t) + \psi_f] \quad (2.12)$$

Dla przejrzystości zapisu można przyjąć $u_{dr}(t) = u_{dn}(t) - u_{do}(t)$ oraz $u_{qr}(t) = u_{qn}(t) - u_{qo}(t)$ i uzupełnić model o pozostałe równania, które nie wymagały linearyzacji, otrzymując ostateczną postać liniowego modelu matematycznego napędu z silnikiem PMSM:

$$K_p u_{dr}(t) = R_s i_d(t) + L_s \frac{d}{dt} i_d(t) \quad (2.13)$$

$$K_p u_{qr}(t) = R_s i_q(t) + L_s \frac{d}{dt} i_q(t) \quad (2.14)$$

$$m_e(t) = \frac{3}{2} p \psi_f i_q(t) \quad (2.15)$$

$$\frac{d\omega_m(t)}{dt} = \frac{1}{J}(m_e(t) - m_o(t) - B\omega_m(t)) \quad (2.16)$$

Jak wspomniano na początku rozdziału, analizowany jest silnik PMSM o magnesach trwałych montowanych powierzchniowo na wirniku, co powoduje, że indukcyjności w osiach d oraz q są sobie równe. W rezultacie stała momentu elektromagnetycznego silnika jest opisana następującą zależnością:

$$K_t = \frac{3}{2}p\psi_f \quad (2.17)$$

Dzięki czemu równanie (2.15) przyjmuje następującą postać:

$$m_e(t) = K_t i_q(t) \quad (2.18)$$

Zlinearyzowany model napędu z silnikiem PMSM opisany równaniami (2.15)-(2.17) zapisany w postaci równania stanu jest następujący:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}_{ss}\mathbf{x}(t) + \mathbf{B}_{ss}\mathbf{u}(t) + \mathbf{E}_{ss}z(t) \quad (2.19)$$

przy czym:

$$\mathbf{A}_{ss} = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & 0 \\ 0 & \frac{K_t}{J} & -\frac{B}{J} \end{bmatrix}, \quad \mathbf{B}_{ss} = \begin{bmatrix} \frac{K_p}{L_s} & 0 \\ 0 & \frac{K_p}{L_s} \\ 0 & 0 \end{bmatrix}, \quad \mathbf{E}_{ss} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J} \end{bmatrix},$$

$$\mathbf{x}(t) = \begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_m(t) \end{bmatrix}, \quad \mathbf{u}(t) = \begin{bmatrix} u_{dr}(t) \\ u_{qr}(t) \end{bmatrix}, \quad z(t) = m_o(t),$$

2.2. Struktury sterowania napędu z silnikiem PMSM w trybie regulacji prędkości kątovej

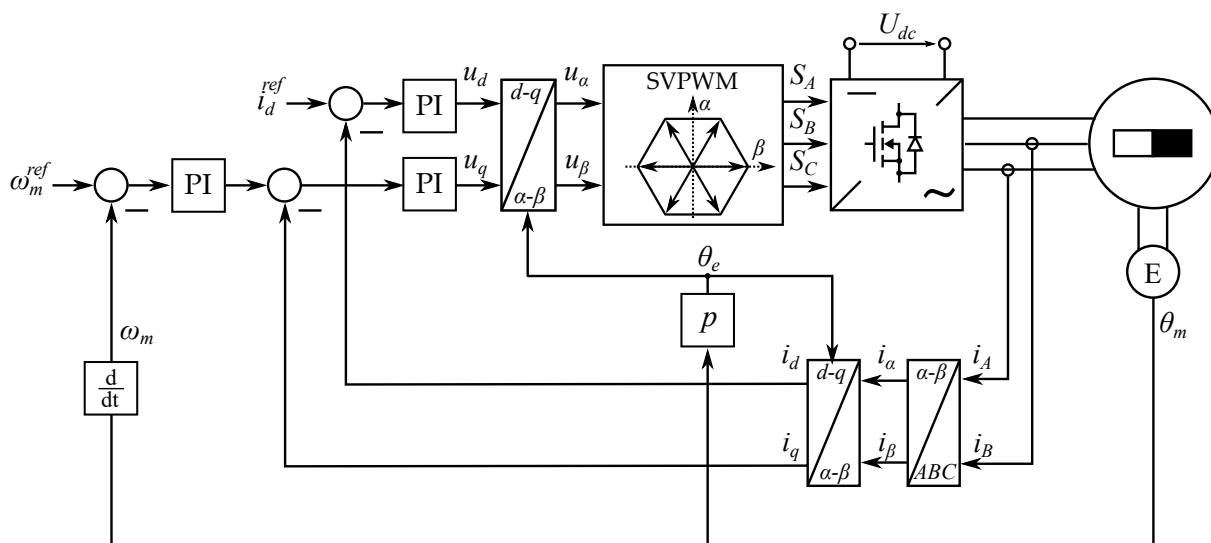
Jedną z najczęściej stosowanych metod sterowania napędu z silnikiem PMSM jest sterowanie metodą orientacji wektora pola (ang. *Field Oriented Control*, FOC) [48]. Co więcej metoda FOC oraz jej pochodne stanowią podstawę większości nowoczesnych napędów z silnikami asynchronicznymi i synchronicznymi [47]. Metoda ta pozwala na odsprężenie strumienia i momentu elektromagnetycznego silnika poprzez niezależne sterowanie składowymi wektora prądu w wirującym układzie współrzędnych $d-q$. Pozwala to na uzyskanie bardzo dobrych właściwości dynamicznych napędu w szerokim zakresie regulacji prędkości kątovej silnika.. Podstawowymi wadami metody FOC jest konieczność stosowania czujników pomiarowych położenia wału silnika.

2.2.1. Kaskadowa struktura regulacji

Najczęściej stosowaną strukturą regulacji prędkości kątovej napędu z silnikiem PMSM jest kaskadowa struktura regulacji z regulatorami proporcjonalno-całkującymi [51]. W literaturze naukowej można także spotkać struktury regulacji bazujące na: sterowaniu ślizgowym [52], nieliniowym regulatorze bazującym na logice rozmytej [53] lub sztucznych sieciach neuronowych [54], których zadaniem jest zapewnienie lepszej odporności układu sterowania na zewnętrzne zakłócenie lub/i zmienne parametry obiektu. Uzyskanie dobrych właściwości tłumienia oscylacji oraz ograniczenia przeregulowania wymaga często ograniczenia pasma przenoszenia układu. Przekłada się to na ograniczoną możliwość uzyskania przez kaskadową strukturę regulacji dużej dynamiki odpowiedzi przy jednocześnie dobrym tłumieniu niepożądanych oscylacji i przeregulowania [8].

Schemat blokowy kaskadowej struktury sterowania z regulatorami typu PI dla napędu elektrycznego z silnikiem PMSM został przedstawiony na rysunku 2.1. Na schemacie można wyróżnić dwa tory regulacji składowych prądu stojana odpowiednio w osiach d i q wirującego układu współrzędnych. Dla rozważanych silników o magnesach trwałych montowanych powierzchniowo na wirniku najczęściej stosuje się zerową wartość prądu referencyjnego w osi d . Regulatorem nadrzędnym regulatora prądu w osi q jest regulator prędkości. W celu przejścia z prądów fazowych silnika (i_A, i_B, i_C) na prądy w wirującym układzie współrzędnych $d-q$ stosuje się transformację Clarka (i_α, i_β), a następnie transformację Parka (i_d, i_q). Sygnały sterujące z kaskadowej struktury regulacji są poddawane odwrotnej

transformacji Parka, a następnie podawane na modulator generujący odpowiednie sygnały sterujące dla trójfazowego przekształtnika napięcia.



Rysunek 2.1. Schemat blokowy kaskadowej struktury regulacji napędu z silnikiem PMSM

2.2.2. Regulator bazujący na sprzężeniu od wektora zmiennych stanu

Alternatywną strukturą regulacji prędkości kątowej silnika PMSM jest zastosowanie regulatora bazującego na sprzężeniu od wektora zmiennych stanu. Głównymi zaletami regulatora ze sprzężeniem od wektora zmiennych stanu jest lepsza dynamika oraz odporność na zakłócenia w stosunku do kaskadowej struktury regulacji. W celu syntezy regulatora ze sprzężeniem od wektora zmiennych stanu niezbędna jest znajomość regulowanego obiektu w postaci równań stanu. Zakłada się, że pasmo przenoszenia zastosowanych czujników (pomiar prądu oraz położenia wału silnika) jest na tyle duże, że można je traktować jako bezinercyjne. W celu uzyskania zerowego uchybu w stanie ustalonym dla skokowych zmian sygnału referencyjnego prędkości kątowej oraz dla zmian momentu obciążenia wprowadza się dodatkową zmienną stanu będącą wewnętrznym modelem skokowego sygnału referencyjnego, wyrażoną równaniem [55]:

$$\frac{d}{dt}x_{\omega}(t) = \omega_m^{ref}(t) - \omega_m(t) \quad (2.20)$$

Wprowadzenie jej do układu powoduje, że niemierzalne obciążenie nie wprowadza niezerowego uchybu w stanie ustalonym, dzięki czemu można potraktować moment obciążenia $m_o(t)$ z równania (2.16) jako niemierzalny i wykluczyć je z modelu obiektu sterowania. Zastosowane założenia, uproszczenia i wprowadzenie dodatkowej zmiennej stanu

do równań (2.13)-(2.16) prowadzi do zapisu modelu matematycznego liniowego obiektu regulacji w postaci równania stanu:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}_{ss}\mathbf{x}(t) + \mathbf{B}_{ss}\mathbf{u}(t) + \mathbf{F}_{ss}r(t) \quad (2.21)$$

przy czym:

$$\mathbf{A}_{ss} = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & 0 & 0 \\ 0 & \frac{K_t}{J} & -\frac{B}{J} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B}_{ss} = \begin{bmatrix} \frac{K_p}{L_s} & 0 \\ 0 & \frac{K_p}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{F}_{ss} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix},$$

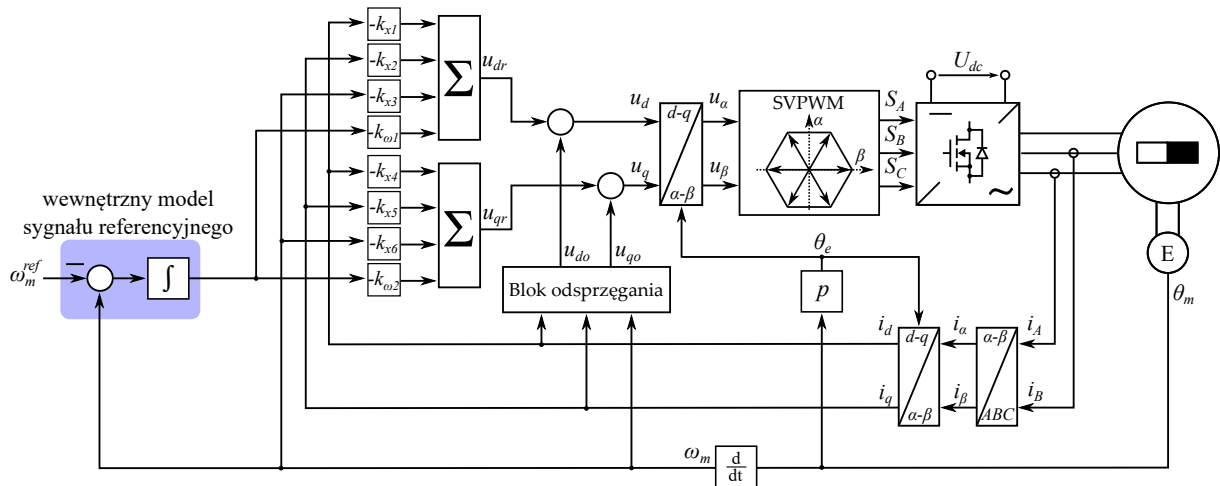
$$\mathbf{x}(t) = \begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_m(t) \\ x_\omega(t) \end{bmatrix}, \quad \mathbf{u}(t) = \begin{bmatrix} u_{dr}(t) \\ u_{qr}(t) \end{bmatrix}, \quad r(t) = \omega_m^{ref}(t)$$

Dla przedstawionego modelu prawo sterowania regulatora ze sprzężeniem od wektora zmiennych stanu jest następujące:

$$\mathbf{u}(t) = \begin{bmatrix} u_{dr}(t) \\ u_{qr}(t) \end{bmatrix} = -\mathbf{K}\mathbf{x}(t) = \begin{bmatrix} k_{x1} & k_{x2} & k_{x3} & k_{\omega1} \\ k_{x4} & k_{x5} & k_{x6} & k_{\omega2} \end{bmatrix} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_m(t) \\ x_\omega(t) \end{bmatrix} \quad (2.22)$$

Schemat blokowy napędu z silnikiem PMSM i regulatorem ze sprzężeniem od wektora zmiennych stanu został przedstawiony na rysunku 2.2. W strukturze regulacji występuje tylko jeden

regulator sterujący napięciami w osiach d oraz q . Jest on odpowiedzialny za regulację prądów w wirującym układzie współrzędnych $d-q$ oraz prędkości kątowej. Blok odsprężenia bazujący na równaniach (2.11)-(2.12) został uwzględniony w torze regulacji zgodnie z założeniami przyjętej metody linearyzacji.



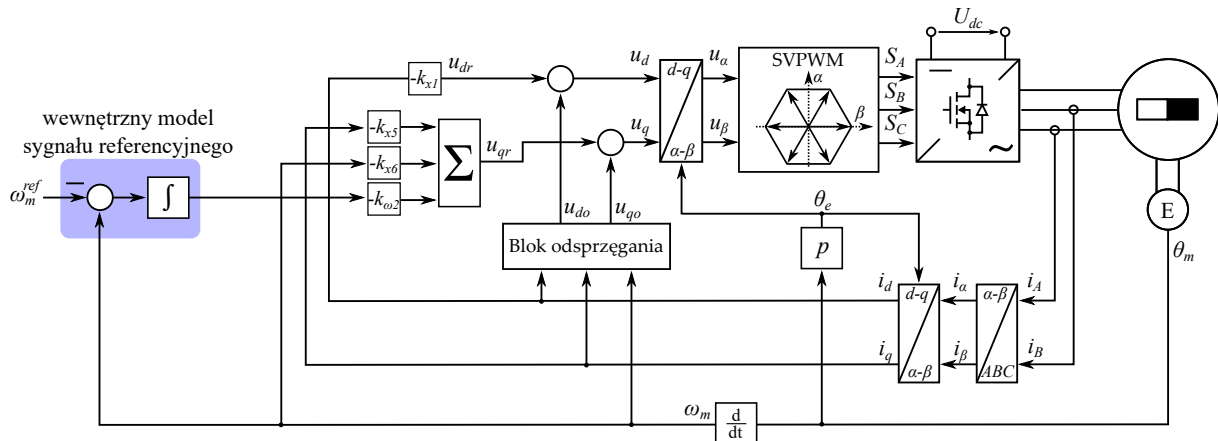
Rysunek 2.2. Schemat blokowy napędu z silnikiem PMSM i regulatorem ze sprzężeniem od wektora zmiennych stanu

Warto zauważyć, że w przedstawionym zlinearyzowanym modelu matematycznym silnika PMSM, prędkość kątowa jest niezależna od prądu w osi d , a prąd w tej osi nie zależy od prądu w osi q ani od prędkości kątowej silnika. Brak sprzężenia między osią d a osią q oraz faktem, że prędkość kątowa zależy tylko od napięć w osi q , współczynniki k_{x2} , k_{x3} , $k_{\omega1}$ oraz k_{x4} mają wartości zerowe [10,36]. Dzięki temu prawo sterowania może zostać uproszczone do następującej postaci:

$$\mathbf{u}(t) = \begin{bmatrix} u_{dr}(t) \\ u_{qr}(t) \end{bmatrix} = -\mathbf{K}\mathbf{x}(t) = \begin{bmatrix} k_{x1} & 0 & 0 & 0 \\ 0 & k_{x5} & k_{x6} & k_{\omega2} \end{bmatrix} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_m(t) \\ x_\omega(t) \end{bmatrix} \quad (2.23)$$

Schemat blokowy napędu z silnikiem PMSM bazującym na regulatorze ze sprzężeniem od wektora zmiennych stanu o zredukowanej liczbie współczynników został przedstawiony na rysunku 2.3. Ponadto zmiana momentu bezwładności zgodnie z przyjętym w równaniu (2.13) modelem matematycznym nie ma wpływu na układ regulacji w osi d . Dzięki czemu owe zmiany nie wpływają bezpośrednio na pętlę regulacji w osi d , tj. nie ma konieczności zmiany współczynnika k_{x1} w przypadku zmiany momentu bezwładności układu [20,46]. Zastosowanie

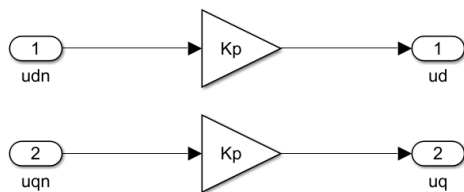
regulatora ze sprzężeniem od wektora zmiennych stanu pozwala na uzyskanie lepszej dynamiki odpowiedzi skokowej oraz lepszej kompensacji momentu obciążenia [56, 57] w stosunku do kaskadowej struktury regulacji z regulatorami. Ze względu na wyżej wspomniane właściwości, przedmiotem dalszych rozważań w zakresie projektowania adaptacyjnej struktury regulacji zostanie zastosowana struktura sterowania bazująca na sprzężeniu od wektora zmiennych stanu.



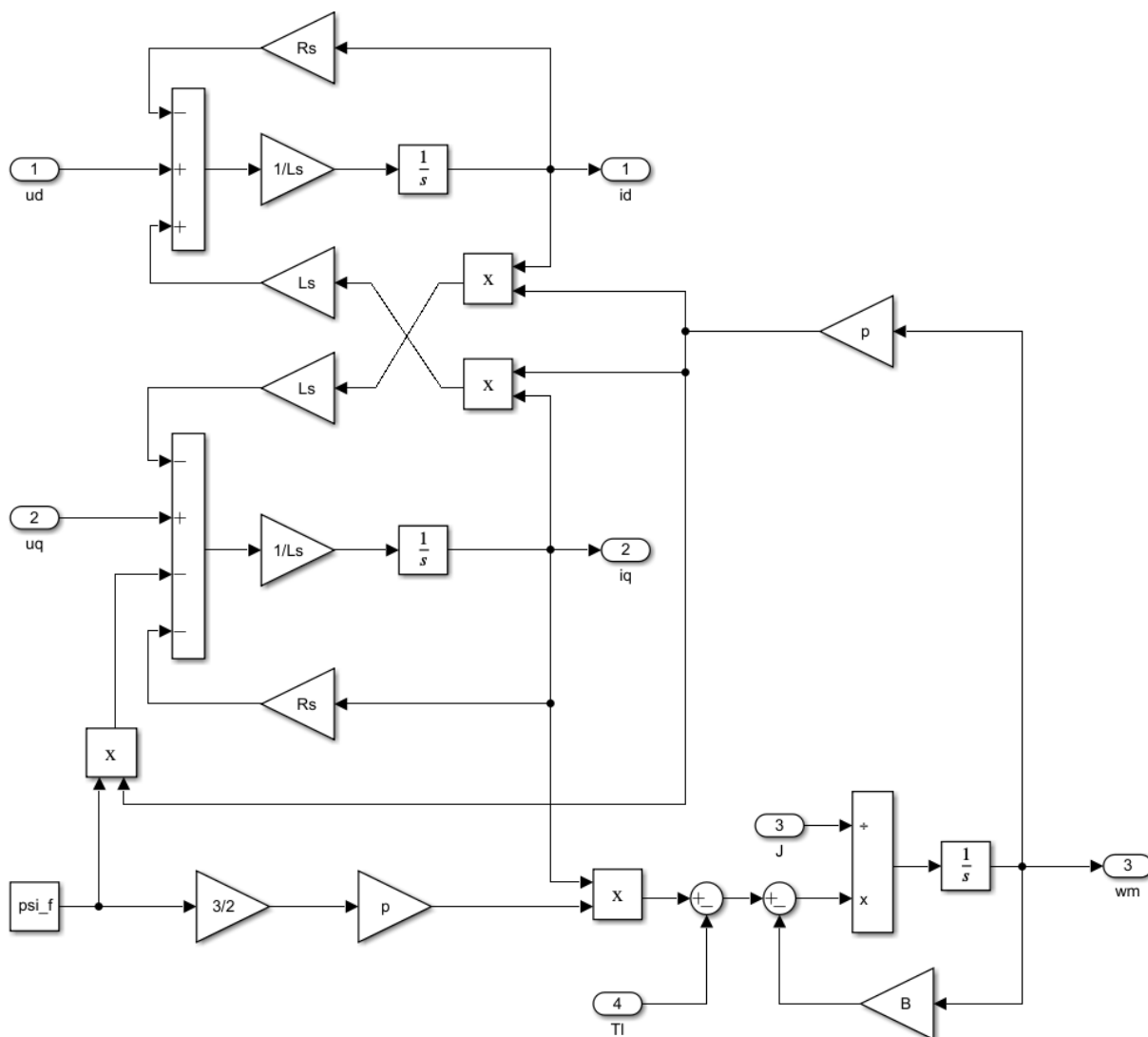
Rysunek 2.3. Schemat blokowy napędu z silnikiem PMSM bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu o zredukowanej liczbie współczynników

2.3. Wyniki badań symulacyjnych

W ramach badań symulacyjnych z wykorzystaniem środowiska MATLAB/SIMULINK zbadano wpływ zmian momentu bezwładności na odpowiedź układu. Model symulacyjny napędu został opracowany bazując na schemacie przedstawionym na rysunku 2.3, wykorzystując nieliniowy model matematyczny wyrażony w wirującym układzie współrzędnych $d-q$ oraz mechanizm odsprężania nieliniowości (równania (2.9)-(2.12)). Został on przedstawiony na rysunku 2.4. Schemat przedstawia regulator ze sprzężeniem od wektora zmiennych stanu, model silnika PMSM wyrażony w wirującym układzie współrzędnych $d-q$ oraz model przekształtnika energoelektronicznego. Implementacja modelu silnika PMSM o zmiennym momencie bezwładności oraz modelu przekształtnika energoelektronicznego zostały przedstawione na rysunkach 2.6 oraz 2.5. W celu zapewnienia możliwości dowolnej zmiany momentu bezwładności silnika, owa zmienna została zaimplementowana jako dodatkowe wejście do modelu silnika PMSM. Na schemacie znajduje się definicja globalnej zmiennej J oznaczającej moment bezwładności, przełącznik czasowy zmieniający jej wartość oraz została



Rysunek 2.5. Model symulacyjny przekształtnika energoelektronicznego opracowany w programie MATLAB/SIMULINK



Rysunek 2.6. Model symulacyjny silnika PMSM o zmiennym momencie bezwładności opracowany w programie MATLAB/SIMULINK

ona wprowadzona jako dodatkowe wejście do modelu silnika PMSM analogicznie do momentu obciążenia. W ramach struktury sterowania zaimplementowano wewnętrzny model sygnału referencyjnego, zredukowaną postać regulatora ze sprzężeniem od wektora zmiennych stanu

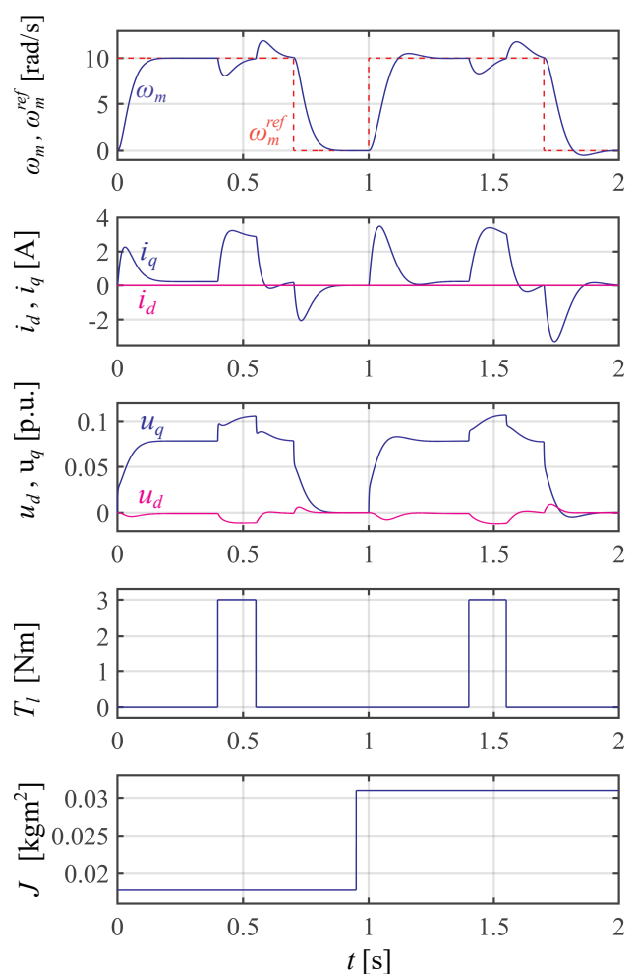
dla napędu elektrycznego z silnikiem PMSM, blok odsprężania oraz ograniczenie sygnału sterującego w osiach d i q do wartości z przedziału $\langle -1, 1 \rangle$.

Dobór współczynników został przeprowadzony ręcznie przez autora, bazując na doświadczeniu oraz wcześniejszych wynikach badań naukowych związanych z automatycznym doбором współczynników regulatora ze sprzężeniem od wektora zmiennych stanu [58, 59]. Głównymi założeniami było uzyskanie stosunkowo krótkiego czasu narastania, brak lub pomijalnie małe przeregulowanie oraz zapewnienie, że suma chwilowych prądów w wirującym układzie współrzędnych $d-q$ jest poniżej połowy wartości nominalnej prądu silnika PMSM. Ostatnie założenie dotyczące sumy prądów jest związane z dużym wpływem zmian momentu bezwładności na szczytową wartość prądu w osi q podczas skokowej zmiany sygnału referencyjnego. Parametry napędu oraz współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu wykorzystane do badań symulacyjnych zostały przedstawione w tabeli 2.1, a odpowiedź skokową oraz odpowiedź na skokową zmianę momentu obciążenia można zaobserwować na rysunku 2.7 w pierwszej sekundzie eksperymentu.

Tabela 2.1. Parametry napędu oraz współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu

Symbol	Wartość	Jednostka	Symbol	Wartość	Jednostka
P_N	1.73	kW	J_{nom}	0.0178	kgm ²
I_N	5.8	A	J_{add}	0.0134	kgm ²
p	3		$J_{nom} + J_{add}$	0.0312	kgm ²
R_s	1.05	Ω	U_{dc}	200	V
L_s	12.68	mH	K_p	100	
Ψ_f	0.2544		f_{PWM}	22	kHz
B	0.0252	Nms/rad	T_s	45.(45)	us
k_{x1}	0.0725		k_{x5}	0.0900	
k_{x6}	0.0979		$k_{\omega 2}$	1.9286	

W celu zaprezentowania wpływu zmian momentu bezwładności na odpowiedź napędu elektrycznego na wymuszenie sygnałem skokowym na rysunku 2.7 przedstawiono odpowiedzi dla nominalnego (J_{nom}) oraz zwiększonego ($J_{nom} + J_{add}$) momentu bezwładności. Dolny wykres przedstawia moment zmiany momentu bezwładności – zmiana następuje 0.05 s przed wystąpieniem narastającego zbocza sygnału referencyjnego. Dodatkowo wprowadzono zewnętrzne zakłócenie w postaci momentu obciążenia o wartości 3 Nm. Warto zaznaczyć, że współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu zostały dobrane dla nominalnej wartości momentu bezwładności. Odpowiedź dla tej wartości charakteryzuje



Rysunek 2.7. Wpływ zmiany momentu bezwładności na odpowiedź skokową napędu elektrycznego z silnikiem PMSM bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu

się brakiem przeregulowania (wartość poniżej 0.1%), czasem narastania na poziomie 82.1 ms oraz czasem regulacji dla 2% tolerancji wynoszącym 137.8 ms. W przypadku zwiększonego momentu bezwładności pojawia się przeregulowanie na poziomie 5%, a czas narastania oraz regulacji wynoszą odpowiednio 76.9 ms oraz 221.4 ms. Wprowadzenie skokowej zmiany momentu obciążenia dla obu wartości przyniosło bardzo zbliżony efekt. W przypadku wartości nominalnej momentu bezwładności prędkość chwilowo spadła do wartości 8.06 rad/s, a dla zwiększonej wartości była to wartość 8.27 rad/s. Czas powrotu do wartości zadanej wynosił odpowiednio 124.8 ms i 134.4 ms dla nominalnej i zwiększonej wartości momentu bezwładności. Znaczna różnica jest również widoczna w przebiegach składowej q wektora przestrzennego prądu stojana. Podczas rozpędzania silnika przy nominalnym momencie bezwładności wartość maksymalna prądu osiągnęła wartość 2.27 A, podczas gdy dla zwiększonego momentu bezwładności ta wartość wynosiła 3.49 A.

Otrzymane wyniki ukazują znaczne zmiany w odpowiedziach skokowych prędkości kątowej spowodowane fluktuacją momentu bezwładności. Najbardziej znaczącymi zmianami w parametrach określających jakość dynamiczną odpowiedzi skokowej jest pojawienie się przeregulowania oraz znaczne wydłużenie czasu regulacji. W przypadku, gdy tego typu zmiany są niepożądane (np. w przenośnikach taśmowych [60–62]), można zastosować strukturę regulacji adaptacyjnej.

2.4. Sterowanie adaptacyjne w napędach elektrycznych

Niezależnie od wybranej struktury regulacji (np. kaskada regulatorów PI, regulator ze sprzężeniem od wektora zmiennych stanu, regulator rozmyty), głównym założeniem podczas doboru jej współczynników jest niezmiennosc parametrów sterowanego obiektu. W takim przypadku, po zakończeniu procesu syntezy otrzymuje się stacjonarny regulator o stałych współczynnikach. Jednak taki układ nie zawsze sprawdzi się w systemach cechujących się relatywnie dużą zmianą parametrów obiektu, skutkując pogorszeniem właściwości dynamicznych i statycznych napędu [63]. W celu zapewnienia oczekiwanej odpowiedzi układu w szerokim zakresie zmian parametrów obiektu, niezbędne jest zastosowanie regulacji adaptacyjnej. Jest to struktura, która zmienia swoje zachowanie w odpowiedzi na zewnętrzne zakłócenia lub zmianę parametrów obiektu. Sterowanie adaptacyjne może być rozważane jako specjalny rodzaj sterowania z nieliniowym sprzężeniem zwrotnym, w którym można odseparować dwie skale czasowe procesu [11]. Szybsza, tj. wewnętrzna, jest związana z tradycyjnym sprzężeniem zwrotnym, a wolniejsza, tj. zewnętrzna, odpowiada za aktualizację współczynników regulatora. Zastosowanie regulatora adaptacyjnego pozwala na zwiększenie wydajności, odporności oraz funkcjonalności systemu. Z tego powodu, w ostatnich latach regulatory adaptacyjne były tematem wielu prac naukowych [12–21]. Najbardziej znanymi schematami adaptacji są: sterowanie ślizgowe, sterowanie z harmonogramem wzmocnienia, automatyczne strojenie oraz sterowanie adaptacyjne z modelem odniesienia [11].

2.4.1. Sterowanie z harmonogramem wzmocnienia

W wielu przypadkach dynamika procesu zmienia się wraz z jego punktem pracy. Jednym z możliwych przypadków jest nieliniowość obiektu, np. wysokość poziomu cieczy w zbiorniku o nieregularnym kształcie. W takim przypadku pomiar lub estymacja aktualnego punktu pracy może zostać wykorzystana do przestrajania współczynników regulatora w celu kompensacji

zmian parametrów procesu. Regulator sterujący obiektem jest zamknięty jedną pętlą sprzężenia zwrotnego, a pomiar lub estymacja punktu pracy obiektu i harmonogram wzmocnień regulatora jest zamknięty dodatkową pętlą sprzężenia zwrotnego odpowiedzialną za adaptację regulatora.

Sterowanie z harmonogramem wzmocnienia zastosowane dla regulatora PI prędkości kątowej sterującego hybrydowym silnikiem krokowym zostało zaproponowane w [64]. Rozwiązanie to bazuje na zwiększaniu współczynników regulatora, wykorzystując formułę bazującą m.in. na uchybie prędkości. Pozwala ono poprawić kompensację zakłóceń oraz zredukować oscylacje prędkości w porównaniu do stacjonarnego regulatora PI prędkości o zbliżonej dynamice. Zbadana została również odporność układu na zmiany momentu bezwładności. Badania wykazały, że zaproponowane rozwiązanie cechuje się większą odpornością niż regulator stacjonarny. W pracy [65] zaproponowano adaptacyjny regulator prądu dla przełączalnego silnika reluktancyjnego. Zastosowane sterowanie modyfikowało współczynniki regulatora PI w odpowiedzi na zmiany prądu i pozycji wirnika. Pozwoliło to na uzyskanie oczekiwanego kształtu oraz lepszej dynamiki odpowiedzi skokowej prądu w porównaniu do stacjonarnego regulatora PI prądu. Ponadto dzięki adaptacyjnemu doborowi parametrów regulatora, tętnienia momentu zostały również zmniejszone. Regulator PI prędkości z dodatkowym wzmocnieniem toru sprzężenia w przód dla napędu z silnikiem PMSM został zaproponowany w [66]. Bazując na aktualnym punkcie pracy, współczynniki w strukturze toru sprzężenia w przód są modyfikowane w celu poprawienia jakości regulacji w przypadku zmian parametrów silnika PMSM lub wystąpienia momentu obciążenia. Wykazano, że zaproponowana struktura regulacji z regulatorem adaptacyjnym wykazuje lepsze właściwości dynamiczne odpowiedzi skokowej w przypadku wystąpienia zewnętrznego zakłócenia, a także przy zmianach parametrów silnika PMSM (tj. rezystancji uzwojeń, indukcyjności w osiach d i q oraz momentu bezwładności).

2.4.2. Automatyczne strojenie regulatora

Automatyczny dobór współczynników regulatora zawsze wymaga przyjęcia założeń specyficznych dla danego rozwiązania. Z tego powodu znane metody strojenia regulatorów, np. metody Zieglera-Nicholsa dla regulatora PID lub reguła Brysona i optymalizacja liniowo-kwadratowa dla regulatora ze sprzężeniem od wektora zmiennych stanu, dostarczają przeważnie początkowe wartości współczynników, które wymagają późniejszej korekty dokonywanej najczęściej przez inżyniera procesu lub dodatkowych mechanizmów dostrajania [67].

Ze względu na złożoność procesów sterowania, niemożliwym okazało się opracowanie jednej, uniwersalnej metody doboru współczynników regulatora, pozwalającej na wyznaczenie ich optymalnych wartości. To w efekcie przyczyniło się do rozpoczęcia przez środowisko naukowe prac nad zaawansowanymi algorytmami przeznaczonymi do automatycznego strojenia regulatorów. W przeciwieństwie do większości mechanizmów adaptacji, w tym przypadku dodatkowa pętla sprzężenia zwrotnego jest zamknięta tylko podczas uruchamiania stanowiska, natomiast nie wymaga zdefiniowania początkowych współczynników regulatora. Konsekwencją omówionej powyżej zasady działania jest konieczność ponownej procedury automatycznego strojenia w przypadku zmian parametrów obiektu sterowania.

W pracy [36], zaproponowano metodę automatycznego strojenia napędu elektrycznego z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu. Algorytm wykorzystuje globalne algorytmy optymalizacyjne inspirowane przyrodą i funkcję celu pozwalającą uzyskać możliwie najlepszą dynamikę oraz kompensację momentu obciążenia, jednocześnie uwzględniającą problem minimalizacji błędu nadążania składowej prądu $i_d(t)$ oraz ograniczenie pochodnej sygnału sterującego. Otrzymane wyniki potwierdziły skuteczność zaproponowanego mechanizmu automatycznego strojenia, który nie wymaga definicji początkowych współczynników regulatora. Automatyczne strojenie regulatora ze sprzężeniem od wektora zmiennych stanu dla układu dwumasowego zostało zaproponowane w [58]. Ze względu na złożoną strukturę mechaniczną układu z występującą sprężystością, część mechaniczna ma charakter rezonansowy, co przekłada się na występowanie oscylacji prędkości kątowej. Zaproponowana metoda bazuje na algorytmie optymalizacyjnym sztucznej kolonii pszczoł, wykorzystującym funkcję celu uwzględniającą różnicę pomiędzy prędkością silnika a prędkością wyjściową układu. Uzyskane współczynniki pozwoliły na uzyskanie wysokiej dynamiki oraz zapewniły skuteczne tłumienie oscylacji. Metodę doboru współczynników regulatora dla predykcyjnego sterowania momentem elektromagnetycznym zaproponowano w [68]. Ta grupa algorytmów sterowania cechuje się wysoką dynamiką, jednak wymaga właściwego doboru współczynników. Wykorzystanie opracowanego mechanizmu automatycznego strojenia regulatora pozwoliło uzyskać optymalne sterowanie momentem oraz strumieniem silnika indukcyjnego. Zaproponowana funkcja celu została przystosowana specjalnie dla układów predykcyjnych, dzięki czemu uzyskano wysoką dynamikę w szerokim zakresie pracy napędu.

2.4.3. Sterowanie ślizgowe

Sterowanie ślizgowe jest nieliniową metodą sterowania z nieciągłym sygnałem sterującym. Nieciągłość sygnału wynika z przełączania pomiędzy ciągłymi strukturami regulacji w zależności od aktualnych wartości zmiennych stanu. Dzięki zastosowaniu sterowania ślizgowego możliwe jest uzyskanie wysokiej dynamiki oraz odporności na zmiany parametrów obiektu lub zewnętrzne zakłócenia.

Dyskretne sterowanie ślizgowe silnika indukcyjnego zostało zaproponowane w [14]. Opracowano dyskretny algorytm sterowania ślizgowego prędkości kątovej oraz strumienia. Omawiana metoda jest oparta o zmienną w czasie linię przełączeń, dzięki czemu uzyskano wysoką dynamikę i odporność na zewnętrzne zakłócenia, a także zmiany momentu bezwładności. Otrzymane wyniki symulacyjne i eksperymentalne potwierdzają odporność zaproponowanej struktury regulacji. W [69] sterowanie ślizgowe z linearyzacją ze sprzężeniem zwrotnym zostało zaproponowane dla napędu z silnikiem PMSM. Metoda wykorzystuje pojedynczą pętlę sprzężenia zwrotnego do regulacji prędkości i prądów. Ponadto zaproponowano nieliniowy obserwator momentu obciążenia wykorzystany do kompensacji zakłóceń metodą sprzężenia w przód. Zaproponowana metoda pozwoliła na uzyskanie wysokiej odporności na zakłócenia związane z momentem obciążenia oraz wysoką dynamikę w różnych warunkach pracy napędu.

2.4.4. Sterowanie adaptacyjne z modelem odniesienia

Zmiany parametrów obiektu lub zewnętrzne zakłócenie prowadzą do zmiany odpowiedzi układu regulacji. W przypadku wymaganej stałej dynamiki obiektu niezależnej od tych zmian należy zastosować sterowanie adaptacyjne z modelem odniesienia. Główną ideą jest zachowanie stałej odpowiedzi układu, co jest realizowane poprzez zdefiniowane modelu odniesienia i minimalizację uchybu pomiędzy wspomnianym modelem a rzeczywistą odpowiedzią układu. Za minimalizację tej różnicy odpowiedzialny jest mechanizm adaptacji.

Adaptacyjny regulator ze sprzężeniem od wektora zmiennych stanu dla układu dwumasowego został przedstawiony w [44]. W roli mechanizmu adaptacji wykorzystana została reguła Widrow'a-Hoff'a, która pierwotnie została opracowana do nauki modelu perceptronu AdaLiNe (ang. *adaptive linear neuron*). Omawiana metoda pozwala na zapewnienie stałej odpowiedzi skokowej regulatora ze sprzężeniem od wektora zmiennych stanu przy zmianach momentu bezwładności. Sterowanie adaptacyjne z modelem odniesienia bazujące na sztucznej sieci

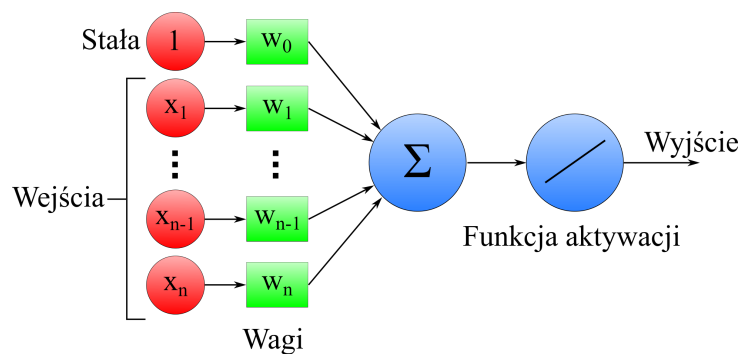
neuronowej dla silnika indukcyjnego zostało zaproponowane w [70]. Regulator neuronowy wykorzystujący radialne funkcje bazowe w połączeniu z mechanizmem adaptacji w torze regulacji prądów, pozwolił na uzyskanie stałości odpowiedzi w zmiennych warunkach pracy oraz kompensację nieliniowości struktury sterowania. Mechanizmem adaptacji była metoda wstecznej propagacji błędów. Nowatorski mechanizm adaptacji bazujący na algorytmie sztucznej kolonii pszczół został zaproponowany przez autora niniejszej rozprawy doktorskiej w [46]. Obiektem sterowania był napęd z silnikiem PMSM w trybie regulacji prędkości kątowej o zmiennym momencie bezładności, a adaptowaną strukturą sterowania regulator ze sprzężeniem od wektora zmiennych stanu. Zastosowanie algorytmu optymalizacyjnego w mechanizmie adaptacji pozwoliło na uzyskanie odporności na zmiany parametrów mechanicznych na odpowiedź skokową prędkości. Algorytm wykorzystał optymalizator liniowo-kwadratowy w celu wyznaczenia współczynników regulatora ze sprzężeniem od wektora zmiennych stanu dla wygenerowany przez algorytm sztucznej kolonii pszczół macierzy wagowych \mathbf{Q} i \mathbf{R} .

Niniejsza rozprawa doktorska skupia się na zastosowaniu sterowania adaptacyjnego z modelem odniesienia dla napędu z silnikiem PMSM w trybie regulacji prędkości kątowej.

3. Zastosowanie reguły Widrow'a-Hoff'a w mechanizmie adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu

3.1. Reguła Widrow'a-Hoff'a

Reguła Widrow'a-Hoff'a została zaproponowana w 1960 roku jako metoda uczenia adaptacyjnej maszyny klasyfikującej, zwanej AdaLiNe [71]. Jest ona powszechnie znana jako metoda najmniejszych kwadratów (ang. *least mean squares*). Schemat blokowy maszyny klasyfikującej AdaLiNe został przedstawiony na rysunku 3.1.



Rysunek 3.1. Schemat blokowy maszyny klasyfikującej AdaLiNe

Funkcja aktywacji AdaLiNe, jak sama nazwa wskazuje, jest funkcją liniową $y(x) = x$. Przyjmując oznaczenie stałej (tzw. bias) jako wejście o zerowym indeksie ($x_0 = 1$), równanie opisujące wyjście maszyny klasyfikującej (y) jest następujące:

$$y = w_0 \cdot 1 + w_1 x_1 + \dots + w_{n-1} x_{n-1} + w_n x_n = \sum_{i=0}^n w_i x_i \quad (3.1)$$

Reguła uczenia minimalizuje błąd średniokwadratowy pomiędzy wartością oczekiwaną (d) a wartością wyjściową:

$$E(\mathbf{w}) = \frac{1}{2} e^2 = \frac{1}{2} (d - y)^2 \quad (3.2)$$

Aktualizacja wag modelu AdaLiNe bazuje na metodzie optymalizacji gradientowej najszybszego spadku i wyraża się następującym równaniem:

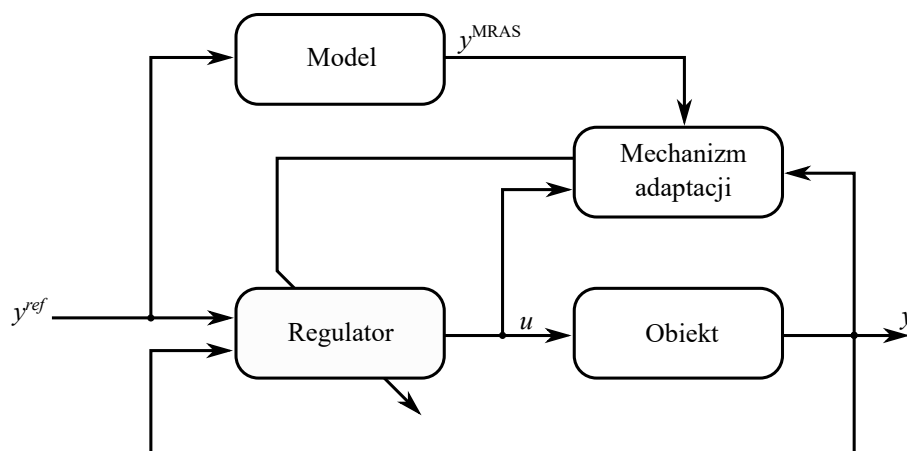
$$w_j(k+1) = w_j(k) + \mu \frac{\partial E(w)}{\partial w_j} \quad (3.3)$$

gdzie: $w_j(k+1)$ i $w_j(k)$ są wagami j -tego wejścia odpowiednio w $k+1$ i k iteracji uczenia, μ to współczynnik uczenia. Wykorzystując równania (3.1)-(3.3) uzyskuje się wzór na aktualizację wagi j -tego wejścia:

$$w_j(k+1) = w_j(k) - \mu e(k)x_j(k) \quad (3.4)$$

3.2. Mechanizm adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu

W celu zastosowania reguły Widrow'a-Hoff'a jako mechanizmu adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu w napędzie elektrycznym z silnikiem PMSM o zmiennym momencie bezwładności, należy zdefiniować wartość oczekiwaną prędkości kątowej silnika w aktualnej chwili czasu oraz prawo adaptacji. Jak zostało zaznaczone w rozdziale 2, w niniejszej rozprawie doktorskiej skupiono się na zastosowaniu sterowania adaptacyjnego z modelem odniesienia, którego schemat blokowy przedstawia rysunek 3.2. W związku z tym, wartość oczekiwana prędkości kątowej jest zdefiniowana przez model odniesienia.



Rysunek 3.2. Schemat blokowy sterowania adaptacyjnego z modelem odniesienia

3.2.1. Model odniesienia

Model odniesienia odpowiada za oczekiwany kształt odpowiedzi na zadany sygnał referencyjny. W związku z tym, jest on ekwiwalentem odpowiedzi obiektu z regulatorem w zamkniętej pętli sprzężenia zwrotnego. W analizowanym zlinearyzowanym modelu silnika PMSM (równanie (2.21)) moment elektromagnetyczny generowany przez silnik zależy tylko od prądu w osi q . Z tego powodu prędkość kątowna silnika jest niezależna od prądu w osi d . Chcąc oszacować potrzebny rząd transmitancji zastępczej należy uwzględnić część elektryczną w osi q , część mechaniczną oraz regulator ze sprzężeniem od wektora zmiennych stanu z wewnętrznym modelem sygnału referencyjnego. Wykorzystując zlinearyzowany model silnika PMSM przedstawiony w formie równania stanu w (2.19) oraz prawo sterowania regulatora ze sprzężeniem od wektora zmiennych stanu z równania 2.23, opracowano schemat blokowy zamkniętej pętli sprzężenia zwrotnego. W związku z analizowanym silnikiem PMSM o zmiennym momencie bezwładności, wprowadzono oznaczenie J_{nom} w celu zwiększenia czytelności, jaka wartość momentu bezwładności jest analizowana. Schemat początkowy oraz kolejne etapy jego przekształcenia zostały przedstawione na rysunku 3.3. Wynikiem podjętych działań jest transmitancja pomiędzy sygnałem referencyjnym prędkości a odpowiedzią obiektu z układem regulacji:

$$G_{\omega}(s) = \frac{\Omega_m(s)}{\Omega_m^{ref}(s)} = \frac{a_0}{b_3s^3 + b_2s^2 + b_1s + b_0} \quad (3.5)$$

przy czym, zgodnie z oznaczeniami z rysunku 3.3:

$$a_0 = b_0 = K_t K_p k_{\omega 2}$$

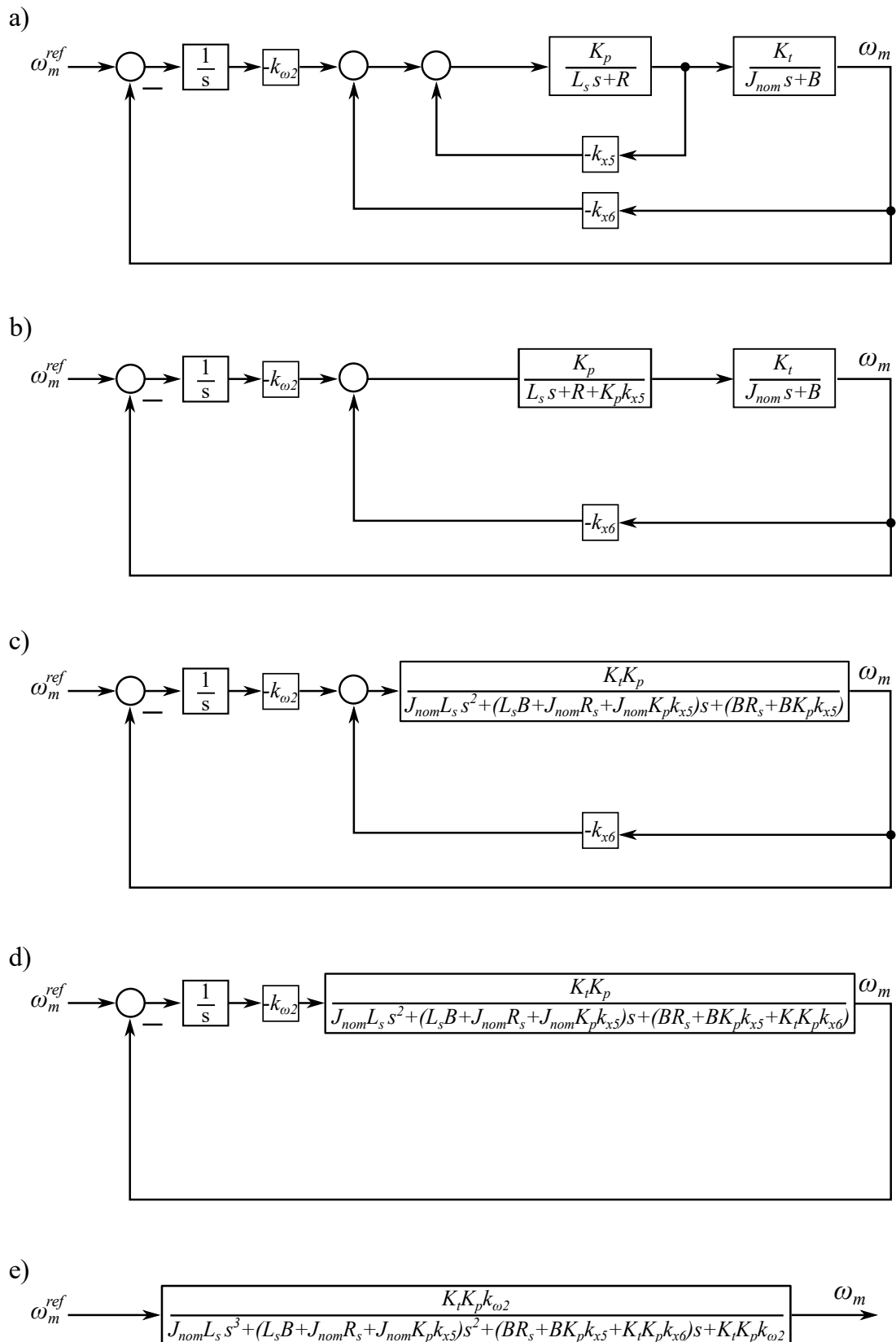
$$b_1 = BR_s + BK_p k_{x5} + K_t K_p k_{x6}$$

$$b_2 = L_s B + J_{nom} R_s + J_{nom} K_p k_{x5}$$

$$b_3 = J_{nom} L_s$$

Przechodząc na stałe czasowe części elektrycznej (T_e) i mechanicznej (T_m) oraz wzmocnienia statyczne części elektrycznej (k_e) i mechanicznej (k_m), otrzymuje się następującą postać:

$$G_{\omega}(s) = \frac{k_e k_m k_{\omega 2}}{T_e T_m s^3 + (T_e + T_m + T_m k_e k_{x5}) s^2 + (k_e k_{x5} + k_e k_m k_{x6} + 1) s + k_e k_m k_{\omega 2}} \quad (3.6)$$



Rysunek 3.3. Wyprowadzenie transmitancji układu zamkniętego z regulatorem stanu dla prędkości kątowej napędu z silnikiem PMSM

przy czym:

$$T_e = \frac{L_s}{R_s}, \quad T_m = \frac{J_{nom}}{B}, \quad k_e = \frac{K_p}{R_s}, \quad k_m = \frac{K_t}{B}$$

Ze względu na znaczną różnicę pomiędzy stałymi czasowymi części elektrycznej i mechanicznej silnika ($T_e \ll T_m$), bardzo często na etapie modelowania napędu elektrycznego pomija się dynamikę pętli prądowej (tj. $T_e = 0$) [44]. Dzięki temu uzyskuje się znacznie prostszy model z niewielką lub pomijalnie małą różnicą pomiędzy rzeczywistą a zamodelowaną odpowiedzią prędkości kątowej silnika elektrycznego. Stosując wspomniane uproszczenie w analizowanym napędzie elektrycznym z silnikiem PMSM i regulatorem stanu, transmitancja wyrażona równaniem (3.6) upraszcza się do następującej postaci:

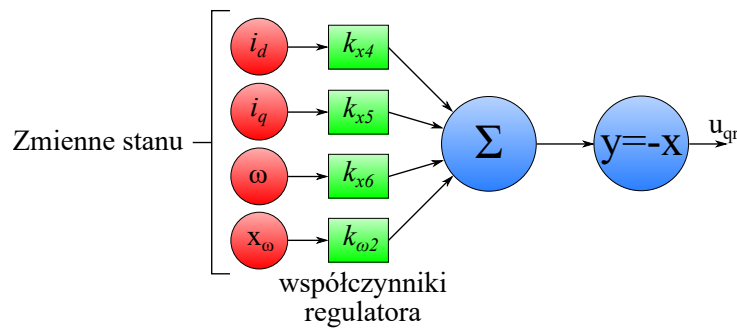
$$G_{\omega}^*(s) = \frac{k_e k_m k_{\omega 2}}{(T_m + T_m k_e k_{x5})s^2 + (k_e k_{x5} + k_e k_m k_{x6} + 1)s + k_e k_m k_{\omega 2}} \quad (3.7)$$

Podsumowując, model odniesienia powinien mieć charakter transmitancji drugiego rzędu w celu zapewnienia możliwej do uzyskania odpowiedzi przez rzeczywisty układ. Jest to niezmiernie istotne ze względu na fakt, że reguła Widrow'a-Hoff'a aktualizuje współczynniki regulatora w każdej pętli regulacji. W przypadku przyjęcia modelu odniesienia o nieadekwatnym rzędzie, układ może nie być w stanie zredukować uchybu pomiędzy odpowiedzią układu a odpowiedzią modelu odniesienia do zera, tj. przyjęcie modelu inercyjnego I rzędu w przypadku, gdy obiekt ma charakter obiektu wyższego rzędu spowoduje, że będzie oczekiwał dużej stromości odpowiedzi w początkowej fazie odpowiedzi skokowej, która jest nieosiągalna. Niezerowy uchyb będzie skutkował nieustanną aktualizacją współczynników regulatora. Rezultatem takiego niedopasowania może być brak uogólniania, tj. oscylacyjny charakter odpowiedzi układu lub nawet utrata jego stabilności [20]. Oba z nich są niedopuszczalne, więc odpowiednia analiza układu jest kluczowa do poprawnego działania reguły Widrow'a-Hoff'a.

3.2.2. Prawo adaptacji

Celem zastosowania sterowania adaptacyjnego z modelem odniesienia jest utrzymanie odpowiedzi obiektu takiej samej jak odpowiedzi modelu niezależnie od zmian parametrów obiektu. W tym celu niezbędny jest mechanizm adaptacji współczynników regulatora pozwalający zminimalizować różnicę pomiędzy odpowiedzią rzeczywistego obiektu a odpowiedzią modelu odniesienia.

W ostatnich latach reguła Widrow'a-Hoff'a była stosowana jako mechanizm adaptacji w adaptacyjnym regulatorze neuronowym [72]. Podobieństwo strukturalne regulatora ze sprzężeniem od wektora zmiennych stanu oraz perceptronu (patrz rysunek 3.1 oraz rysunek 3.4) spowodowało zastosowanie reguły Widrow'a-Hoff'a w adaptacyjnej wersji tego regulatora [44, 73].



Rysunek 3.4. Regulator ze sprzężeniem od wektora zmiennych stanu odpowiedzialnego za wyznaczenie sygnału sterującego w osi q wirującego układu współrzędnych

Formuła (3.2) dostosowana do adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu silnika PMSM przyjmuje następującą postać:

$$E(k_{x5}, k_{x6}, k_{\omega2}) = \frac{1}{2} (e_{\omega \text{ MRAS}})^2 = \frac{1}{2} (\omega_m^{\text{MRAS}} - \omega_m)^2 \quad (3.8)$$

gdzie: ω_{MRAS} to model odniesienia. Wyprowadzając formułę na aktualizację współczynników regulatora ze sprzężeniem od wektora zmiennych stanu stosując regułę Widrow'a-Hoff'a daną równaniem (3.3), otrzymujemy:

$$k_{x5}(k+1) = k_{x5}(k) + \mu \frac{\partial E(k_{x5}, k_{x6}, k_{\omega2})}{\partial k_{x5}} \quad (3.9)$$

$$k_{x6}(k+1) = k_{x6}(k) + \mu \frac{\partial E(k_{x5}, k_{x6}, k_{\omega2})}{\partial k_{x6}} \quad (3.10)$$

$$k_{\omega2}(k+1) = k_{\omega2}(k) + \mu \frac{\partial E(k_{x5}, k_{x6}, k_{\omega2})}{\partial k_{\omega2}} \quad (3.11)$$

Pochodna cząstkowa błędu średniokwadratowego po współczynnikach regulatora ze sprzężeniem od wektora zmiennych stanu jest zależna od parametrów napędu elektrycznego z silnikiem PMSM. W literaturze można znaleźć różne uproszczenia dotyczące wyznaczania pochodnych cząstkowych. Na przykład w regulatorach neuronowych dla napędu z silnikiem PMSM stosuje

się funkcję znaku tej pochodnej [72, 74]:

$$w_{ij}(k+1) = w_{ij}(k) + \mu_{ij}(k) \cdot \text{sign} \left(\frac{\partial E(w)}{\partial w_j} \right) \quad (3.12)$$

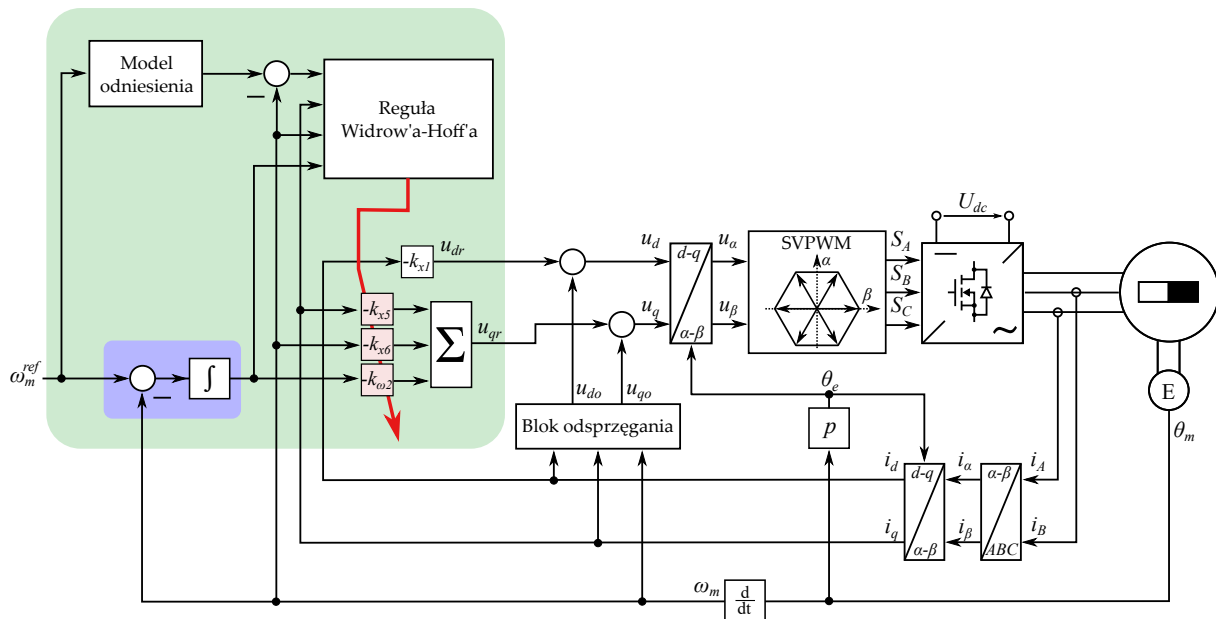
gdzie: funkcja $\text{sign}(x)$ zwraca znak argumentu wejściowego x , μ_{ij} jest indywidualnym współczynnikiem uczenia dla każdego połączenia w sieci neuronowej. Wspomniane rozwiązanie pozwala wyznaczyć pochodną w sposób numeryczny, a funkcja znaku wyeliminuje wpływ wzmacniania szumów pomiarowych, spowodowany przez różniczkowanie numeryczne, na wartość korekcji wagi neuronu. Innym podejściem, które zostało wykorzystane w niniejszej rozprawie doktorskiej jest założenie, że odpowiedź obiektu jest funkcją liniową, dzięki czemu otrzymujemy wzory na aktualizację wag analogiczną z równaniem (3.4) [44]. Podejście pozwala na zastosowanie jednej wartości współczynnika uczenia w przeciwieństwie do wcześniej wspomnianej metody. Wykorzystując to założenie oraz wzory (3.4), (3.8) i (3.9)-(3.11) otrzymano równania opisujące korektę współczynników regulatora ze sprzężeniem od wektora zmiennych stanu:

$$k_{x5}(k+1) = k_{x5}(k) - \mu e_{\omega} \text{MRAS}(k) i_q(k) \quad (3.13)$$

$$k_{x6}(k+1) = k_{x6}(k) - \mu e_{\omega} \text{MRAS}(k) \omega(k) \quad (3.14)$$

$$k_{\omega 2}(k+1) = k_{\omega 2}(k) - \mu e_{\omega} \text{MRAS}(k) x_{\omega}(k) \quad (3.15)$$

Otrzymane równania pozwalają na cyfrową implementację mechanizmu adaptacji. Wartość współczynnika w kolejnej dyskretnej próbce czasu jest zależna od: (i) wartości w aktualnej dyskretnej próbce, (ii) różnicy pomiędzy wartością aktualną prędkości a odpowiedzią modelu odniesienia, (iii) zmiennej stanu związanej z adaptowanym współczynnikiem regulatora ze sprzężeniem od wektora zmiennych stanu oraz (iv) współczynnika adaptacji. Schemat blokowy adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego na regule Widrow'a-Hoff'a został przedstawiony na rysunku 3.5.



Rysunek 3.5. Schemat blokowy adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego na regule Widrow'a-Hoff'a

3.3. Wyniki badań symulacyjnych

Opracowany model symulacyjny adaptacyjnego regulatora prędkości bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu oraz regule Widrow'a-Hoff'a dla napędu elektrycznego z silnikiem PMSM o zmiennym momencie bezwładności został przedstawiony na rysunku 3.6. Model symulacyjny ze stacjonarnym regulatorem ze sprzężeniem od wektora zmiennych stanu przedstawiony na rysunku 2.4 został rozszerzony o model odniesienia oraz regułę Widrow'a-Hoff'a dla każdego z adaptowanych współczynników. Ponadto zaimplementowano całkowity wskaźnik jakości z wartości bezwzględnej błędu, który został ujęty w błękitnym bloku. Warto zaznaczyć, że jest on liczony względem sygnału modelu odniesienia, a nie, jak klasycznie się go stosuje, względem sygnału referencyjnego. Pozwala to na bardzo prostą interpretację jakości odwzorowania odpowiedzi układu do modelu odniesienia, tj. wartość 0 oznacza perfekcyjne dopasowanie, do którego dążymy, stosując sterowanie adaptacyjne z modelem odniesienia. Wskaźnik jakości obliczany jest dla pełnego okresu sygnału referencyjnego, tj. skokowej zmiany sygnału referencyjnego prędkości od 0 do 10 rad/s, a następnie od 10 do 0 rad/s. Dzięki temu można w sposób graficzny przedstawić jakość dopasowania w zdyskretyzowanej co 1 okres sygnału referencyjnego dziedzinie czasu.

3.3.1. Implementacja modelu odniesienia

Model odniesienia w modelach symulacyjnych z regulatorami adaptacyjnymi został zaimplementowany jako transmitancja wyrażona równaniem (3.7). W celu uproszczenia jej zapisu w modelu przyjęto następującą formę:

$$G_{\text{MRAS}}(s) = \frac{\omega_{\text{MRAS}}(s)}{\omega_m^{\text{ref}}(s)} = \frac{a}{b_2 s^2 + b_1 s + b_0} \quad (3.16)$$

gdzie:

$$a = b_0 = k_e k_m k_{\omega 2}, \quad b_1 = k_e k_{x5} + k_e k_m k_{x6} + 1, \quad b_2 = T_m + T_m k_e k_{x5}$$

Warto zaznaczyć, że występujące w powyższych równaniach współczynniki regulatora przyjmują wartości nominalne przedstawione w tabeli 2.1.

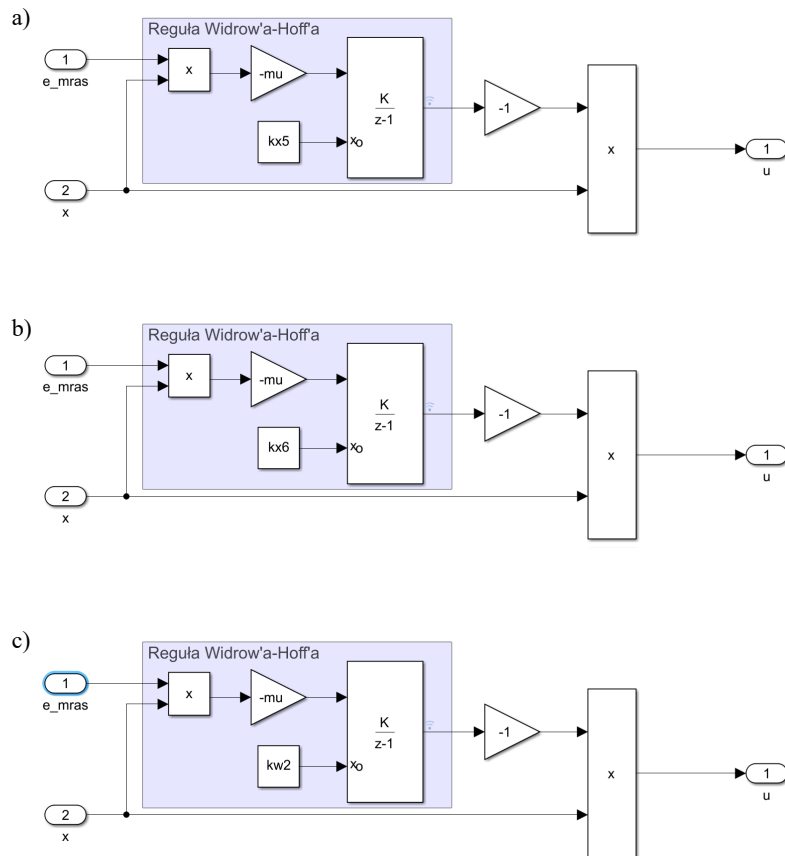
3.3.2. Implementacja mechanizmu adaptacji

Mechanizm adaptacji wyrażony równaniami (3.13)-(3.15) został zaimplementowany, wykorzystując gotowe bloki dostępne w programie MATLAB/SIMULINK. W celu cyklicznej aktualizacji wagi został wykorzystany akumulator o zadanym okresie próbkowania (T_s) oraz wartości początkowej równej dobranemu współczynnikowi dla nominalnych parametrów regulatora. Opracowane modele symulacyjne adaptacyjnych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu bazujących na regule Widrow'a-Hoff'a zostały przedstawione na rysunku 3.7.

3.3.3. Analiza wpływu współczynnika uczenia na jakość adaptacji

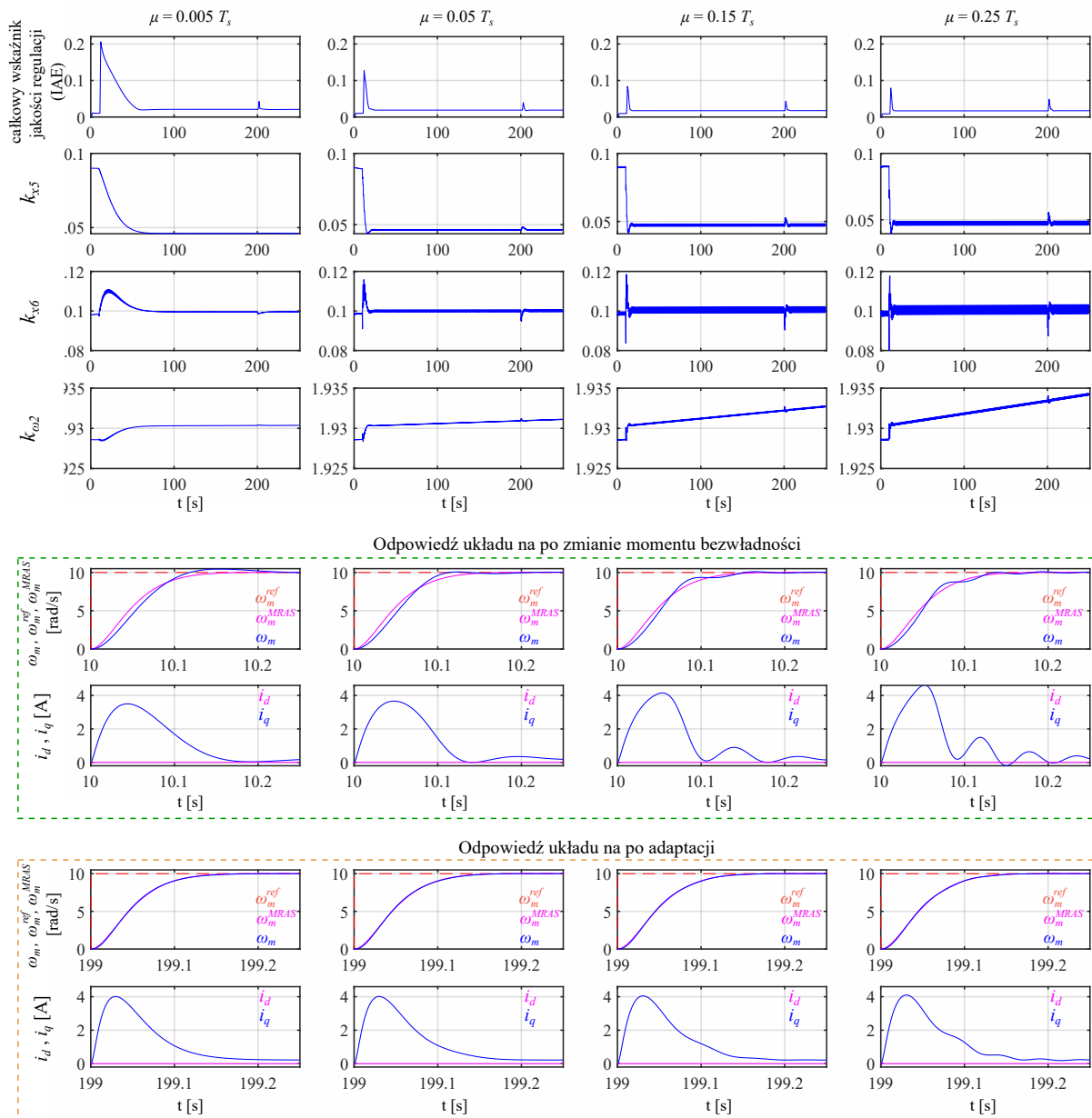
Reguła Widrow'a-Hoff'a posiada tylko jeden parametr, który należy odpowiednio dobrać. Jest nim współczynnik adaptacji (μ). W celu analizy jego wpływu na jakość adaptacji przygotowano następujący eksperyment:

- układ zaczyna pracę z nominalnym momentem bezwładności, co skutkuje wysokim dopasowaniem odpowiedzi układu do modelu odniesienia,
- w dziesiątej sekundzie następuje skokowa zmiana momentu bezwładności o wartość dodatkowego momentu bezwładności (J_{add}),
- w 200.25 sekundzie występuje zewnętrzne zakłócenie w postaci momentu obciążenia wynoszącego 1 Nm i trwa ono do końca eksperymentu,
- czas symulacji to 250 s.



Rysunek 3.7. Model symulacyjny mechanizmu adaptacji współczynników k_{x5} (a), k_{x5} (b) oraz $k_{\omega 2}$ (c) regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na regule Widrow'a-Hoff'a opracowany w programie MATLAB/SIMULINK

Cztery wartości współczynnika adaptacji zostały porównane w formie całkowego wskaźnika jakości oraz adaptacyjnych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu w dziedzinie czasu na rysunku 3.8. Do momentu wystąpienia zwiększenia momentu bezwładności wskaźnik całkowity IAE miał wartość bliską zero. Jego niezerowa wartość wynika z uproszczonego modelu odniesienia, tj. zredukowanego rzędu poprzez pominięcie stałej czasowej elektrycznej. W chwili wystąpienia zmiany parametru układu wskaźnik całkowity przyjmuje szczytową wartość, a następnie jest redukowany dzięki zastosowaniu mechanizmu adaptacji. Jak było wspomniane w rozdziale 3, algorytm Widrow'a-Hoff'a bazuje na gradiencie funkcji. Jest to widoczne na całkowym kryterium jakości IAE, które z biegiem czasu jest redukowane. Jedyne moment, w którym ów wskaźnik rośnie jest spowodowane wystąpieniem skokowego momentu obciążenia. Natomiast podczas pracy ze stałym obciążeniem całkowite kryterium jakości IAE nie zmieniło się w zauważalny sposób. Przedstawione zbrocza narastające po zmianie momentu bezwładności układu oraz po zakończonej adaptacji wskazują



Rysunek 3.8. Wpływ współczynnika adaptacji na dynamikę mechanizmu adaptacji bazującego na regule Widrow'a-Hoff'a

na problem występujący przy wysokich wartościach współczynnika uczenia. Po zmianie momentu bezwładności współczynniki adaptacji $\mu = 0.15T_s$ oraz $\mu = 0.25T_s$ układ oscylował wokół odpowiedzi modelu odniesienia. Ponadto po adaptacji uzyskana odpowiedź dobrze odzwierciedla odpowiedź modelu odniesienia, lecz posiada delikatne oscylacje, co jest bardzo dobrze widoczne na przebiegach prądów w wirującym układzie współrzędnych $d-q$. Powodem jest fakt, że układ próbuje dopasować odpowiedź do modelu odniesienia w każdej pętli regulacji i przy zbyt dużych wartościach współczynnika uczenia przestaje on generalizować rozwiązanie z punktu widzenia pełnego okresu sygnału referencyjnego. Problem generalizacji

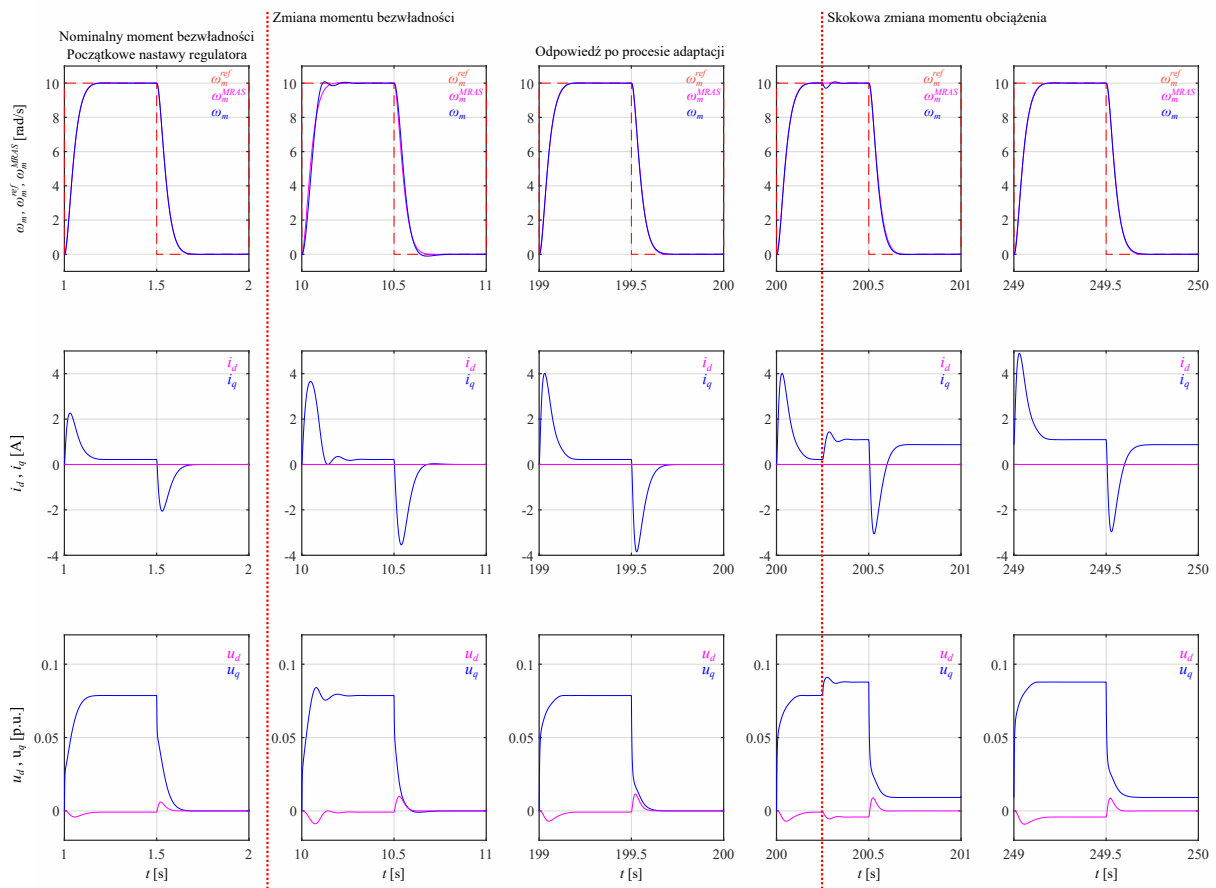
jest bardzo dobrze znany w uczeniu sieci neuronowych, z których reguła Widrow'a-Hoff'a się wywodzi. Mówimy o nim, gdy sieć zaczyna dopasowywać się do drugorzędnych szczegółów zbioru danych uczących, które nie mają istotnego znaczenia z punktu widzenia istotnych cech rozwiązywanego zadania [75]. Na rysunku 3.9 zostały przedstawione przebiegi czasowe prędkości kątowej, prądów oraz sygnałów sterujących dla pięciu wybranych okresów sygnału referencyjnego dla współczynnika uczenia równego $\mu = 0.05T_s$. Odpowiedź układu na pierwszy okres sygnału referencyjnego po zmianie momentu bezwładności charakteryzuje się znaczącymi oscylacjami w stanie przejściowym i dłuższym czasem narastania niż oczekiwana odpowiedź modelu odniesienia. Po adaptacji różnice w odpowiedziach nie są zauważalne, więc można stwierdzić, że algorytm w zakładanym czasie dopasował odpowiedź układu do aktualnych warunków pracy napędu, tj. zwiększony moment bezwładności. Wystąpienie skokowej zmiany momentu obciążenia spowodował odchylenie prędkości kątowej silnika od wartości zadanej, praca pod stałym obciążeniem nie wprowadzała fluktuacji w dynamice prędkości kątowej.

3.3.4. Implementacja modelu odniesienia o nieadekwatnym rzędzie

Jak wspomniano we wcześniejszym rozdziale, odpowiednio zdefiniowany model odniesienia odgrywa istotną rolę w syntezie regulatora adaptacyjnego bazującego na regule Widrow'a-Hoff'a. W celu zaprezentowania problemów z tym algorytmem, podczas gdy zaimplementowany zostanie model odniesienia o nieadekwatnym rzędzie, wykonano badania układu dla modelu odniesienia zdefiniowanego jako inercja pierwszego rzędu o wzmocnieniu statycznym równym 1 oraz stałej czasowej 0.0568 s:

$$G_{\omega}^{**}(s) = \frac{1}{0.0568s + 1} \quad (3.17)$$

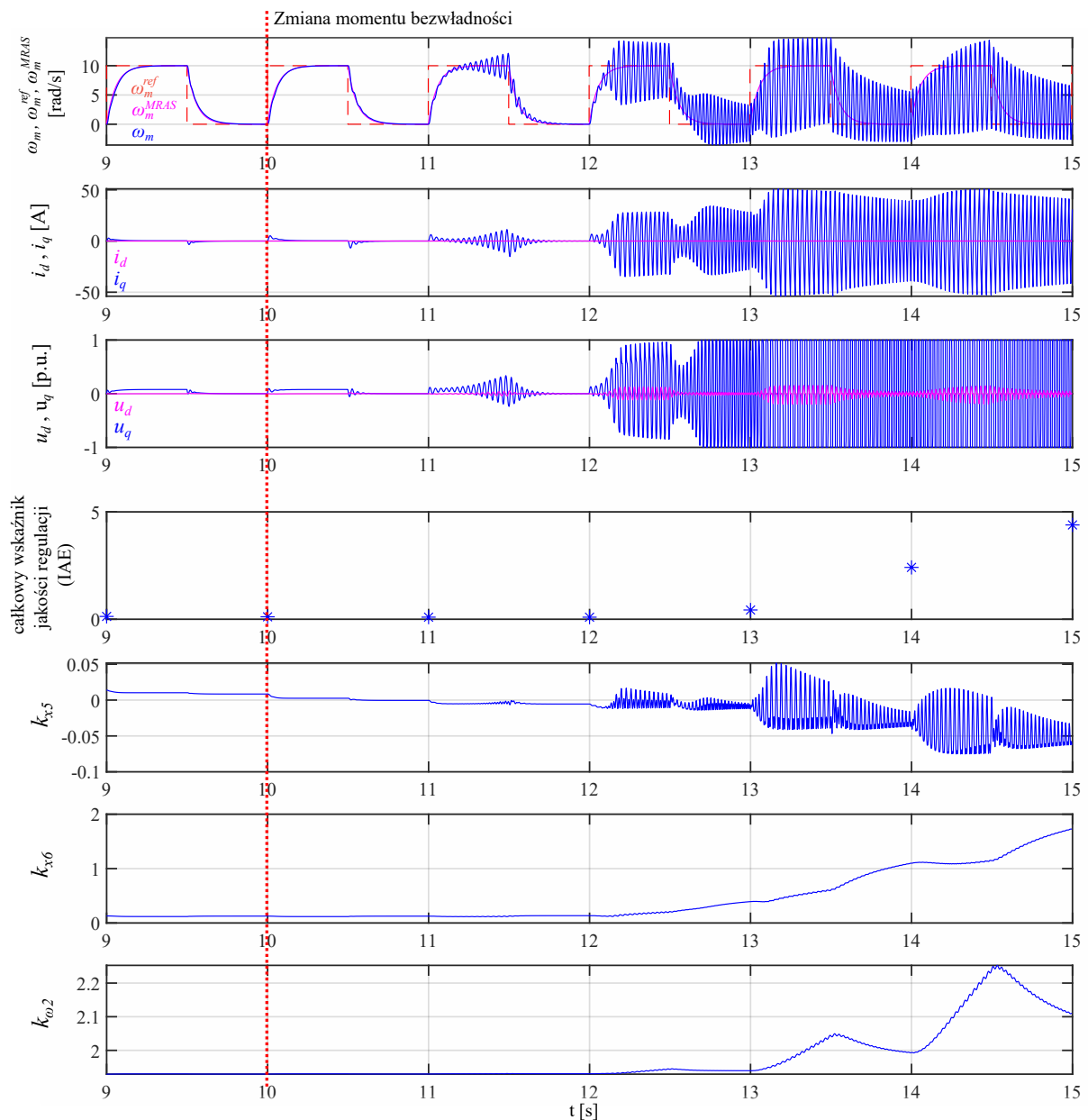
Stała czasowa została odczytana z odpowiedzi układu dla nominalnych parametrów jako czas potrzebny na osiągnięcie przez układ prędkości równej 0.632 wartości w stanie ustalonym. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących, jak również całkowity wskaźnik jakości regulacji i adaptowane współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu zostały przedstawione na rysunku 3.10. Współczynnik uczenia reguły Widrow'a-Hoff'a wynosił $\mu = 0.05T_s$. Reakcja układu na zmianę momentu bezwładności przy nieadekwatnym rzędzie modelu odniesienia skutkuje bardzo dużymi oscylacjami prędkości kątowej już w drugim okresie sygnału referencyjnego. Ponadto oscylacje składowej



Rysunek 3.9. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$

q prądu sięgają wartości ± 50 A, co w przypadku analizowanej pracy napędu w zakresie ± 6 A mogłoby uszkodzić przekształtnik energoelektroniczny lub/i silnik PMSM. Pomimo, że celem reguły Widrow'a-Hoff'a jest minimalizacja różnicy pomiędzy odpowiedzią rzeczywistą a odpowiedzią modelu odniesienia, to przez zdefiniowanie modelu jako inercję pierwszego rzędu spowodowało, że mechanizm adaptacji próbuje uzyskać nieosiągalną dynamikę układu, ponieważ wyprowadzony model odniesienia dla analizowanego układu w rozdziale 3 jest drugiego rzędu. Różnica pomiędzy tymi odpowiedziami skutkuje ciągłą adaptacją i próbą jej zniwelowania. Warto zaznaczyć, że całkowity wskaźnik jakości regulacji zwiększa się, ponieważ mechanizm zaczyna błędnie generalizować kształt oczekiwanej odpowiedzi i próbuje dopasować się do nieznaczących różnic pomiędzy odpowiedzią modelu odniesienia a odpowiedzią układu.

Z przedstawionych wyników badań można wywnioskować, że czas adaptacji jest proporcjonalny do współczynnika uczenia. Wyższa wartość pozwala przyspieszyć proces



Rysunek 3.10. Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowane współczynniki regulatora oraz wybrane przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q ukazujące reakcję układu na zmianę momentu bezwładności przy nieadekwatnym rzędzie modelu odniesienia. Mechanizm adaptacji bazujący na regule Widrwa-Hoffa ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$

adaptacji, lecz w przypadku zbyt wysokich wartości może doprowadzić do oscylacyjnego charakteru odpowiedzi układu lub nawet do utraty stabilności [73]. Takie same problemy mogą być wynikiem nieodpowiednio zdefiniowanego modelu odniesienia. W obu przypadkach głównym powodem jest przeuczanie się algorytmu i próba dopasowania się do chwilowych wartości odpowiedzi i słabe uogólnianie w celu uzyskania kształtu oczekiwanej odpowiedzi. Związane jest to z aktualizacją wag w każdej pętli regulacji. Pomimo, że możliwe jest uzyskanie

stosunkowo krótkiego czasu adaptacji na poziomie nawet kilkunastu sekund, to współczynnik uczenia musi zostać znacznie zredukowany w celu uniknięcia oscylacyjnego charakteru odpowiedzi układu. Przebieg wolny od oscylacji uzyskano dla odpowiednio zdefiniowanego modelu odniesienia oraz dla $\mu = 0.05T_s$, dla którego czas adaptacji można oszacować na 20 s, podczas gdy stosunkowo dobre odzwierciedlenie odpowiedzi modelu odniesienia występuje już po 10-tej sekundzie adaptacji.

3.4. Podsumowanie

W niniejszym rozdziale przedstawiono zastosowanie reguły Widrow'a-Hoff'a w mechanizmie adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu oraz wyniki badań symulacyjnych. Algorytm z powodzeniem realizuje proces adaptacji współczynników regulatora. Reguła Widrow'a-Hoff'a posiada tylko jeden parametr, który należy dobrać, lecz już w symulacji zauważono efekt przeuczania się regulatora dla zbyt dużych współczynników, który powodował oscylacyjny charakter odpowiedzi układu. Przedstawiono również problemy reguły Widrow'a-Hoff'a w przypadku implementacji modelu odniesienia o nieadekwatnym rzędzie. Oba elementy tego mechanizmu adaptacji mają kluczowe znaczenie nie tylko ze względu na możliwość minimalizacji czasu zbieżności czy końcowej jakości odpowiedzi układu, lecz również pod kątem poprawnego działania algorytmu, ponieważ oscylacyjny charakter odpowiedzi nasila się ze wzrostem współczynnika uczenia lub zastosowania modelu odniesienia, którego układ nie jest w stanie odwzorować.

4. Zastosowanie algorytmów optymalizacyjnych w mechanizmie adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu

Sterowanie adaptacyjne można rozpatrywać jako zadanie optymalizacyjne. W rozważanym zagadnieniu sterowania adaptacyjnego z modelem odniesienia optymalizacja dotyczy minimalizacji różnicy pomiędzy aktualną odpowiedzią układu a modelem odniesienia. W tak postawionym problemie zauważono, że mechanizm adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu może bazować na algorytmie optymalizacyjnym, jeżeli zostanie on odpowiednio dostosowany do problemu. Algorytmy optymalizacyjne bazują na porównywaniu aktualnego rozwiązania z nowym, przeważnie będącym jego modyfikacją. Wybór lepszego rozwiązania polega na porównaniu wartości funkcji celu, jaka została zdefiniowana. Rozważana funkcja celu dla sterowania adaptacyjnego z modelem odniesienia bazująca na wskaźniku jakości układu regulacji jakim jest całka z wartości bezwzględnej uchybu (ang. *integral absolute error*, IAE) jest następująca:

$$f(k_{x5}, k_{x6}, k_{\omega 2}) = \int_0^T |\omega_m^{\text{MRAS}}(\tau) - \omega_m(\tau)| d\tau \quad (4.1)$$

Warto zaznaczyć, że jest to optymalizacja jednokryterialna bez ograniczeń. Ta informacja jest niezmiernie istotna pod względem doboru odpowiedniego algorytmu optymalizacyjnego.

4.1. Algorytmy optymalizacyjne

Zagadnienie optymalizacji jest powszechnie znane, a problemy optymalizacyjne prawdopodobnie występują we wszystkich aspektach inżynierskich. W zależności od złożoności problemu, wymagane są mniej lub bardziej złożone algorytmy do ich rozwiązania. Główny ich podział jest związany z przeszukiwaną przestrzenią rozwiązań, tj. optymalizacja lokalna lub globalna. Innym ważnym podziałem jest wymóg obliczania gradientu funkcji celu,

w tym podziale możemy wyróżnić metody gradientowe i bezgradientowe. W rozważanym przypadku zastosowania algorytmu optymalizacyjnego jako mechanizm adaptacji dla regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu z silnikiem PMSM, metody gradientowe zostały odrzucone ze względu na występowanie szumów w mierzonych sygnałach i błędy kwantyzacji, które przyczyniają się do niedokładności wyznaczania pochodnej funkcji. Wybór pomiędzy algorytmem optymalizacyjnym lokalnym a globalnym nie jest tak oczywisty jak w przypadku poprzedniego podziału. W ostatnich latach, algorytmy optymalizacyjne inspirowane przyrodą zyskały popularność w zagadnieniu optymalizacji globalnej ze względu na dobrą eksplorację całej przestrzeni rozwiązań i dobrą eksploatację w otoczeniu najlepszego rozwiązania. Ponadto warto przytoczyć jeden z wniosków pracy D. Wolpert i W. Macready z 1997 roku zatytułowanej "*No free lunch theorems for optimization*" [76]:

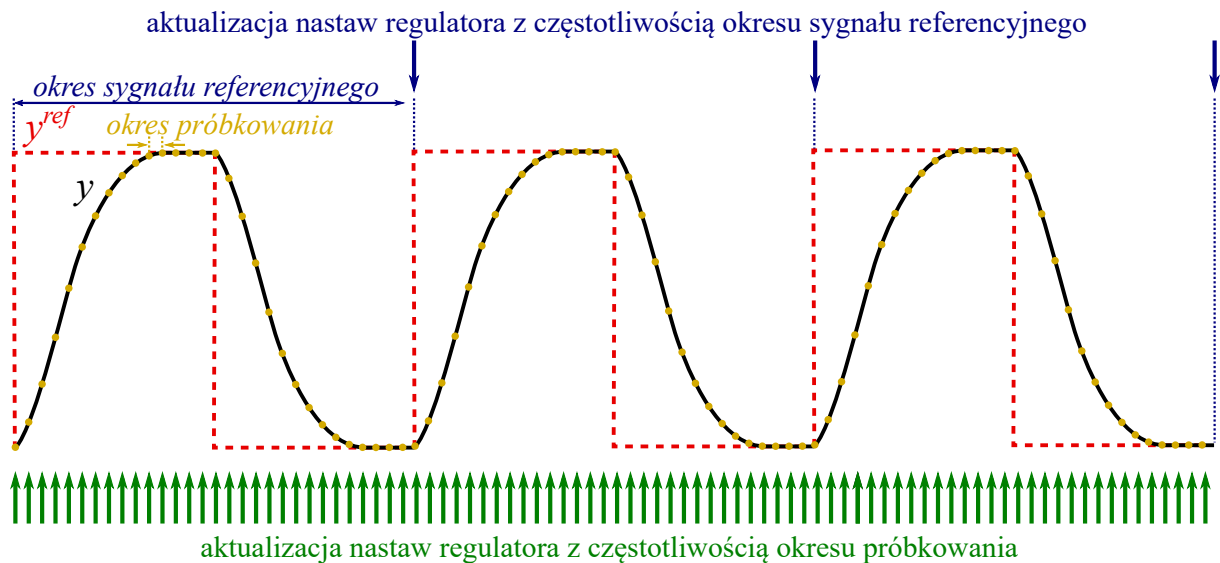
Algorytm może przewyższać inny w rozwiązaniu problemu, jeśli żaden z nich nie jest wyspecjalizowany w tym problemie – D. Wolper et. al.

Oznacza to, że porównując dwa algorytmy, jeden z nich może szybciej znaleźć rozwiązanie. W przypadku zastosowania algorytmów optymalizacyjnych do adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu, oznacza to, że odpowiedni wybór algorytmu pozwoli skrócić czas adaptacji lub/i poprawić jakość dopasowania do modelu odniesienia. W związku z tym, algorytm optymalizacyjny inspirowany przyrodą zostanie porównany z lokalnym, bezgradientowym algorytmem optymalizacyjnym. Wybór algorytmów został dokonany bazując na ich rozpowszechnieniu w literaturze i jest on następujący: optymalizacja rojem cząstek (ang. *particle swarm optimization*, PSO) jako algorytm optymalizacyjny inspirowany przyrodą oraz przeszukiwanie wzorcem (ang. *pattern search*, PS) jako bezgradientowy algorytm optymalizacji lokalnej.

4.2. Mechanizm adaptacji regulatora ze sprzężeniem od wektora zmiennych stanu

Równanie (4.1) przedstawia ideę porównania różnych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu w napędzie elektrycznym z silnikiem PMSM. Aby wyniki tego porównania były miarodajne, należy zagwarantować te same warunki początkowe i warunki pracy napędu. W związku z tym wymagany jest powtarzalny sygnał sterujący, którego okres jest nazywany również oknem czasowym ze względu

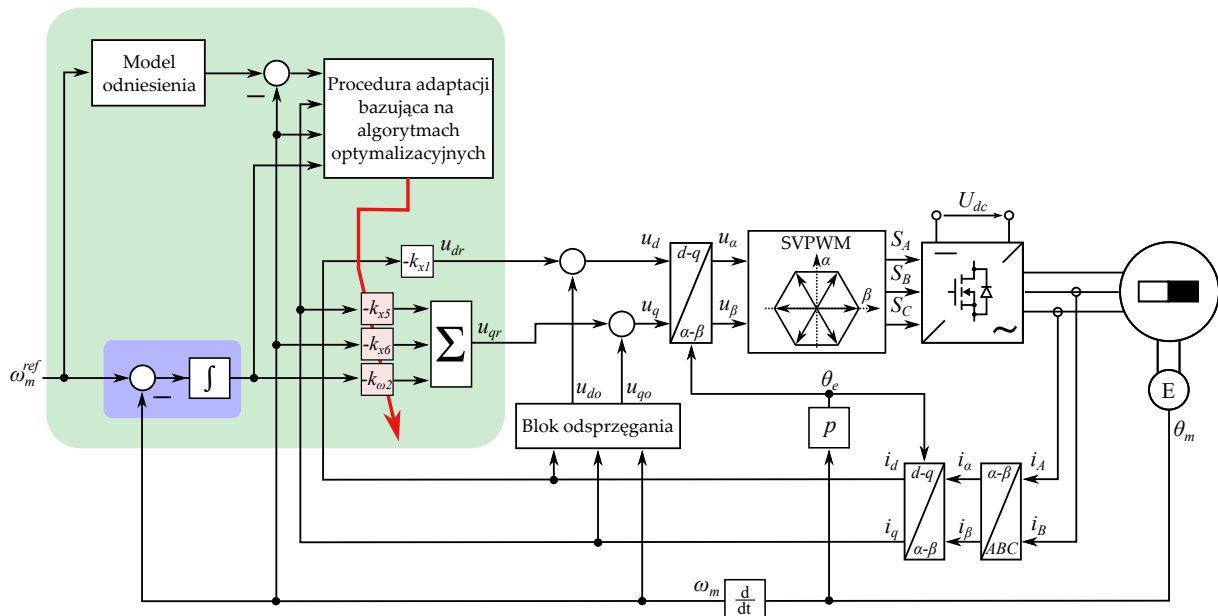
na tzw. okienkowanie odpowiedzi w celu wyznaczenia funkcji celu. Następnie podczas pracy napędu za wykrywanie zmian w układzie i zatrzymywanie adaptacji odpowiedzialna jest opracowana przez Autora niniejszej rozprawy procedura adaptacji dla algorytmów optymalizacyjnych (PAAO). Algorytm optymalizacyjny minimalizuje błąd w pełnym oknie w przeciwieństwie do reguły Widrow'a-Hoff'a i innych klasycznych algorytmów adaptacyjnych, które minimalizują chwilowy błąd i aktualizują współczynniki w każdej pętli regulacji.



Rysunek 4.1. Wizualizacja okienkowanej funkcji celu w porównaniu do aktualizacji w każdym okresie próbkowania regulatora

Próba minimalizacji chwilowego błędu powoduje, że klasyczne algorytmy próbują zniwelować wpływ nieliniowości, szumów pomiarowych, momentów zaczepowych i błędów kwantyzacji, których całkowite zniwelowanie jest nieosiągalne w badanej strukturze regulacji [73]. Skutkuje to koniecznością redukcji współczynnika uczenia tego algorytmu w celu odpowiedniej generalizacji rozwiązania w dłuższym horyzoncie adaptacji przez algorytm. W proponowanej metodzie funkcja celu obliczana jest na bazie określonego przedziału czasu, pokrywającego się z kolejnymi okresami sygnału referencyjnego. Dzięki temu algorytm jest odpowiedzialny za minimalizację różnicy pomiędzy odpowiedzią modelu odniesienia a rzeczywistą odpowiedzią napędu w pełnym oknie. W ten sposób wspomniane nieliniowości przestają bezpośrednio wpływać na jakość adaptacji [46]. Wizualizacja okienkowanej funkcji celu w porównaniu do aktualizacji w każdym okresie próbkowania regulatora została przedstawiona na rysunku 4.1. Schemat blokowy adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu elektrycznego z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego

na procedurze adaptacji wykorzystującej algorytmy optymalizacyjne został przedstawiony na rysunku 4.2.



Rysunek 4.2. Schemat blokowy adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego na procedurze adaptacji wykorzystującej algorytmy optymalizacyjne

4.2.1. Model odniesienia

Zaproponowany model odniesienia dla reguły Widrow'a-Hoff'a z poprzedniego rozdziału może zostać również wykorzystany w niniejszej metodzie. Warto natomiast podkreślić, że zastosowanie okienkowanej funkcji celu pozwala na znaczne uproszczenie w tym zakresie. Wyliczenie wskaźnika jakości regulacji bazując na pełnym oknie sygnału referencyjnego, pozwala na zastosowanie zarówno bardzo dobrze dopasowanego modelu w postaci otrzymanej transmitancji dla reguły Widrow'a-Hoff'a, jak również uproszczonej inercji pierwszego rzędu wyrażonej równaniem:

$$G_{\omega}^{**}(s) = \frac{1}{\tau_{MRAS} s + 1} \quad (4.2)$$

gdzie: τ_{MRAS} jest stałą czasową, którą dobiera się bazując na odpowiedzi dla nominalnych parametrów obiektu. Stała czasowa może zostać przyjęta jako czas, w którym odpowiedź układu osiąga 0.632 wartości w stanie ustalonym. Niezależnie od wybranego modelu, zaproponowany w niniejszej rozprawie mechanizm adaptacji bazujący na algorytmach optymalizacyjnych skutecznie dopasuje do niego odpowiedź układu z punktu widzenia pełnego okresu sygnału referencyjnego. Niemniej jednak, stosując model odniesienia o adekwatnym rzędzie, pozwala

na dokładniejsze doprecyzowanie oczekiwanej odpowiedzi układu. Dlatego model odniesienia wyrażony równaniem (3.7) zostanie wykorzystany w badaniach symulacyjnych w celu umożliwienia porównania z regułą Widrow'a-Hoff'a.

4.2.2. Procedura adaptacji bazująca na algorytmie optymalizacyjnym

W literaturze algorytmy optymalizacyjne są szeroko stosowane w metodach automatycznego doboru współczynników regulatora [36, 44, 58, 59]. Mechanizmy te działają w trybie off-line, czyli optymalizacja odbywa się na modelu symulacyjnym układu, a następnie otrzymane współczynniki są implementowane w napędzie elektrycznym. Procedura takiej aplikacji jest następująca:

- inicjalizacja algorytmu optymalizacyjnego,
- uruchomienie algorytmu na wcześniej zdefiniowaną liczbę iteracji,
- zapisanie i wyświetlenie otrzymanego wyniku.

W przypadku regulatora adaptacyjnego, cały proces adaptacji powinien przebiegać na rzeczywistym obiekcie, czyli w trybie on-line. W związku z tym konieczne jest dostarczenie procedury umożliwiającej:

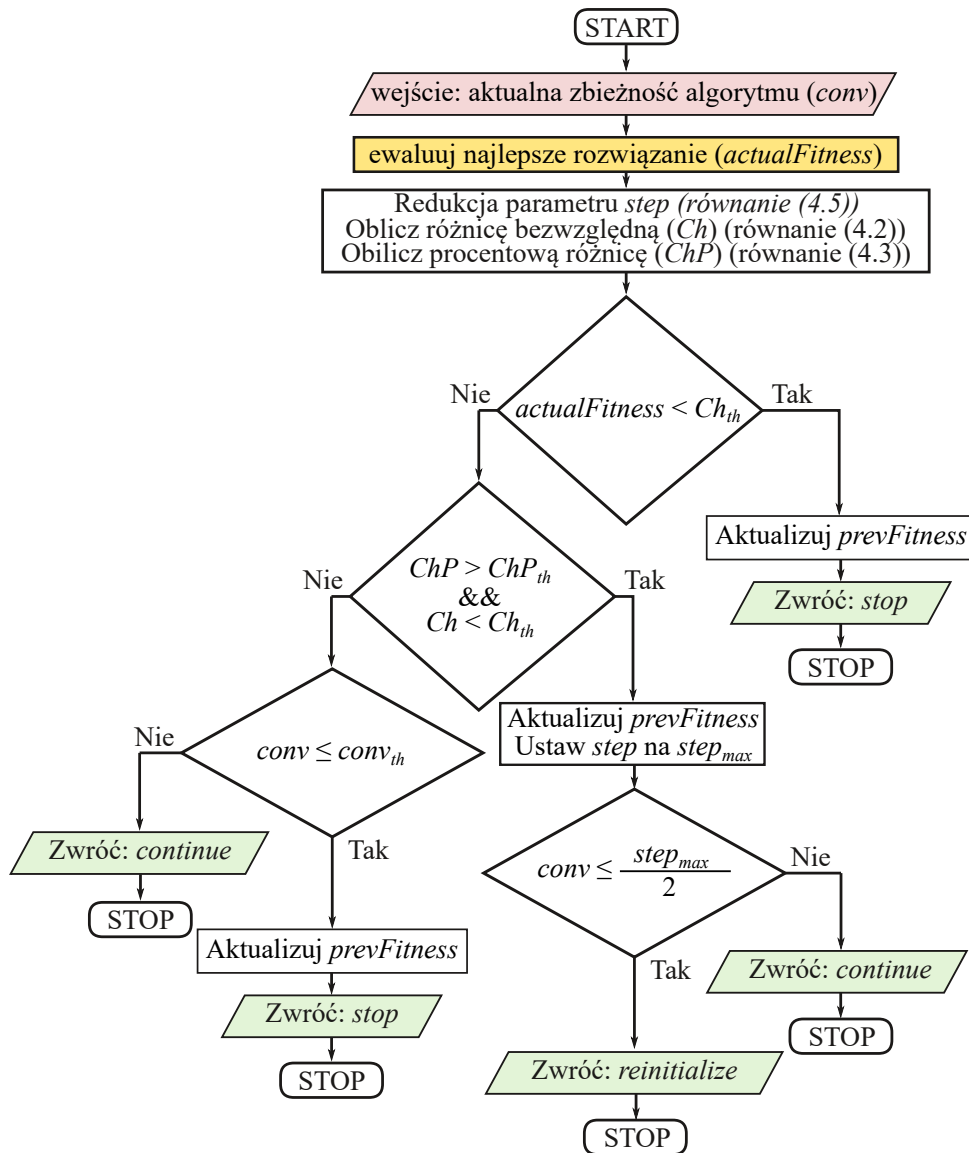
- automatyczne wyzwolenie algorytmu w razie wykrycia zmian w parametrach obiektu,
- optymalizację współczynników regulatora,
- automatyczne zatrzymanie optymalizacji, np. uzyskanie oczekiwanej dokładności odwzorowania modelu odniesienia lub uzyskanie zbieżności algorytmu.

W tym celu została opracowana procedura adaptacji dla algorytmów optymalizacyjnych (PAAO). Proponowana procedura wymaga cyklicznego ewaluowania najlepszego rozwiązania co określoną liczbę ewaluacji funkcji celu ($PAAO_{\text{period}}$). Zwraca ona jedną z trzech dozwolonych stanów w zależności od wartości funkcji celu najlepszego rozwiązania:

- *reinitialize flag* – obszar przeszukiwania algorytmu optymalizacyjnego powinien zostać rozszerzony,
- *continue flag* – optymalizacja powinna być kontynuowana,
- *stop flag* – optymalizacja powinna być zatrzymana.

Schemat blokowy PAAO został przedstawiony na rysunku 4.3.

Głównym zadaniem procedury adaptacji jest okresowa ewaluacja najlepszego rozwiązania, które znalazł algorytm optymalizacyjny. Zmiana funkcji celu najlepszego rozwiązania może być spowodowana zmianą parametrów obiektu lub znalezionym lepszym rozwiązaniem przez



Rysunek 4.3. Schemat blokowy procedury adaptacji dla algorytmów optymalizacyjnych

algorytm optymalizacyjny podczas adaptacji. Ze względu na brak dodatkowej informacji z pomiarów lub obserwatorów nie jest możliwe zdefiniowanie powodu zmiany wartości funkcji celu najlepszego rozwiązania. W takim przypadku wykrycie zmiany powoduje wyzwolenie adaptacji lub zwiększenie obszaru przeszukiwań algorytmu (*reinitialization flag*). W celu ograniczenia rozrzutu testowanych rozwiązań zwiększenie obszaru przeszukiwań jest wyzwalane tylko, jeżeli aktualna zbieżność algorytmu ($conv$) jest mniejsza lub równa połowie maksymalnej dozwolonej przestrzeni przeszukiwań algorytmu $step_{max}$. Ze względu na szumy pomiarowe oraz inne nieliniowości, decyzja, czy zmiana wartości funkcji celu jest wystarczająco znacząca, by wyzwolić ponowną adaptację, bazuje na dwóch aspektach:

— wartości bezwzględnej z różnicy (Ch) pomiędzy aktualną wartością funkcji celu dla najlepszego rozwiązania ($currentFitness$) i poprzednią wartością ($prevFitness$):

$$Ch = |actualFitness - prevFitness| \quad (4.3)$$

— procentowej zmianie (ChP) pomiędzy powyższymi wartościami:

$$ChP = 100\% \cdot \frac{Ch}{prevFitness} \quad (4.4)$$

Pierwszy z aspektów jest konieczny ze względu na zapewnienie odporności PAAO na szumy pomiarowe i błędy spowodowane nieliniowościami układu, np. moment zaczepowy silnika. Dodatkowo wartość Ch określa zakładaną dokładność adaptacji, tj. jeżeli założymy, że wartość funkcji celu dla tego samego rozwiązania uruchamianego raz po raz zmienia się w granicach np. 0.2, to zejście poniżej tej wartości podczas procesu adaptacji oznacza, że znalezione rozwiązanie jest na poziomie szumów i można zatrzymać adaptację. Drugi parametr (ChP) jest wykorzystany do ograniczenia liczby ponownych wyzwoleń adaptacji. Jest to istotne, ponieważ wykrycie zmiany powoduje konieczność ponownych ewaluacji aktualnych rozwiązań, co wydłuża proces adaptacji. Ponowne wyzwolenie adaptacji zwiększa przeszukiwany obszar przez algorytm optymalizacyjny do jego maksymalnego dozwolonego rozmiaru:

$$step = step_{max} \quad (4.5)$$

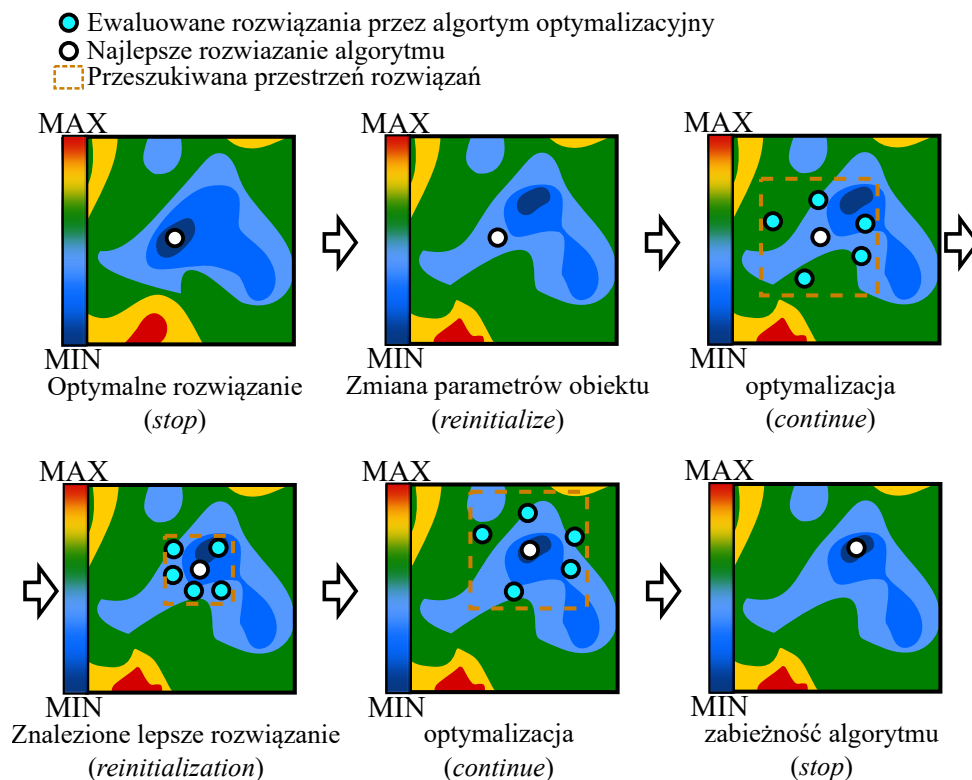
Każde wywołanie PAAO zmniejsza przestrzeń przeszukiwań ($step$), wykorzystując następującą zależność:

$$step = \alpha \cdot step \quad (4.6)$$

gdzie: $\alpha \in (0; 1)$ jest parametrem inspirowanym symulowanym zachowaniem wyżarzania (ang. *simulated annealing behaviour*) [77]. Zastosowany mechanizm pozwala na przyspieszenie zbieżności algorytmu. Warto zaznaczyć, że przestrzeń rozwiązań jest ruchoma i jej środek wyznacza aktualnie najlepsze rozwiązanie. Przestrzeń rozwiązań jest ograniczona jako tzw. *box-constrained*, czyli w każdym wymiarze parametr musi być odległości $\pm step_{max}$ od najlepszego rozwiązania.

W przypadku, gdy warunki wykrycia zmiany w parametrach układu nie zostały spełnione, PAAO sprawdza, czy adaptacja powinna zostać zatrzymana ze względu na brak lepszego

rozwiązania w analizowanej przestrzeni rozwiązań. Warunkiem do zatrzymania procesu adaptacji jest zbieżność algorytmu ($conv$) mniejsza niż predefiniowany próg ($conv_{th}$). Zakładając, że możliwe jest dopasowanie odpowiedzi układu do modelu odniesienia, ten warunek nie będzie wykorzystany, ponieważ, jak wspomniano wcześniej, za kryterium stopu służy również wartość Ch jako oczekiwana dokładność dopasowania. Jeżeli powyższe warunki nie zostały spełnione, żeby wyzwolić ponowną adaptację lub zatrzymać algorytm, PAAO zwraca flagę *continue flag* oznaczającą kontynuację optymalizacji przez algorytm. Wizualizacja procesu adaptacji bazującej na PAAO została przedstawiona na rysunku 4.4.



Rysunek 4.4. Wizualizacja procesu adaptacji bazującej na PAAO

4.2.3. Globalny algorytm optymalizacyjny - Optymalizacja Rojem Cząstek

W 1995 roku James Kennedy zaproponował algorytm optymalizacji rojem cząstek (ang. *Particle Swarm Optimization*, PSO), który jest inspirowany zachowaniem stad ptaków [24]. PSO jest algorytmem optymalizacyjnym stochastycznym, bezgradientowym i bazującym na populacji. Algorytm podczas inicjalizacji generuje losowe pozycje (x) oraz prędkości (v) cząstek w przestrzeni rozwiązań, których jest predefiniowana liczba (N). Każda cząstka pamięta o swojej personalnej najlepszej pozycji (p) oraz ma informację

o najlepszej globalnej pozycji (g), jaka była znaleziona przez cały rój. Algorytm w każdej iteracji aktualizuje prędkości wszystkich cząstek, następnie ich pozycje i ewaluuje wszystkie cząstki. Podstawowym kryterium stopu algorytmu jest predefiniowana liczba iteracji (M). Zmiana prędkości wykorzystuje następującą zależność:

$$v_{i,j}^{new} = \omega \cdot v_{i,j} + \phi_1 \cdot rand_{(0,1)}^U \cdot (p_{i,j} - x_{i,j}) + \phi_2 \cdot rand_{(0,1)}^U \cdot (g_i - x_{i,j}) \quad (4.7)$$

gdzie: i - numer cząstki, j - indeks wymiaru, v_i oraz v_i^{new} są aktualną i nową prędkością i -tej cząstki, ω , ϕ_1 oraz ϕ_2 są parametrami algorytmu PSO. Zmianę pozycji cząstki wyraża poniższe równanie:

$$x_{i,j}^{new} = x_{i,j} + v_{i,j}^{new} \quad (4.8)$$

gdzie: x_i oraz x_i^{new} są aktualną i nową pozycją i -tej cząstki. Pseudokod PSO został przedstawiony w algorytmie 1.

Algorytm 1 Optymalizacja rojem cząstek

```

1: Inicjalizacja prędkości oraz pozycji cząstek
2: for  $i = 1$  to  $N$  do
3:   Ustaw najlepszą pozycję cząstki jako aktualną pozycję  $p_i = x_i$ 
4: end for
5: Zaktualizuj najlepszą pozycję globalną ( $g$ )
6: for  $iteration = 1$  to  $M$  do
7:   for  $i = 1$  to  $N$  do
8:     Aktualizuj prędkość cząstki ( $v_i$ ) wykorzystując równanie (4.7)
9:     Aktualizuj pozycję cząstki ( $x_i$ ) wykorzystując równanie (4.8)
10:    if  $f(x_i) < f(p_i)$  then
11:       $p_i = x_i$ 
12:    end if
13:    if  $f(x_i) < f(g)$  then
14:       $g = x_i$ 
15:    end if
16:  end for
17: end for

```

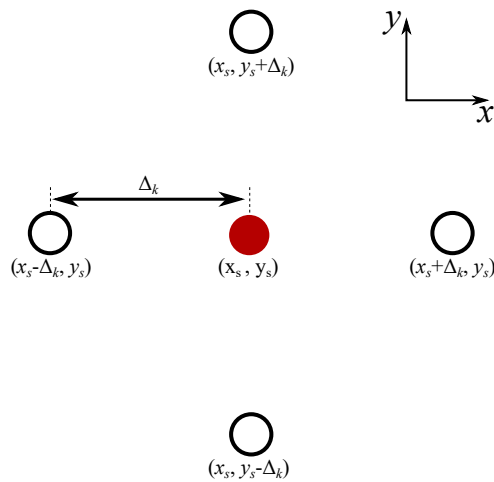
W celu implementacji algorytmu optymalizacji rojem cząstek w mechanizmie adaptacji należy zdefiniować reakcję algorytmu na flagę *reinitialize flag* z PAAO, która oczekuje, że obszar przeszukiwania algorytmu optymalizacyjnego zostanie rozszerzony. W tym celu zwiększana jest prędkość w każdym wymiarze o losową wartość z przedziału $\pm step \cdot g_i$:

$$v_{i,j}^{new} = v_{i,j} + rand_{(-1,1)}^U \cdot (step \cdot g_i) \quad (4.9)$$

Dla przykładu, gdy $step$ jest równe wartości 0.1, oznacza to, że zwiększamy prędkości o maksymalnie 10% aktualnej wartości najlepszego rozwiązania. Warto zaznaczyć, że takie samo ograniczenie dotyczy pozycji cząstek, tj. ograniczona jest ona do wartości $g \pm step \cdot g$. Poza dodaniem losowej wartości do prędkości konieczne jest wymazanie z pamięci cząstek pozycji $p_i \quad \forall i = 1..N$ oraz g określających najlepszą pozycję znaną przez cząstkę oraz najlepszą pozycję globalną oraz przypisanych im wartości funkcji celu $f(p_i) \quad \forall i = 1..N$ oraz $f(g)$. Jest to spowodowane, tym że ponowna inicjalizacja algorytmu może wynikać ze zmian parametrów obiektu, przez co poprzednie wartości funkcji celu będą niepoprawne, co utrudni zbieżność algorytmu. Poza zdefiniowaniem odpowiedniej funkcji związanej ze zwiększeniem obszaru przeszukiwania, należy również określić zachowanie roju w przypadku otrzymania flagi *stop flag* z PAAO. Jest to niezmiernie istotne ze względu na możliwość ponownej zmiany parametrów obiektu, która spowoduje kolejne zwiększenie obszaru przeszukiwania. Zatrzymanie algorytmu polega na przypisaniu każdej cząstce pozycji najlepszego rozwiązania oraz wyzerowaniu jej prędkości. Dzięki temu wystąpienie flagi *reinitialize flag* będzie niezależne od poprzednich pozycji i prędkości cząstek. Ostatnim elementem łączącym zaproponowaną PAAO z algorytmem optymalizacji rojem cząstek jest umożliwienie cyklicznego ewaluowania najlepszego rozwiązania. Jak było wcześniej wspomniane, jednym z parametrów PAAO jest $PAAO_{period}$, który definiuje liczbę ewaluacji funkcji celu przez algorytm optymalizacyjny, po której następuje ewaluacja najlepszego rozwiązania. W przypadku omawianego algorytmu optymalizacyjnego założono, że liczba ewaluacji porównywana jest ze wspomnianym parametrem w każdej iteracji algorytmu, czyli w momencie gdy wszystkie cząstki w roju będą miały zaktualizowaną prędkość oraz pozycję, a następnie zostanie wyznaczona dla każdej z nich funkcja celu. Jest to spowodowane koniecznością zachowania spójności w fazach eksploracji i eksploatacji przestrzeni rozwiązań wszystkich cząstek. Niemniej jednak powoduje to, że w przypadku niepodzielności liczby $PAAO_{period}$ przez liczbę cząstek w roju, najlepsza pozycja będzie ewaluowana z dłuższym okresem niż zakładany, np. dla liczby cząstek równej 7 oraz $PAAO_{period} = 10$ po pierwszej iteracji liczba ewaluacji funkcji celu będzie wynosiła 7, więc PAAO nie zostanie wyzwolona, a po drugiej iteracji będzie wynosiła 14 i PAAO zostanie wyzwolone. Oznacza to, że pomimo oczekiwanego okresu ewaluacji najlepszego rozwiązania równego 10 będzie ono ewaluowane z okresem równym 14.

4.2.4. Lokalny algorytm optymalizacyjny - Przeszukiwanie wzorcem

Algorytm przeszukiwania wzorcem jest bezgradientowym algorytmem optymalizacyjnym, który może być wykorzystany do rozwiązywania nieliniowych problemów [78]. Algorytm posiada tylko dwa parametry, które należy zdefiniować: długość kroku (ang. *step-length*) (Δ_k), gdzie indeks k wskazuje iterację algorytmu oraz minimalna długość kroku (Δ_{min}), która jest kryterium stopu algorytmu. Algorytm zaczyna od predefiniowanej początkowej pozycji, która następnie jest modyfikowana o $\pm\Delta_k$ w każdym wymiarze. Przykładowa iteracja algorytmu dla dwuwymiarowego problemu została przedstawiona na rysunku 4.5.



Rysunek 4.5. Przykładowa iteracja algorytmu przeszukiwania wzorcem dla dwuwymiarowego problemu optymalizacyjnego

Następnie kolejne możliwości rozwiązania są ewaluowane i, jeżeli rozwiązanie jest lepsze niż aktualna pozycja, to algorytm się przesuwa na tę pozycję i ponownie modyfikuje wszystkie wymiary. W przeciwnym wypadku, gdy aktualna pozycja jest najlepsza, długość kroku zmniejsza się o połowę i cały proces jest powtarzany do momentu, gdy długość kroku jest mniejsza niż wartość minimalna. Warto zaznaczyć, że kolejność ewaluowanych rozwiązań jest losowa w każdej iteracji. Pseudokod algorytmu przeszukiwania wzorcem został przedstawiony w algorytmie 2.

W przypadku implementacji algorytmu przeszukiwania wzorcem w PAAO należy zdefiniować elementy związane z flagami *reinitialize flag* i *stop flag* oraz sprawdzanie, czy predefiniowany okres wyzwolenia PAAO został osiągnięty. Algorytm ten jest zdecydowanie prostszy niż algorytm rojem cząstek, co przekłada się na uproszczoną implementację w PAAO. W przypadku ponownego wyzwolenia optymalizacji zwiększony zostaje aktualny krok algorytmu (Δ) do wartości $step_{max}$. Powoduje to, że algorytm przeszukiwania

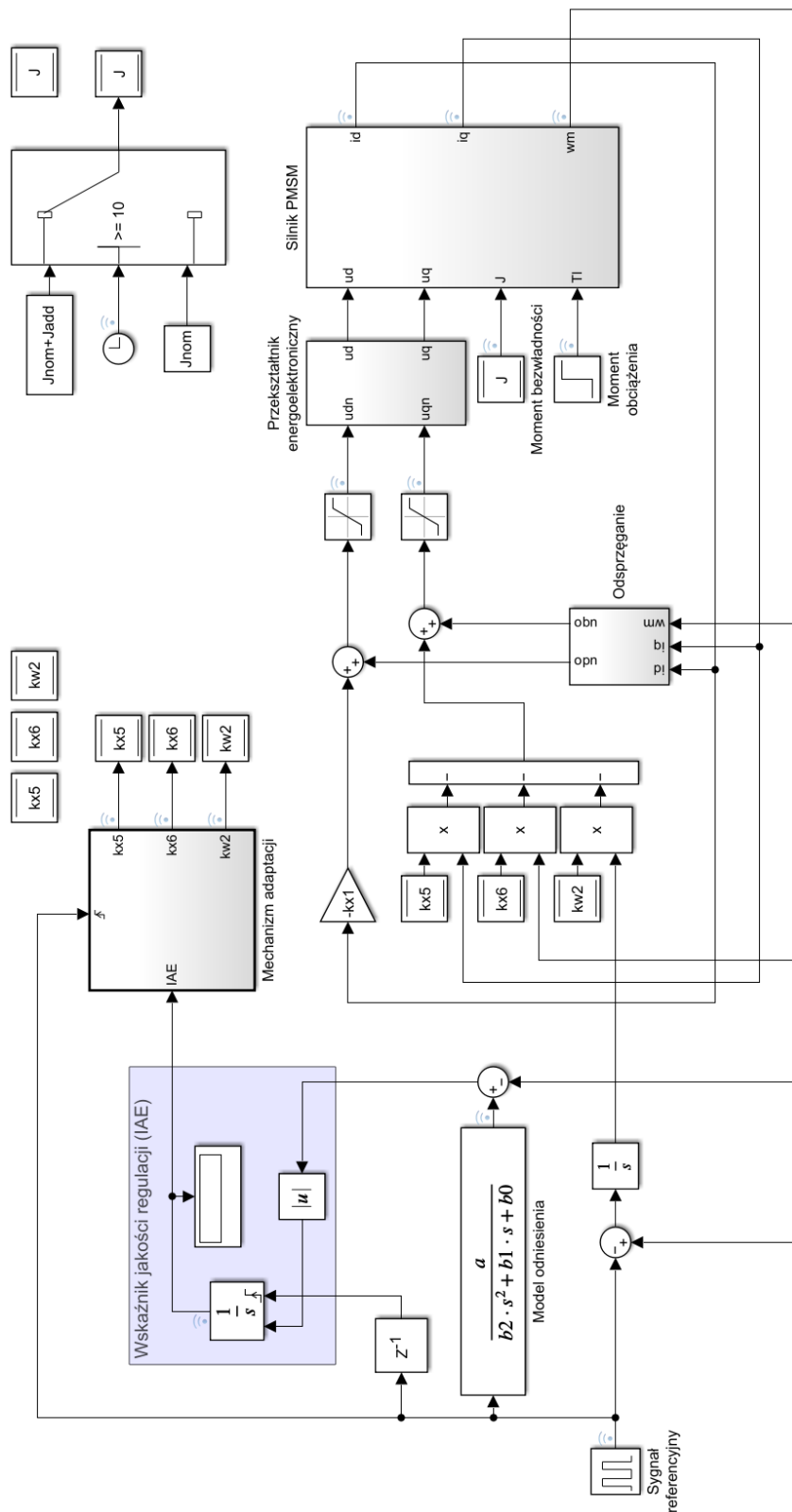
Algorytm 2 Algorytm przeszukiwania wzorcem

```
1:  $k = 0$ 
2: Ustaw  $s_k$  jako początkowe rozwiązanie algorytmu
3: Ustaw  $\Delta_k$  jako początkową długość kroku
4: while  $\Delta_k > \Delta_{min}$  do
5:   Wygeneruj listę rozwiązań w sąsiedztwie aktualnego rozwiązania ( $L_s$ ) dla długości kroku
    $\Delta_k$  (patrz rysunek 4.5)
6:   Wygeneruj losową permutację listy  $L_s$ 
7:   Wyzeruj flagę betterSolutionFound
8:   for all  $s_{new}$  in  $L_s$  do
9:     if  $f(s_{new}) < f(s_k)$  then
10:        $s_{k+1} = s_{new}$ 
11:       break for
12:     end if
13:   end for
14:   if betterSolutionFound = false then
15:     Zmniejsz długość kroku algorytmu ( $\Delta_{k+1} = \frac{\Delta_k}{2}$ )
16:   end if
17:    $k = k + 1$ 
18: end while
```

wzorcem jest wolny od parametrów, które muszą być dobrane - krok algorytmu jest bezpośrednio związany z parametrem PAAO ($step_{max}$). Warto zaznaczyć, że równanie (4.6) wpływa bezpośrednio na redukcję kroku algorytmu przeszukiwania wzorcem, tj. równanie $\Delta = step$ jest zawsze prawdziwe. Zatrzymanie algorytmu oznacza wyzerowanie aktualnego kroku algorytmu, tj. $\Delta = step = 0$. Ze względu na fakt, że algorytm nie jest wieloagentowy i kolejne kroki algorytmu mogą zostać przerwane w dowolnym momencie, porównywanie liczby ewaluacji funkcji celu z predefiniowanym parametrem $PAAO_{period}$ jest wykonywane po każdym kroku algorytmu przeszukiwania wzorcem.

4.3. Wyniki badań symulacyjnych

Model symulacyjny adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na PAAO opracowany w programie MATLAB/SIMULINK został przedstawiony na rysunku 4.6. W przeciwieństwie do reguły Widrow'a-Hoff'a, całkowity wskaźnik jakości regulacji jest wykorzystany nie tylko do przedstawienia procesu adaptacji, ale również opracowana PAAO wykorzystuje tę informację w mechanizmie adaptacji. Sama procedura nie jest obciążeniem dla pracy napędu, ponieważ jest ona wywoływana raz na okres sygnału referencyjnego. Dla porównania, reguła Widrow'a-Hoff'a jest wykonywana w każdej pętli

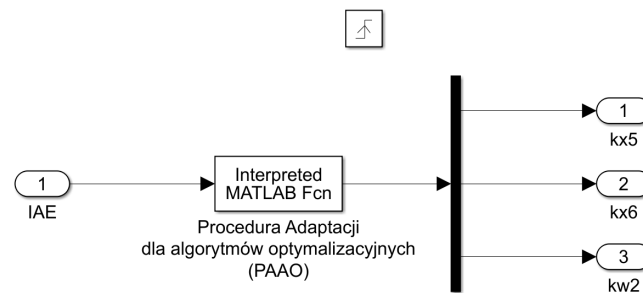


Rysunek 4.6. Model symulacyjny adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na procedurze adaptacji dla algorytmów optymalizacyjnych opracowany w programie MATLAB/SIMULINK

regulacji, czyli 22000 razy na sekundę w tym konkretnym przypadku. Mechanizm adaptacji bazujący na PAAO zwraca kolejne współczynniki, dla których należy wyznaczyć odpowiedź na pełen okres sygnału referencyjnego oraz wyznaczyć całkowity wskaźnik jakości regulacji.

4.3.1. Implementacja mechanizmu adaptacji

Implementacja mechanizmu adaptacji została wykonana w formie interpretowalnej funkcji w języku programowania MATLAB. Sygnałem wejściowym do owej funkcji jest całkowity wskaźnik jakości z poprzedniego okresu sygnału referencyjnego, natomiast na wyjściu przekazywane są nowe współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu. Model symulacyjny mechanizmu adaptacji współczynników k_{x5} , k_{x5} oraz $k_{\omega 2}$ regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na PAAO opracowany w programie MATLAB/SIMULINK został przedstawiony na rysunku 4.7.



Rysunek 4.7. Model symulacyjny mechanizmu adaptacji współczynników k_{x5} , k_{x5} oraz $k_{\omega 2}$ regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na PAAO opracowany w programie MATLAB/SIMULINK

Zarówno algorytm optymalizacji rojem cząstek jak i przeszukiwanie wzorcem zostały zaimplementowane w następującej formie: każde wywołanie algorytmu przypisuje aktualnie otrzymaną wartość funkcji celu do ewaluowanego rozwiązania i zwraca kolejne współczynniki związane z koniecznością ewaluacji odpowiedzi układu. W każdym algorytmie został wprowadzony licznik ewaluacji funkcji celu, który odpowiada za wywoływanie cykliczne PAAO. Warto zaznaczyć, że podczas implementacji algorytmu przeszukiwania wzorcem zastosowano modyfikację polegającą na ewaluowaniu rozwiązań w sąsiedztwie aktualnego rozwiązania w kolejności losowej, a dodatkowo jako pierwsze ewaluowane rozwiązanie z sąsiedztwa jest umieszczana ostatnia zmiana poprawiająca rozwiązanie i jako ostatnie zmiana przeciwna. Pozwala to na poprawę zbieżności algorytmu przeszukiwania wzorcem.

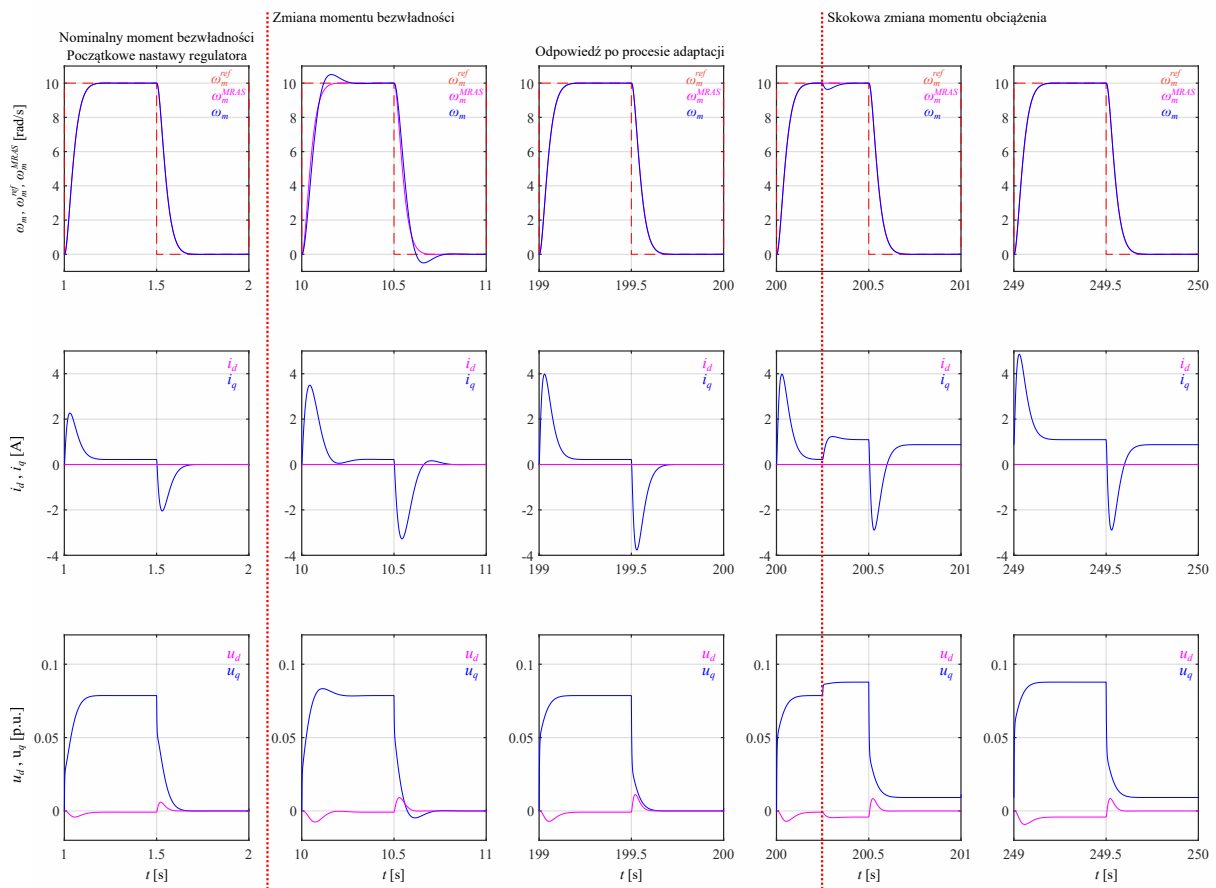
4.3.2. Analiza działania zaproponowanego mechanizmu adaptacji

Jak zostało wspomniane we wcześniejszych rozważaniach, zakłada się, że algorytm zaczyna pracę z nominalnymi współczynnikami regulatora ze sprzężeniem od wektora zmiennych stanu oraz nominalnymi parametrami napędu elektrycznego z silnikiem PMSM. W związku z tym, algorytm uruchamiany jest w fazie ewaluowania najlepszego rozwiązania będącego nominalnymi parametrami regulatora ze sprzężeniem od wektora zmiennych stanu. W momencie zmiany momentu bezwładności niedopasowanie odpowiedzi układu do modelu odniesienia powoduje zwiększenie wartości całkowitego wskaźnika jakości. Wykrycie zmiany przez PAAO rozpoczyna proces adaptacji i trwa aż do osiągnięcia docelowej dokładności dopasowania lub jeżeli to nie jest możliwe, to do momentu zbieżności algorytmu. Parametry PAAO oraz algorytmów optymalizacyjnych wykorzystanych w tym podrozdziale zostały przedstawione w tabeli 4.1.

Tabela 4.1. Parametry PAAO oraz algorytmów optymalizacyjnych

Procedura adaptacji dla algorytmów optymalizacyjnych			
Symbol	Wartość	Symbol	Wartość
$step_{max}$	10%	$conv_{th}$	0.01
$alpha$	0.8	Ch_{th}	0.02
$PAAO_{period}$	30	ChP_{th}	10%
Algorytm optymalizacji rojem cząstek			
ϕ_1	0.5	ω	0.72984
ϕ_2	4.0	N	3
Algorytm optymalizacji przeszukiwania wzorcem			
Δ	$step$		

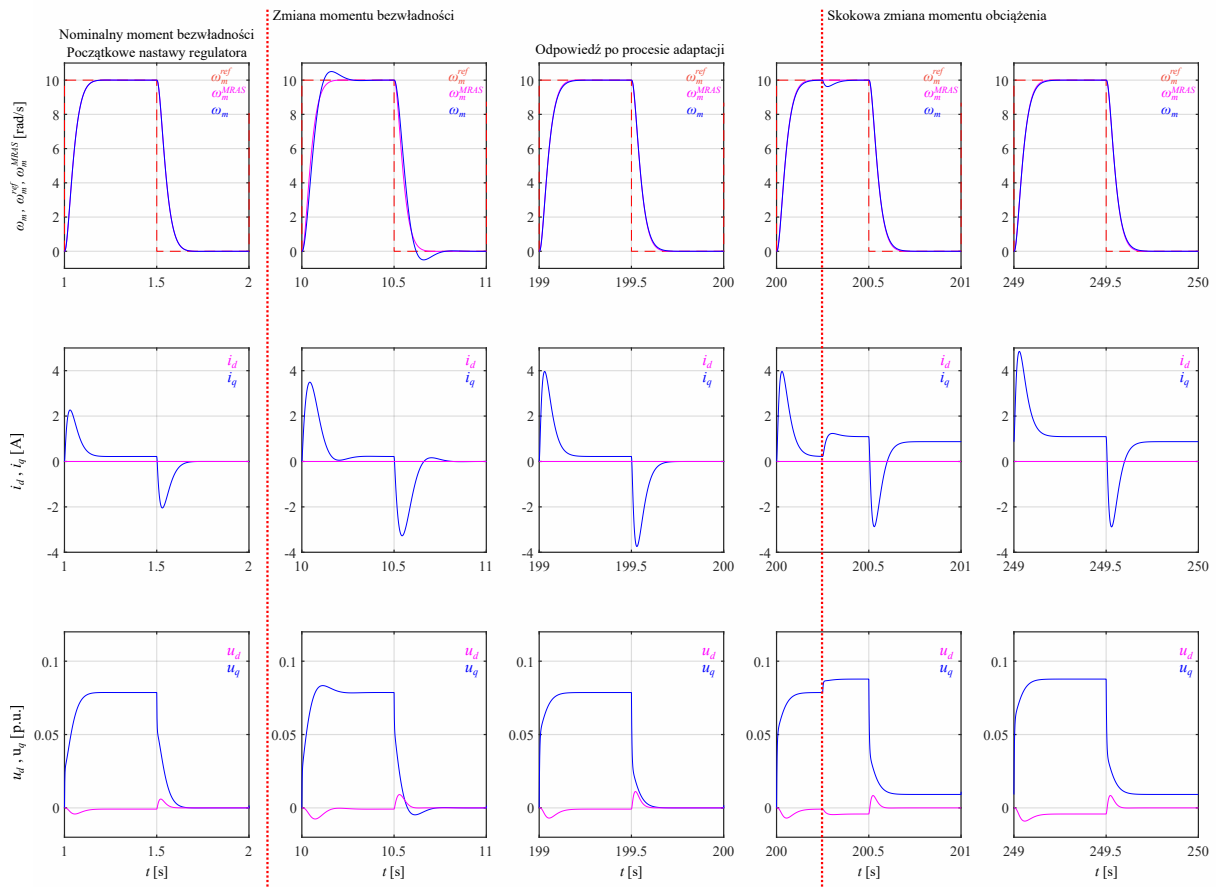
Eksperyment dla mechanizmu adaptacji bazującego na algorytmie optymalizacyjnym został przeprowadzony analogicznie do mechanizmu adaptacji bazującego na regule Widrow'a-Hoff'a. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego zostały przedstawione na rysunku 4.8 dla algorytmu optymalizacyjnego rojem cząstek oraz na rysunku 4.9 dla algorytmu przeszukiwania wzorcem. Przedstawione wykresy dla obu algorytmów są niemal identyczne, ponieważ przedstawiają okresy, w którym nastąpiła zmiana parametrów obiektu lub przyłożony został moment obciążenia oraz przebiegi dla stanu ustalonego procesu adaptacji. Warto zauważyć, że wynikiem adaptacji w obu przedstawionych algorytmach optymalizacyjnych jest dopasowanie odpowiedzi układu do modelu odniesienia. Różnice, jakie występują pomiędzy tymi odpowiedziami, są niezauważalne bez znaczącego przybliżenia wykresu. Oznacza to,



Rysunek 4.8. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego. Zaproponowany mechanizm adaptacji bazujący na algorytmie optymalizacyjnym rojem cząstek

że zaproponowany mechanizm adaptacji bazujący na algorytmach optymalizacyjnych działa prawidłowo i skutecznie zminimalizował różnicę pomiędzy odpowiedzią układu a modelem odniesienia.

Przebieg czasowy całkowego wskaźnika jakości oraz współczynników regulatora ze sprzężeniem od wektora zmiennych stanu został przedstawiony na rysunku 4.10 dla obu algorytmów. Zmiana parametrów w dziesiątej sekundzie spowodowała znaczącą zmianę całkowego wskaźnika jakości regulacji i rozpoczęcie procesu adaptacji. Oba algorytmy zaczęły przeszukiwać przestrzeń rozwiązań, aż do momentu osiągnięcia zakładanego dopasowania odpowiedzi układu do modelu odniesienia. Warto zwrócić uwagę na charakter zmian współczynników regulatora. W przypadku algorytmu optymalizacji rojem cząstek wszystkie parametry są zmieniane równocześnie, ale o różne wartości, natomiast w przypadku algorytmu przeszukiwania wzorcem zmieniany jest jeden parametr względem aktualnego najlepszego

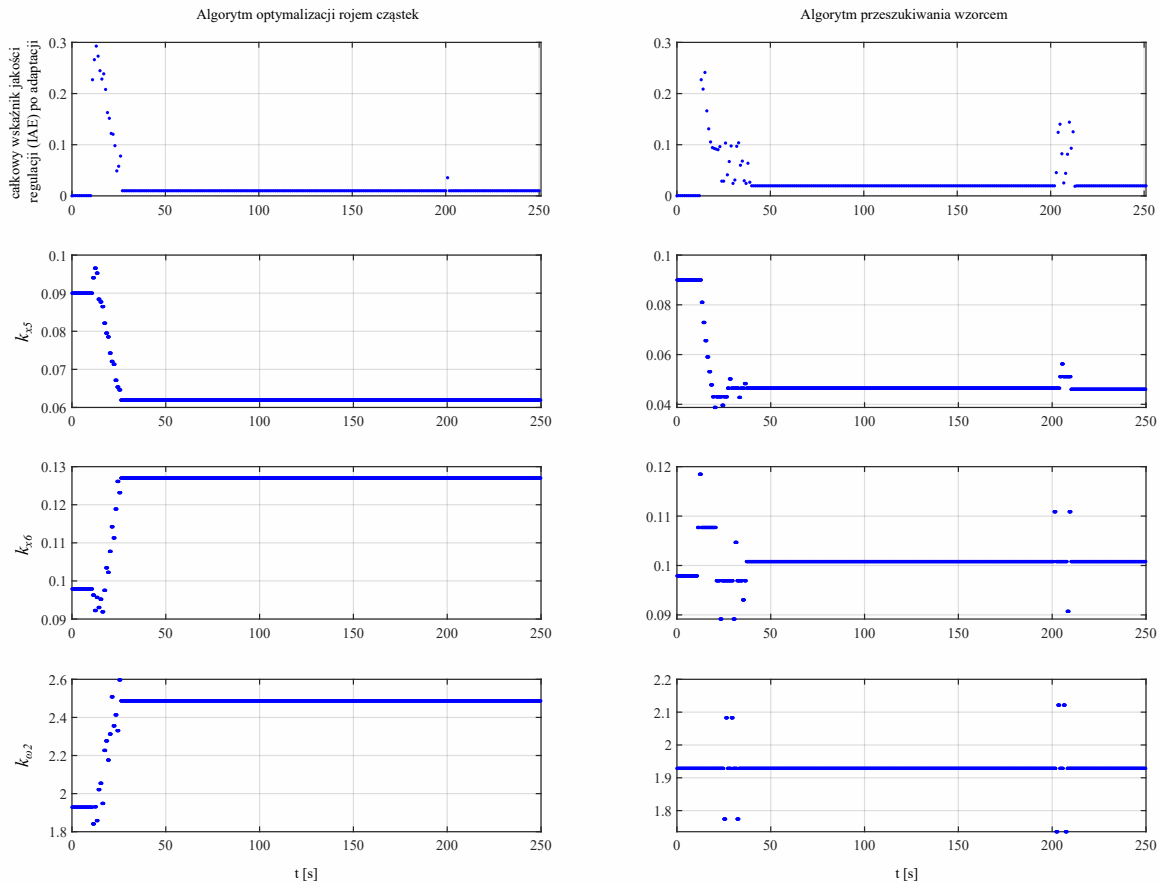


Rysunek 4.9. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego. Zaproponowany mechanizm adaptacji bazujący na algorytmie optymalizacyjnym przeszukiwania wzorcem

rozwiązania w danej ewaluacji i w sposób deterministyczny o wartość równą krokowi algorytmu. Oba algorytmy skutecznie zminimalizowały różnicę względem modelu odniesienia, uzyskując różne współczynniki regulatora:

- k_{x5} : 0.0619 (PSO); 0.0465 (PS),
- k_{x6} : 0.1270 (PSO); 0.1008 (PS),
- $k_{\omega 2}$: 2.4869 (PSO); 1.9286 (PS),

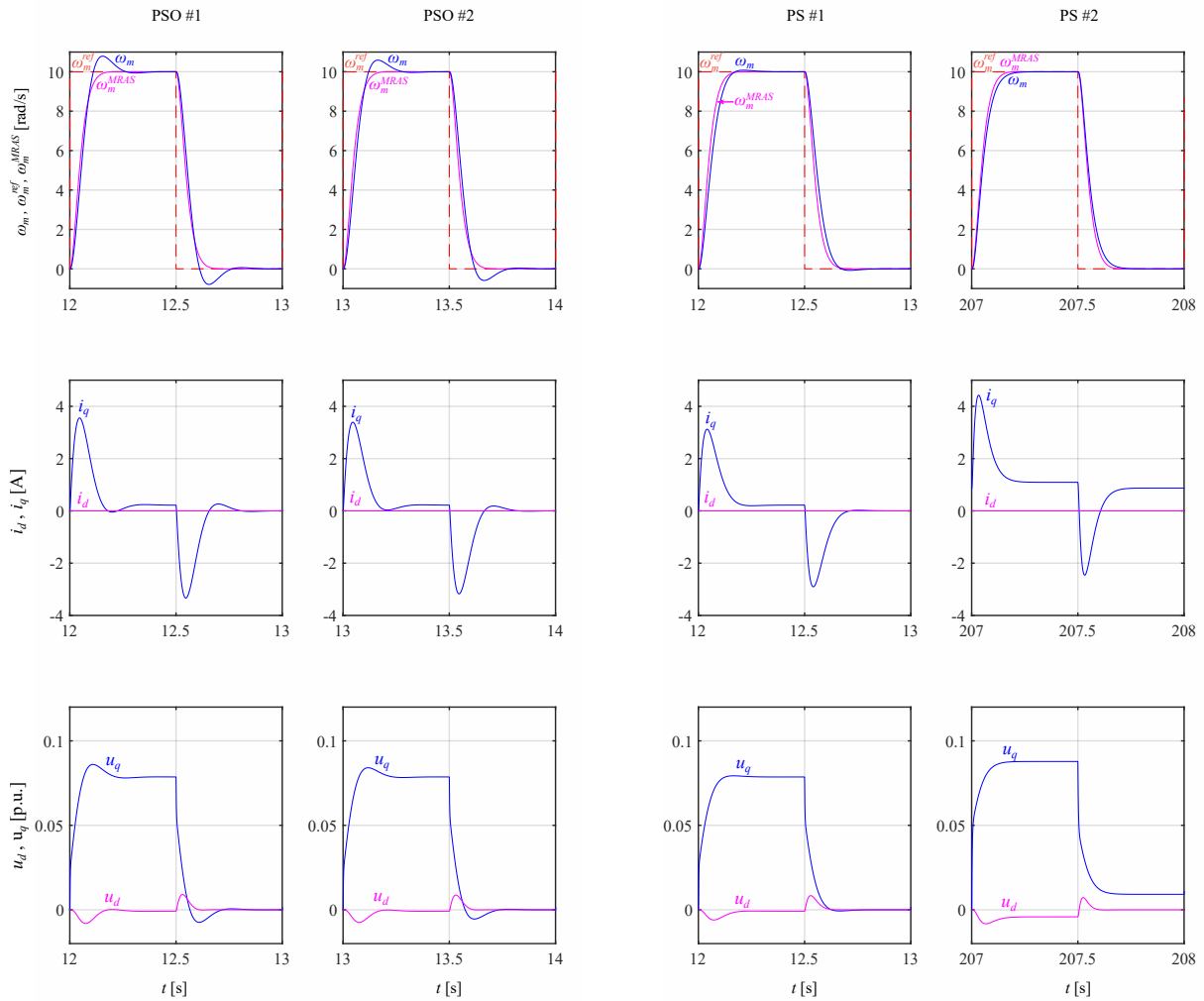
Zmiana momentu obciążenia nie spowodowała wyzwolenia algorytmu adaptacji z algorytmem optymalizacji rojem cząstek, ponieważ zmiana całkowitego wskaźnika jakości regulacji nie przekroczyła wartości progowej powodującej wyzwolenie ponownej optymalizacji. W przypadku algorytmu optymalizacji przeszukiwania wzorcem adaptacja została uruchomiona i algorytm próbował poprawić aktualne rozwiązanie. Ponowna ewaluacja najlepszego rozwiązania spowodowała, że wartość całkowitego wskaźnika jakości regulacji była mniejsza



Rysunek 4.10. Przebiegi całkowego wskaźnika jakości oraz współczynników regulatora ze sprzężeniem od wektora zmiennych stanu dla mechanizmu adaptacji bazującego na algorytmach optymalizacyjnych

niż zakładana dokładność (tj. mniejsza wartość wskaźnika niż minimalna wartość Ch), co spowodowało zatrzymanie adaptacji na aktualnym rozwiązaniu.

Dodatkowo warto zauważyć, że zastosowanie algorytmów optymalizacyjnych ma charakter częściowo losowy. W obu algorytmach element losowości determinuje kolejne ewaluowane współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu w celu eksploracji przestrzeni rozwiązań. Testowane rozwiązania mają czasami wyższą wartość całkowego kryterium jakości regulacji niż nominalne parametry, co oznacza gorsze dopasowanie w trakcie adaptacji niż regulator stacjonarny. W związku z tym na rysunku 4.11 przedstawiono dodatkowe przebiegi prezentujące dwa najgorsze dopasowanie do modelu odniesienia dla obu algorytmów optymalizacyjnych. Pomimo większej wartości całkowego wskaźnika jakości regulacji, wszystkie odpowiedzi są stabilne i osiągają stan ustalony przed wystąpieniem kolejnej zmiany sygnału referencyjnego. Większa wartość wskaźnika w przypadku dwóch



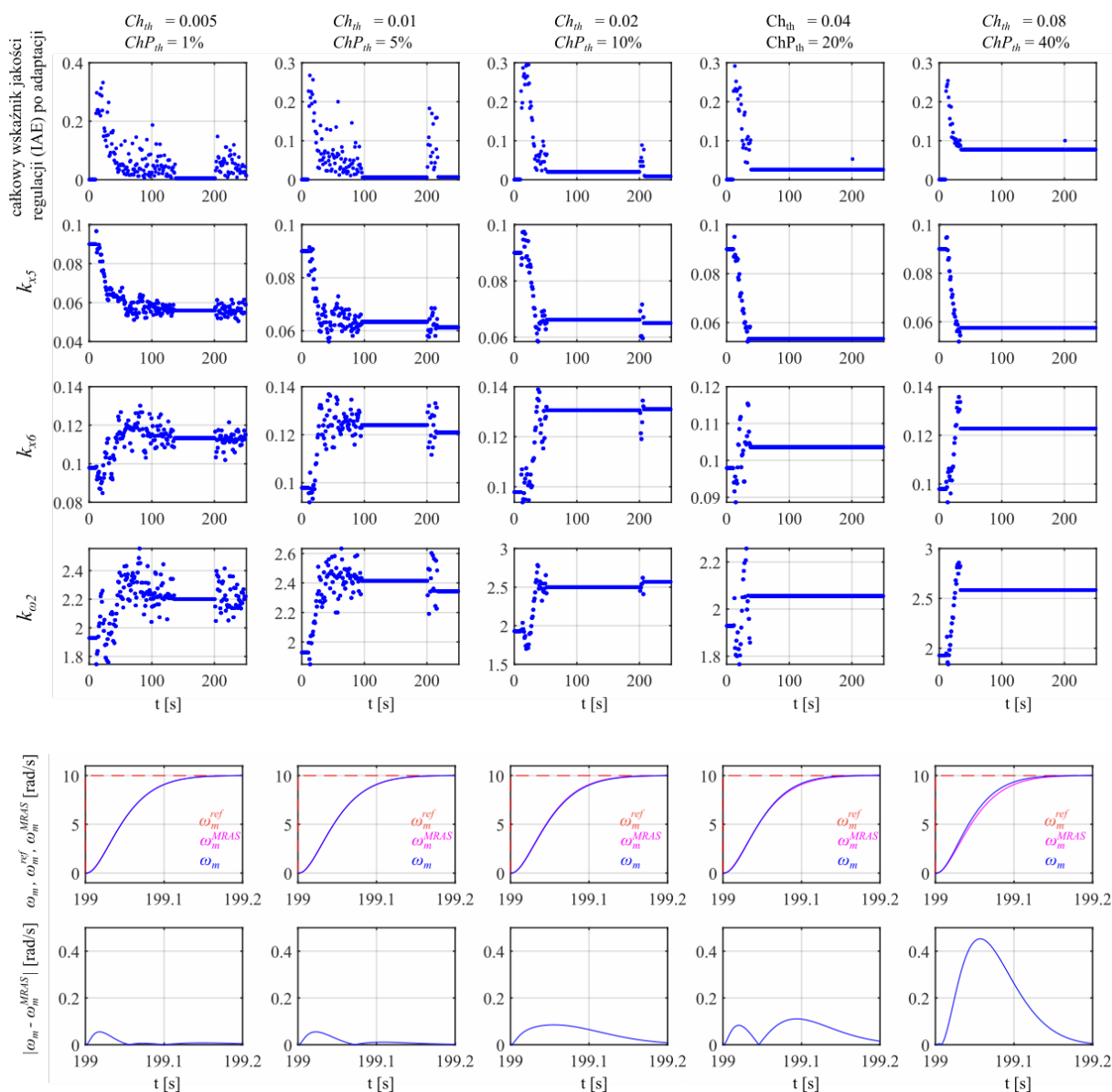
Rysunek 4.11. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla okresów sygnału referencyjnego z najgorszymi całkowitymi wskaźnikami jakości regulacji w trakcie trwania procesu adaptacji.

pierwszych przebiegów wynika ze zwiększenia przeregulowania układu, a w przypadku dwóch ostatnich ze zmniejszenia dynamiki.

4.3.3. Analiza wpływu parametrów procedury adaptacji na przebieg procesu adaptacji

Przeprowadzona analiza wpływu parametrów procedury adaptacji na jakość adaptacji została przeprowadzona tylko dla algorytmu optymalizacji rojem cząstek, ponieważ algorytm optymalizacyjny jest odrębnym aspektem proponowanego mechanizmu adaptacji. Parametry związane z wykrywaniem zmiany parametrów układu, tj. Ch_{th} oraz ChP_{th} , należy dobrać do zakresu zmian parametrów występujących w układzie oraz akceptowalnej różnicy pomiędzy

modelem odniesienia a rzeczywistą odpowiedzią układu. Na rysunku 4.12 przedstawiono wykres całkowego wskaźnika jakości regulacji i współczynników regulatora ze sprzężeniem od wektora zmiennych stanu w czasie dla różnych wartości Ch_{th} oraz ChP_{th} , a także zbrocza narastające odpowiedzi po adaptacji i wartość bezwzględną z różnicy pomiędzy modelem odniesienia a aktualną wartością prędkości kątowej silnika. Zwiększanie wartości Ch_{th}

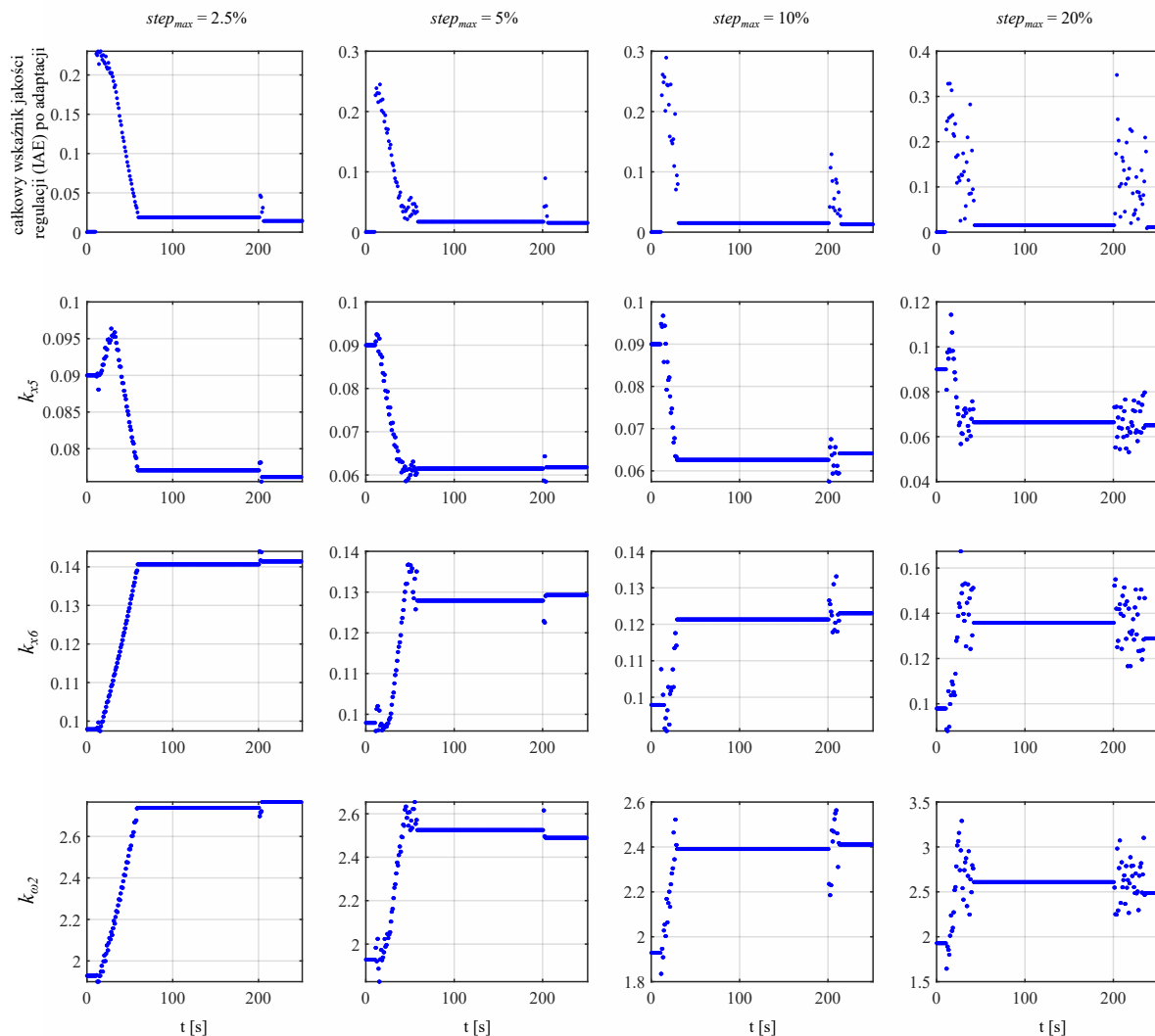


Rysunek 4.12. Wpływ parametrów związanych z wykrywaniem zmian obiektu oraz tolerowanej różnicy pomiędzy modelem odniesienia a rzeczywistą odpowiedzią układu

oraz ChP_{th} powoduje szybszą zbieżność algorytmu, ponieważ zmniejsza czułość układu na zmiany w odpowiedzi prędkości. Dla bardzo małych wartości każde poprawienie odpowiedzi skokowej powoduje ponowną inicjalizację algorytmu, co wydłuża czas potrzebny do osiągnięcia zbieżności algorytmu. Z drugiej strony, końcowe dopasowanie uzyskane przez

najmniejszą wartość tych współczynników jest najlepsze i szczytowa różnica odpowiedzi nie przekracza 0.055 rad/s. Im większe wartości, tym algorytm rzadziej uruchamia ponowną inicjalizację algorytmu, ale prowadzi do gorszego dopasowania do modelu odniesienia podczas adaptacji, gdzie dla największych badanych parametrów różnica przekraczała 0.45 rad/s.

Kolejnym bardzo istotnym parametrem zaproponowanej struktury sterowania adaptacyjnego jest $step_{max}$, który odpowiada za przestrzeń przeszukiwań wokół aktualnego najlepszego rozwiązania. Porównanie wykresów całkowitego wskaźnika jakości oraz współczynników regulatora ze sprzężeniem od wektora zmiennych stanu w czasie dla różnych wartości wspomnianego parametru zostało przedstawione na rysunku 4.13. Niższa wartość parametru



Rysunek 4.13. Wpływ parametru odpowiedzialnego za rozmiar przeszukiwanej przestrzeni rozwiązań w okół aktualnego najlepszego rozwiązania.

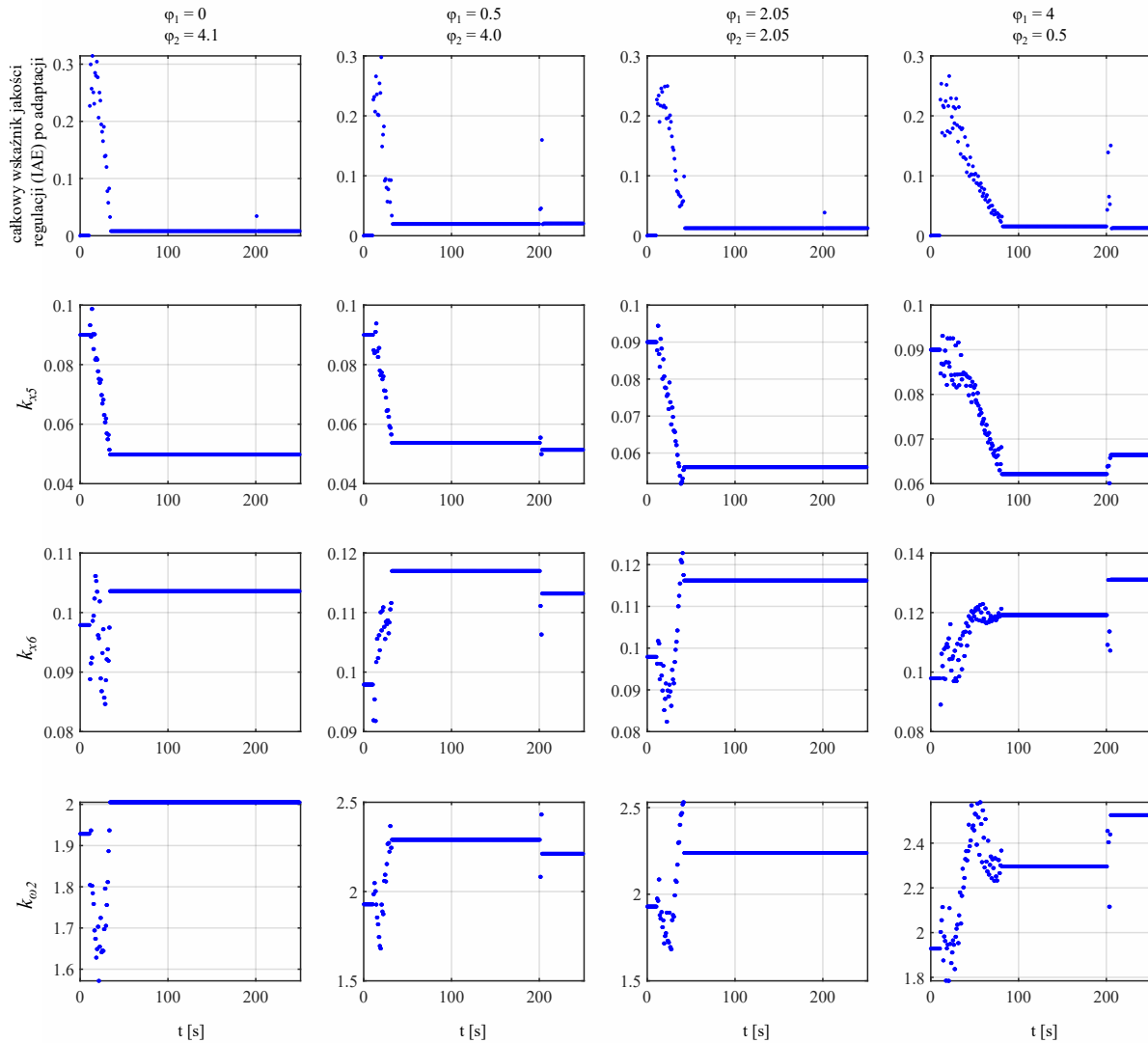
$step_{max}$ powoduje wydłużenie się czasu adaptacji, lecz również pozwala ograniczyć chwilowe zwiększenia się błędu podczas procesu adaptacji. Jest to związane z zawężeniem przeszukiwań, więc algorytm ma możliwość testowania bardziej zbliżonych rozwiązań do aktualnego. Dobór parametru powinien brać pod uwagę możliwości zastosowania silnika. W sytuacji, gdy w procesie przemysłowym niedopuszczalne jest pojawienie się przeregulowania odpowiedzi, parametr $step_{max}$ powinien zostać przyjęty jako stosunkowo mała wartość w celu wyeliminowania szansy na testowanie rozwiązania o jeszcze większym przeregulowaniu niż to występujące ze względu na zmianę parametrów obiektu.

Pozostałe parametry takie jak $alpha$, $PAAO_{period}$ oraz $conv_{th}$ nie mają tak bezpośredniego wpływu na proces adaptacji. Współczynnik $alpha$ pozwala na zapewnienie i przyspieszenie zbieżności w przypadku algorytmów optymalizacyjnych rojowych, np. optymalizacja rojem cząstek, dzięki czemu wszyscy agenci algorytmu zbiegają się w jednym punkcie. W tego rodzaju algorytmach bardzo często niektórzy agenci pozostają w minimum lokalnym, nie dołączając do pozostałych w minimum globalnym. Zastosowanie takiego parametru $alpha$ wymusza przesuwanie agentów ku najlepszemu rozwiązaniu. Jak do tej pory wykazano, że parametr Ch_{th} pozwala na zdefiniowanie oczekiwanej jakości rozwiązania. Natomiast w przypadku, gdyby zadana dokładność była niemożliwa do uzyskania ze względu na nieliniowość lub szumy w układzie, zwiększające wartość całkowitego wskaźnika jakości regulacji, parametr $conv_{th}$ pozwala wywołać zatrzymanie adaptacji, jeżeli zmiany dokonywane przez algorytm optymalizacyjny są stosunkowo relatywnie niewielkie. W przypadku algorytmu optymalizacji rojem cząstek zbieżność algorytmu określa się jako średnią z odchylenia standardowego od średniej z każdego wymiaru pozycji wszystkich cząstek. Dla algorytmu optymalizacji przeszukiwania wzorcem sprawa jest trywialna, ponieważ zbieżność algorytmu jest aktualnym jego krokiem. Ostatnim parametrem jest $PAAO_{period}$ oznaczający, jak często podczas adaptacji jest ewaluowane najlepsze rozwiązanie. Jest on wyrażony w dziedzinie okresów sygnału referencyjnego. Dla przyjętej wartości 30 oznacza to, że co trzydzieści ewaluacji algorytmu optymalizacyjnego jest wykonana ewaluacja najlepszego rozwiązania. Zastosowanie małej wartości, np. 1 byłoby nieefektywne, ponieważ co druga ewaluacja rozwiązania byłaby testowaniem, czy najlepsze rozwiązanie się zmieniło. Jeżeli algorytm potrzebuje średnio 50 ewaluacji funkcji celu do znalezienia rozwiązania, to przy tak małej wartości $PAAO_{period}$ ta średnia wydłużyłaby się do 100 ewaluacji, ponieważ co druga ewaluacja nie dotyczy algorytmu optymalizacyjnego, a mechanizmu adaptacji i wykrywania

zmian w obiekcie. Oczywiście zastosowanie dużej wartości mogłoby spowodować, że jeżeli w trakcie adaptacji wystąpiłaby ponowna zmiana parametrów obiektu, to nie zostałaby ona wykryta i algorytm optymalizacyjny polegałby na funkcji celu dla rozwiązania, która uległa zmianie, co mogłoby uniemożliwić jej zminimalizowanie aż do chwili, gdy ponownie będzie wyzwolona PAAO, która zaktualizuje tę wartość.

4.3.4. Analiza wpływu parametrów algorytmu optymalizacyjnego na jakość adaptacji

Jak zostało wspomniane wcześniej oraz dodatkowo zaznaczone w tabeli 4.1, algorytm optymalizacji przeszukiwania wzorcem nie posiada parametrów, które należy zdefiniować, ponieważ zarówno kryterium stopu, jak i początkowa wartość kroku algorytmu jest definiowana przez zaproponowaną procedurę adaptacji. W związku z tym tylko algorytm inspirowany przyrodą, jakim jest optymalizacja rojem cząstek, jest brany pod uwagę w niniejszym podrozdziale. W analizowanym problemie skutecznie został zastosowany mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a, z czego można wywnioskować, że problem nie jest skomplikowanym problemem optymalizacyjnym. W związku z tym parametry odpowiedzialne za równowagę fazy eksploracji całej przestrzeni rozwiązań i fazy eksploatacji, gdzie poszukiwany jest obszar najlepszego rozwiązania, w pewnym zawężonym wcześniej kręgu, mogą zostać bardziej ukierunkowane na fazę eksploatacji w celu przyspieszenia jego zbieżności. Parametr ψ_1 jest odpowiedzialny za przeszukiwanie przestrzeni rozwiązań związanej z lokalną pozycją cząstki, podczas gdy ψ_2 jest wprost proporcjonalny do przyciągania wszystkich cząstek ku najlepszemu rozwiązaniu, ich suma powinna być większa niż 4 [79]. Na rysunku 4.14 przedstawiono przebiegi całkowego wskaźnika jakości regulacji oraz współczynników regulatora ze sprzężeniem od wektora zmiennych stanu w czasie dla różnych parametrów ψ_1 oraz ψ_2 . Porównując modyfikację parametrów w stosunku do powszechnie używanych wartości równych 2.05 dla obu parametrów, widać, że zwiększenie ψ_1 odpowiedzialnego za eksplorację całej przestrzeni rozwiązań powoduje wydłużenie czasu potrzebnego na zbieżność algorytmu i eksploatację wąskiego kręgu wokół najlepszego rozwiązania. Zmniejszając wartość ψ_1 na rzecz ψ_2 widać skrócenie tego czasu, co jest zgodne z założeniami. Wartość bezwładności prędkości cząstki (ω) powinna być mniejsza niż 1 i jej standardowa wartość wskazana w tabeli 4.1 została wykorzystana w niniejszej rozprawie. Kolejnym parametrem jest populacja roju cząstek. Warto zaznaczyć, że w ogólności liczba cząstek nie wymaga większej liczby iteracji algorytmu, aby uzyskać

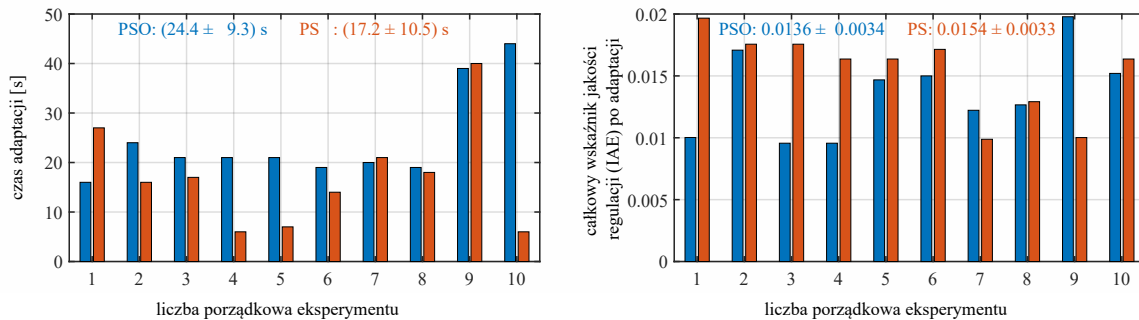


Rysunek 4.14. Wpływ parametrów ψ_1 oraz ψ_2 na proces adaptacji.

zbieżność, lecz w przypadku, gdy rozważamy dziedzinę ewaluacji funkcji celu, to podwójna liczba cząstek wymaga podwójnej liczby ewaluacji funkcji celu w ramach jednej iteracji. Co powoduje, że w przypadku zastosowania do mechanizmu adaptacji większej populacji roju spowoduje wydłużenie bezwzględnego czasu potrzebnego na zakończenie procesu adaptacji. Ze względu na losowość występującą w algorytmie, zbyt mała liczba cząstek może spowodować, że algorytm będzie potrzebował więcej iteracji na eksploatację przestrzeni rozwiązań. W literaturze nie znaleziono informacji o minimalnej liczbie cząstek, aby algorytm skutecznie rozwiązywał problem optymalizacyjny. Natomiast w przypadku omawianego algorytmu rojowego, liczba cząstek powinna być większa niż 1.

4.3.5. Analiza powtarzalności rozwiązania

Ze względu na częściową losowość algorytmów optymalizacyjnych zebrano po 10 symulacji w celu zaprezentowania powtarzalności rozwiązań z wykorzystaniem obu badanych algorytmów. Przyjęto ich parametry zgodne z tabelą 4.1. Na rysunku 4.15 przedstawiono wykres słupkowy czasu zbieżności algorytmu oraz wartości końcowych całkowitego wskaźnika jakości w dziedzinie uruchomień dla algorytmu optymalizacji. Średni czas zbieżności



Rysunek 4.15. Porównanie czasu adaptacji oraz końcowego dopasowania w 10 uruchomieniach algorytmu optymalizacji rojem cząstek i algorytmu przeszukiwania wzorcem

algorytmu przeszukiwania wzorcem jest zdecydowanie krótszy niż optymalizacji rojem cząstek. Ponadto czas adaptacji dla 2 z 10 eksperymentów trwał zaledwie 6 sekund, a dla innego 7 sekund. Algorytm optymalizacji rojem cząstek wymaga dłuższego czasu, lecz odchylenie standardowe od średniej jest niższe w porównaniu do przeszukiwania wzorcem. Wszystkie funkcje celu są poniżej zakładanej dokładności (Ch). Ze względu na większą losowość generowanych rozwiązań przez optymalizację rojem cząstek, częściej znajduje on rozwiązanie lepsze niż przeszukiwanie wzorcem.

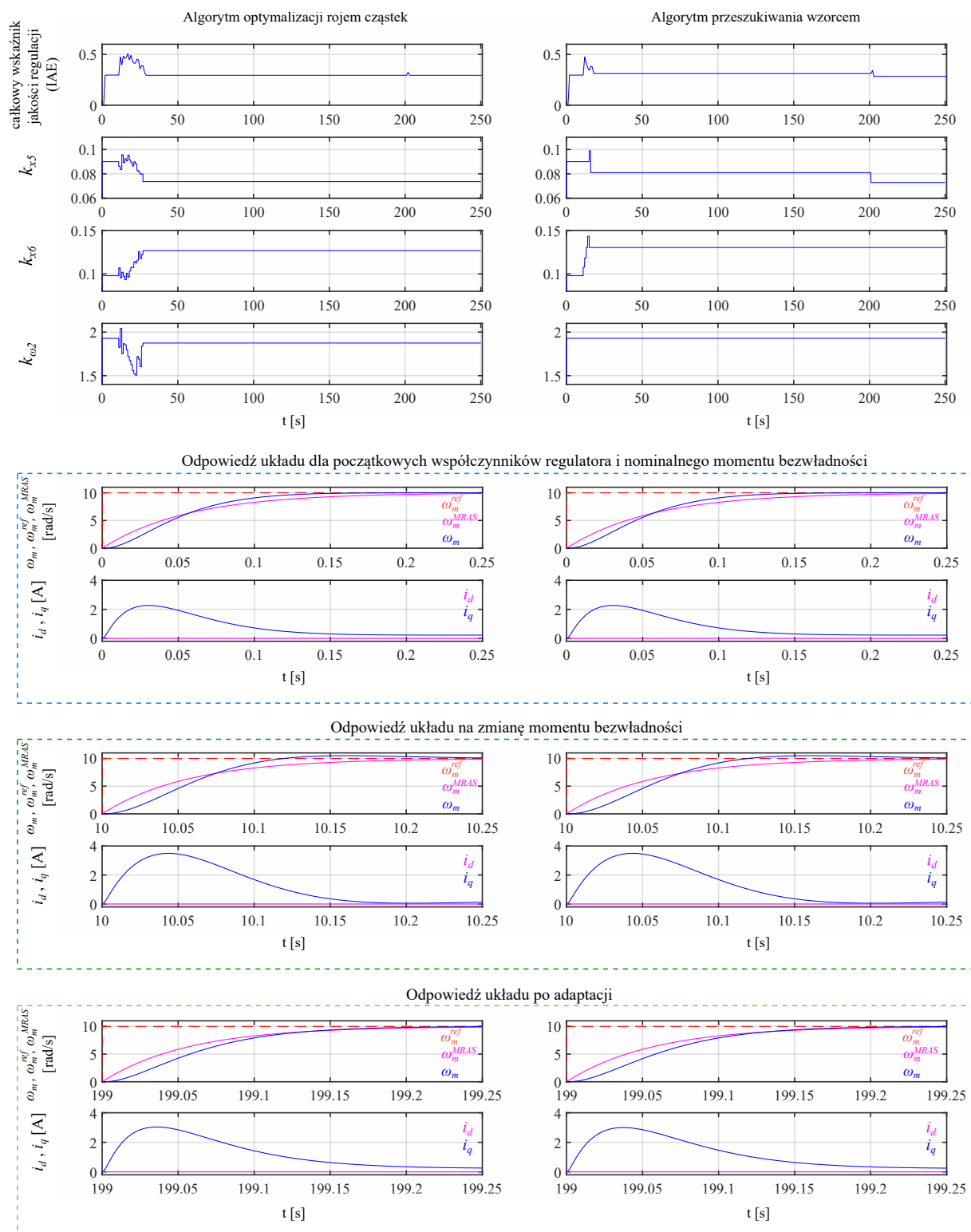
4.3.6. Implementacja modelu odniesienia o nieadekwatnym rzędzie

Analogiczne do badań nad regułą Widrow'a-Hoff'a w zakresie implementacji modelu odniesienia o nieadekwatnym rzędzie wykonano symulację mechanizmu adaptacji bazującego na algorytmach optymalizacyjnych. Warto zaznaczyć, że jednym z głównych własności PAAO jest określenie oczekiwanej dokładności odwzorowania modelu odniesienia poprzez zdefiniowanie parametru Ch_{th} . W przypadku odpowiednio dobranego modelu i wyidealizowanego środowiska, tj. brak szumów pomiarowych, pominięcie nieliniowości sił tarcia oraz momentu zaczepowego, dopasowania odpowiedzi układu do odpowiedzi modelu odniesienia jest niemal idealne. W takim wypadku oczekiwano wartości całkowitego wskaźnika jakości na poziomie

0.02 jako wyniku adaptacji. W przypadku zastosowania modelu odniesienia o nieadekwatnym rzędzie te odpowiedzi są zauważalnie różne, przez co całkowity wskaźnik jakości przyjmuje wartość na poziomie $f_{initial} = 0.296$ i osiągnięcie dokładności odwzorowania na poziomie 0.02 jest niemożliwe. W celu rozwiązania tego problemu, wartość całkowego wskaźnika jakości analizowana w PAAO jest zmniejszona o wartość otrzymaną dla początkowych współczynników regulatora oraz nominalnego momentu bezwładności. Zabieg ten pozwala określić, że użytkownik oczekuje dokładności na poziomie całkowego wskaźnika jakości wynoszącego 0.02, ale względem początkowego dopasowania. Zależność na funkcję celu analizowaną przez PAAO (f_{PAAO}) w stosunku do całkowego wskaźnika jakości (f) wyrażonego równaniem (4.1) jest następująca:

$$f_{PAAO} = \begin{cases} f - f_{initial} & \text{jeżeli } f > f_{initial} \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad (4.10)$$

Wybrane przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących, jak również całkowity wskaźnik jakości regulacji i adaptowane współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu zostały przedstawione na rysunku 4.16. Czas adaptacji dla obu algorytmów, tj. algorytmu optymalizacji rojem cząstek oraz przeszukiwania wzorcem, nie został wydłużony i nadal mieści się w otrzymanym zakresie badań statystycznych. Całkowy wskaźnik jakości został zmniejszony do początkowej lub nieznacznie niższej wartości. Otrzymane odpowiedzi prędkości dla pierwszego okresu sygnału referencyjnego, dla pierwszego po zmianie momentu bezwładności oraz dla zaadaptowanego regulatora ze sprzężeniem od wektora zmiennych stanu wskazują jednoznacznie, że zaproponowana PAAO skutecznie radzi sobie z problemem niedopasowania modelu do kształtu dynamiki obiektu. Warto podkreślić, że odpowiedź układu po zakończonej adaptacji współczynników regulatora jest wolny od oscylacyjnego charakteru. Jego dopasowanie nie odwzorowuje idealnie odpowiedzi modelu odniesienia, lecz jest na zbliżonym poziomie uogólniania zadanej charakterystyki jak w przypadku początkowych współczynników oraz nominalnego momentu bezwładności. Zastosowanie uproszczonego modelu w celu zdefiniowania oczekiwanego charakteru odpowiedzi układu pozwala na duże uproszczenia w zakresie syntezy modelu odniesienia.



Rysunek 4.16. Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowane współczynniki regulatora oraz wybrane przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ ukazujące reakcję układu na zmianę momentu bezwładności przy nieadekwatnym rzędzie modelu odniesienia. Mechanizm adaptacji bazujący na algorytmie optymalizacyjnym

4.4. Podsumowanie

Zaproponowany mechanizm adaptacji bazujący na algorytmach optymalizacyjnych pozwala na zdefiniowanie zakładanej dokładności odwzorowania modelu odniesienia i zatrzymania adaptacji. Dzięki temu problemy dotyczące przeuczania się i błędów podczas symulacji nie występowały. Został zbadany wpływ najistotniejszych parametrów zaproponowanej procedury adaptacji dla algorytmów optymalizacyjnych na jakość adaptacji. Pomimo znacznej liczby parametrów, ich dobranie wydaje się stosunkowo intuicyjne ze względu na możliwość odniesienia do wartości występujących w układzie i ich zdefiniowanie w postaci procentowej. Z drugiej strony mnogość parametrów daje bardzo duże możliwości konfiguracyjne, które są niedostępne w regule Widrow'a-Hoff'a. Zmieniając wymaganą dokładność adaptacji i wartość progową wskaźnika powodującą wyzwolenie adaptacji, można skrócić czas adaptacji. Jednak pociąga to za sobą ograniczenie dokładności odwzorowania modelu odniesienia. Zmiany parametru określającego przestrzeń przeszukiwań algorytmu optymalizacyjnego powodują wydłużenie czasu adaptacji przy jednoczesnej redukcji prawdopodobieństwa otrzymania gorszego rozwiązania od obecnie najlepszego.

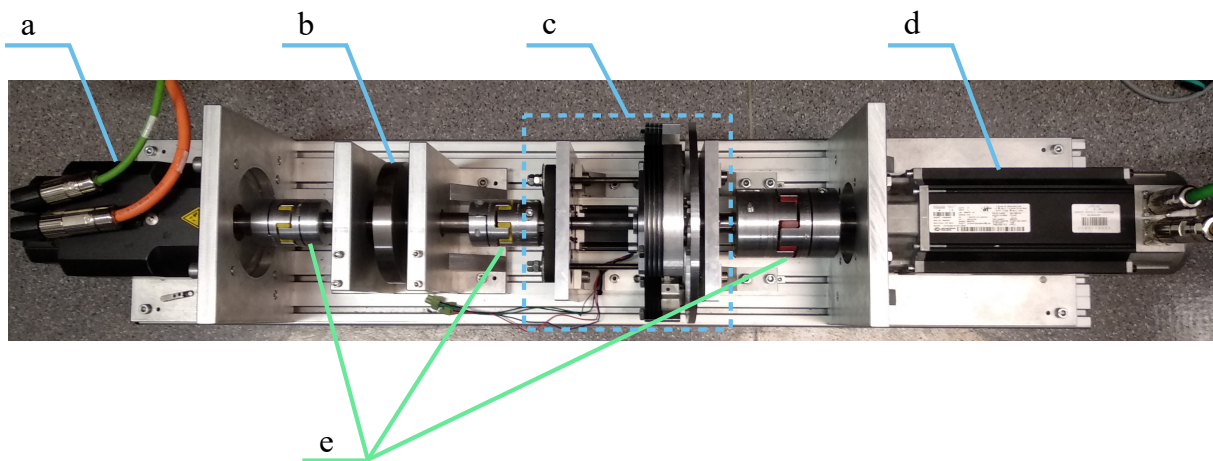
Porównano dwa algorytmy optymalizacyjne - algorytm optymalizacji rojem cząstek i algorytm optymalizacyjny przeszukiwania wzorcem. Oba algorytmy z sukcesem zoptymalizowały współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu we wszystkich przeprowadzonych próbach. Pozwala to na wysunięcie wniosku, że w analizowanym zakresie zmian momentu bezwładności nie występują minima lokalne, które spowodowałyby, że algorytm optymalizacyjny lokalny, jakim jest przeszukiwanie wzorcem, nie znajdowałby globalnego optimum zapewniającego oczekiwaną dokładność odwzorowania modelu odniesienia.

5. Wyniki badań eksperymentalnych

W niniejszym rozdziale zawarto opis stanowiska badawczego z napędem elektrycznym z silnikiem PMSM o zmiennym momencie bezwładności oraz wyniki badań eksperymentalnych. Przedstawiono również najważniejsze kwestie implementacyjne omawianych struktur regulacji adaptacyjnych w układzie mikroprocesorowym.

5.1. Opis stanowiska badawczego

Badania eksperymentalne zaproponowanego adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla silnika PMSM o zmiennym momencie bezwładności zostały przeprowadzone na stanowisku badawczym, którego zespół maszynowy został przedstawiony na rysunku 5.1.



Rysunek 5.1. Zdjęcie zespołu maszynowego stanowiska laboratoryjnego: a) badany silnik PMSM, b) krążek stalowy zwiększający moment bezwładności zespołu, c) urządzenie mechatroniczne o zmiennym momencie bezwładności, d) silnik służący jako obciążenie układu, e) sprężła

W jego skład wchodzi:

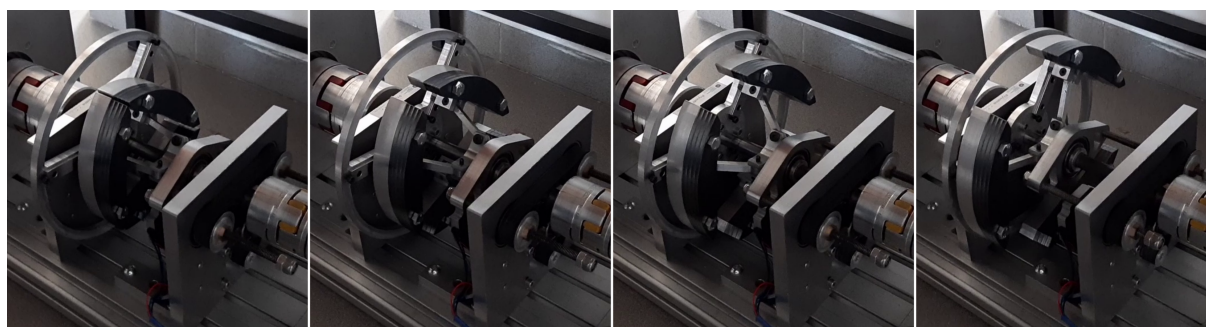
- badany silnik PMSM (ST-127-2-30-560/T1) o mocy 2.76 kW wyprodukowany przez firmę LTi Drives (najważniejsze parametry silnika zostały przedstawione w tabeli 5.1),

- silnik PMSM (115U2D300VAC) o mocy 3.3 kW wyprodukowany przez firmę EMERSON Industrial Automation służący jako obciążenie układu,
- walec stalowy zwiększający moment bezwładności zespołu,
- urządzenie mechatroniczne pozwalające zmieniać moment bezwładności w sposób ciągły.

Tabela 5.1. Parametry znamionowe badanego silnika

Parametr	Wartość	Jednostka
napięcie nominalne	400	VAC
stała momentu	1,65	Nm/Ams
skuteczny prąd nominalny	5,8	A
moment nominalny	8,8	Nm
moc nominalna	2,76	kW
liczba par biegunów	3	
prędkość nominalna	314,16	rad/s
całkowity moment bezwładności	$8,8 \times 10^{-3}$	kgm ²
indukcyjność międzyfazowa	19,02	mH
rezystancja międzyfazowa	2,1	Ω

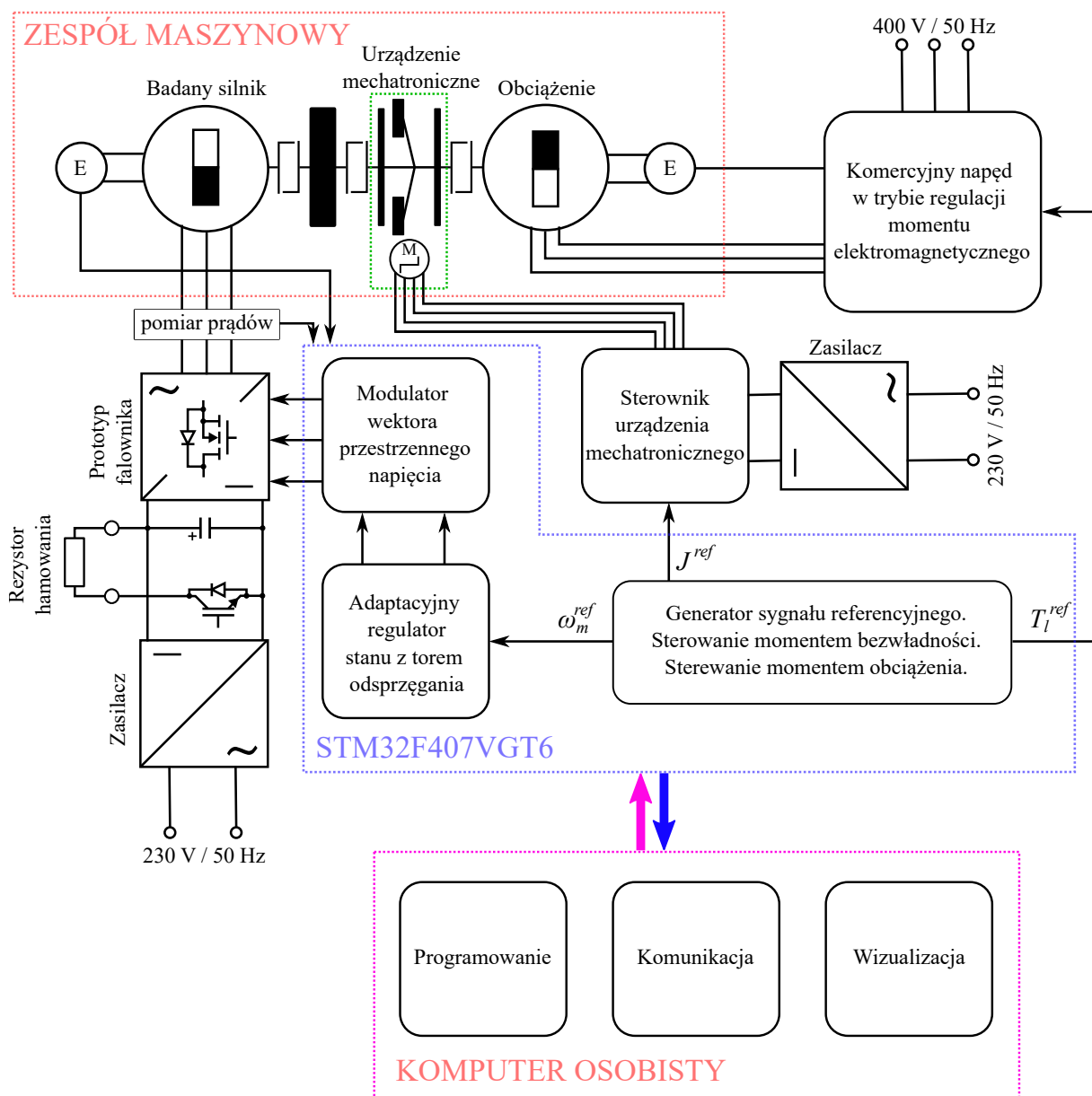
Całość jest połączona ze sobą za pomocą sprzęgieł kłowych i przytwierdzona do podstawy z profilu aluminiowego. Urządzenie mechatroniczne zmienia moment bezwładności układu poprzez rozszerzanie trzech symetrycznie rozłożonych stalowych elementów względem osi obrotu układu. Za zwiększanie oraz zmniejszanie rozpiętości tych ramion odpowiedzialny jest silnik krokowy, który jest sterowany przy użyciu opracowanego i wykonanego sterownika. Zdjęcia w kilku różnych konfiguracjach urządzenia mechatronicznego zostały przedstawione na rysunku 5.2.



Rysunek 5.2. Zdjęcie urządzenia mechatronicznego dla różnego rozstawu ramion

Sterownik urządzenia mechatronicznego został skalibrowany do maksymalnej oraz minimalnej pozycji ramion, a podczas pracy autonomicznej odczytuje sygnał analogowy i na tej podstawie steruje silnikiem krokowym, gdzie sygnał 0 V oznacza maksymalnie ściśnięte ramiona (minimalny moment bezwładności), a 3.3 V maksymalnie rozszerzone (maksymalny moment bezwładności). Dzięki temu urządzenie może być sterowane przez prototyp trójfazowego, dwupoziomowego falownika napięcia, co umożliwi uruchamianie złożonych sekwencji eksperymentalnych bez zatrzymywania eksperymentu na potrzeby montażu lub demontażu elementów zwiększających moment bezwładności.

Poglądowy schemat stanowiska laboratoryjnego został przedstawiony na rysunku 5.3.



Rysunek 5.3. Zdjęcie urządzenia mechatronicznego dla różnego rozstawu ramion

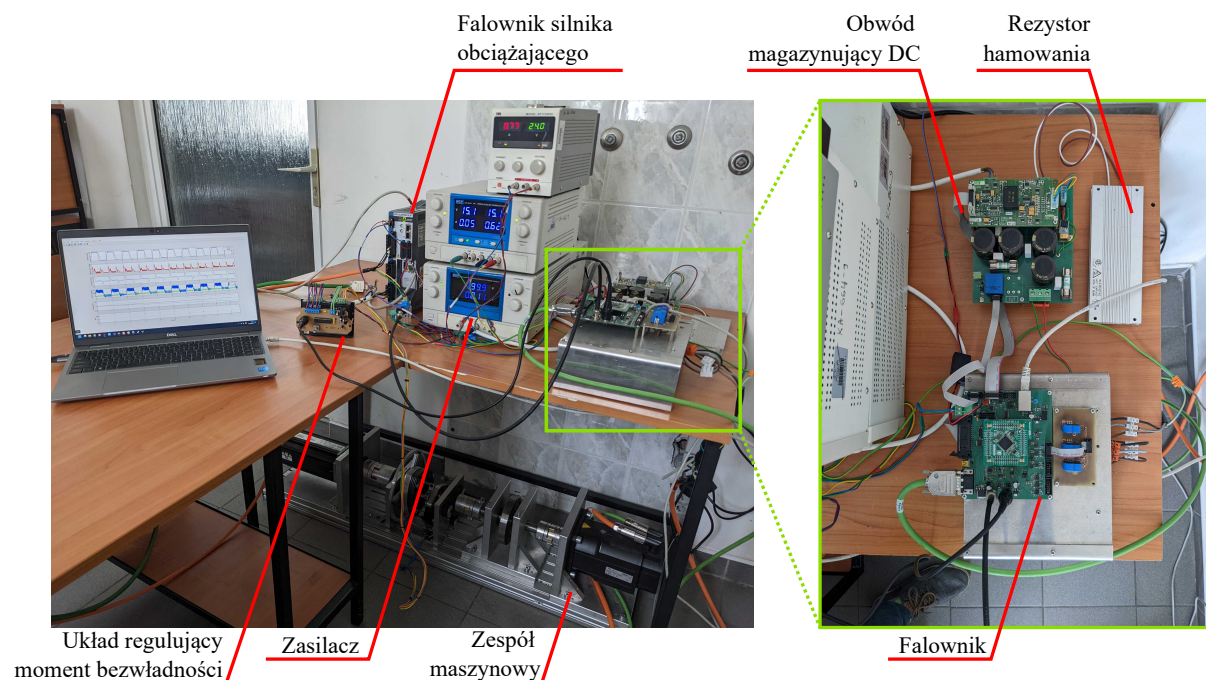
Do sterowania silnikiem obciążającym układ został wykorzystany komercyjny falownik napięcia AKD-P00307-NBCC-E000 wyprodukowany przez firmę Kollmorgen. Został on ustawiony w tryb regulacji momentu elektromagnetycznego i jest sterowany poprzez wejście analogowe oraz sygnał załączający. Prototyp trójfazowego, dwupoziomowego falownika napięcia jest wykorzystywany do sterowania badanym silnikiem. Najważniejsze komponenty układu to:

- mikrokontroler STM32F407VGT6 - są w nim zaimplementowane opracowane algorytmy sterowania wraz z modulatorem wektora przestrzennego napięcia,
- trójfazowy moduł tranzystorowy z tranzystorami na bazie węgla krzemu (SiC) CCS020M12CM2 firmy CREE oraz dedykowane sterowniki bramkowe CGD15FB45P1 tej samej firmy,
- przetworniki prądu LTS 15-NP firmy LEM - układ pomiaru prądów fazowych.

Elementem pomiarowym położenia kąтового silnika PMSM jest inkrementalny przetwornik obrotowo-impulsowy o sinusoidalnym rozkładzie impulsów typu SRS50 firmy SICK/STEGMANN z interfejsem HIPERFACE. Wykorzystane komponenty falownika pozwalają na pracę poza zakresem słyszalnym z częstotliwością 22 kHz. Ponadto wbudowane przetworniki cyfrowo-analogowe pozwalają na sterowanie sterownikiem urządzenia mechatronicznego oraz komercyjnym falownikiem do obciążania układu. Całe stanowisko jest kontrolowane przez komputer osobisty, który przesyła wybrany tryb regulacji, nastawy regulatorów oraz parametry kontrolne. Dodatkowo pozwala ono na wykonywanie długotrwałych eksperymentów ze zdefiniowanym harmonogramem sygnału referencyjnego, momentu bezwładności oraz momentu obciążającego układ, co jest niezbędne do weryfikacji opracowanych struktur regulacji adaptacyjnej. Zdjęcie stanowiska laboratoryjnego zostało przedstawione na rysunku 5.4.

5.2. Mechanizm adaptacji bazujący na regule Widrow'a Hoff'a

Badania symulacyjne zostały przeprowadzone na nieliniowym modelu silnika PMSM, natomiast nie uwzględniał on wszystkich nieliniowości i niedokładności pomiarowych rzeczywistego napędu elektrycznego z silnikiem PMSM, tj. szumy pomiarowe, błędy kwantyzacji prędkości, nieliniowe siły tarcia, moment zaczepowy. Jak zostało wykazane w badaniach symulacyjnych, reguła Widrow'a-Hoff'a posiada ograniczone możliwości uogólniania dla wyższych wartości współczynnika adaptacji. W związku z tym konieczne



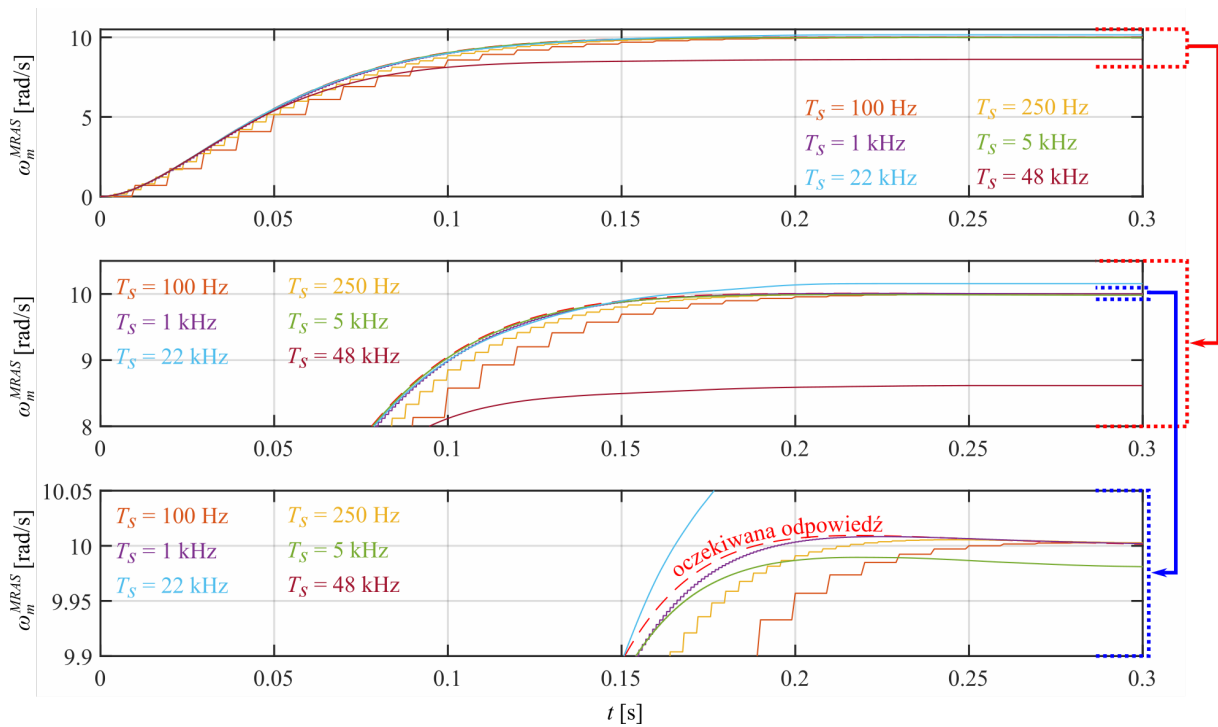
Rysunek 5.4. Zdjęcie stanowiska laboratoryjnego

jest uodpornienie mechanizmu adaptacji na przeuczenie się wynikające z próby dopasowania do powyższych aspektów napędu elektrycznego z silnikiem PMSM. W tym celu wprowadzono strefę nieczułości reguły adaptacji w zakresie ± 0.2 rad/s. Jeżeli różnica pomiędzy odpowiedzią modelu odniesienia a odpowiedzią układu mieści się w powyższym zakresie, to adaptacja jest pomijana w aktualnej pętli regulacji.

Ze względu na pojedynczą precyzję liczb zmiennoprzecinkowych (ang. *single-precision floating-point*) w układzie mikroprocesorowym, implementacja reguły Widrow'a-Hoff'a wymaga dodatkowych nakładów pracy związanych z implementacją i weryfikacją poprawności generowania modelu odniesienia. Problemem jest dokładność pojedynczej precyzji określana jako epsilon maszynowy (ϵ). Jest on definiowany jako najmniejsza liczba nieujemna, która po dodaniu do 1 daje wynik różny od 1, tj. $\epsilon + 1 \neq 1$, i dla liczb o pojedynczej precyzji wynosi $\epsilon = 1.1920929 \cdot 10^{-7}$. Jak zostało wykazane w badaniach symulacyjnych, niedokładność modelu może powodować oscylacyjny charakter odpowiedzi układu. Zdyskretyzowany model odniesienia określony równaniem (3.7) metodą prostokątów wstecz ma następującą postać:

$$\omega_m^{\text{MRAS}}(k) = \frac{2b_2b_1T_s\omega_m^{\text{MRAS}}(k-1) - b_2\omega_m^{\text{MRAS}}(k-2) + aT_s^2\omega_m^{\text{ref}}(k)}{b_2 + b_1T_s + b_0T_s^2} \quad (5.1)$$

Częstotliwość próbkowania wynosząca 22 kHz oznacza, że okres próbkowania wynosi $T_s = 45.(45) \cdot 10^{-5}$ s, a podniesienie go do potęgi drugiej generuje bardzo małą liczbę równą $T_s^2 = 2.066 \cdot 10^{-9}$ s. Powoduje to, że dokładność odwzorowania przy wysokich częstotliwościach próbkowania się pogarsza ze względu na precyzję liczb zmiennoprzecinkowych. Wpływ częstotliwości próbkowania na odpowiedź modelu referencyjnego generowanego w układzie mikroprocesorowym został przedstawiony na rysunku 5.5.



Rysunek 5.5. Wpływ częstotliwości próbkowania na odpowiedź modelu referencyjnego generowanego w układzie mikroprocesorowym o pojedynczej precyzji liczb zmiennoprzecinkowych

Niska częstotliwość próbkowania powoduje błędy kwantyzacji i jakość odpowiedzi, które w połączeniu z regulatorem adaptacyjnym bazującym na regule Widrow'a-Hoff'a mogłyby wprowadzić oscylacyjny charakter odpowiedzi. Wysoka częstotliwość powoduje uchyb w stanie ustalonym, który również mógłby powodować nieoczekiwany wynik adaptacji. Najlepsze dopasowanie uzyskuje się dla częstotliwości próbkowania wynoszącej 1 kHz. Ta wartość została wykorzystana do generowania modelu odniesienia w badaniach eksperymentalnych.

Dokładność liczb zmiennoprzecinkowych ma również negatywny wpływ na regułę adaptacji. Jak zostało wykazane w badaniach symulacyjnych, wysoka częstotliwość dla aktualizacji współczynnika regulatora przez regułę Widrow'a-Hoff'a powoduje, że jego aktualizacja w każdej iteracji musi być niewielka w celu zapewnienia nieoscylacyjnego

charakteru odpowiedzi. Ten fakt przekłada się bezpośrednio na problem z utratą informacji, na przykład jeżeli aktualna wartość współczynnika k_{ω_2} wynosi 1, a jego korekta Δk_{ω_2} jest mniejsza od epsilon maszynowego, to ta wartość nie zostanie zmieniona:

$$k_{\omega_2}(k) = k_{\omega_2}(k-1) + \Delta k_{\omega_2}(k) \quad \longrightarrow \quad k_{\omega_2}(k) = k_{\omega_2}(k-1)$$

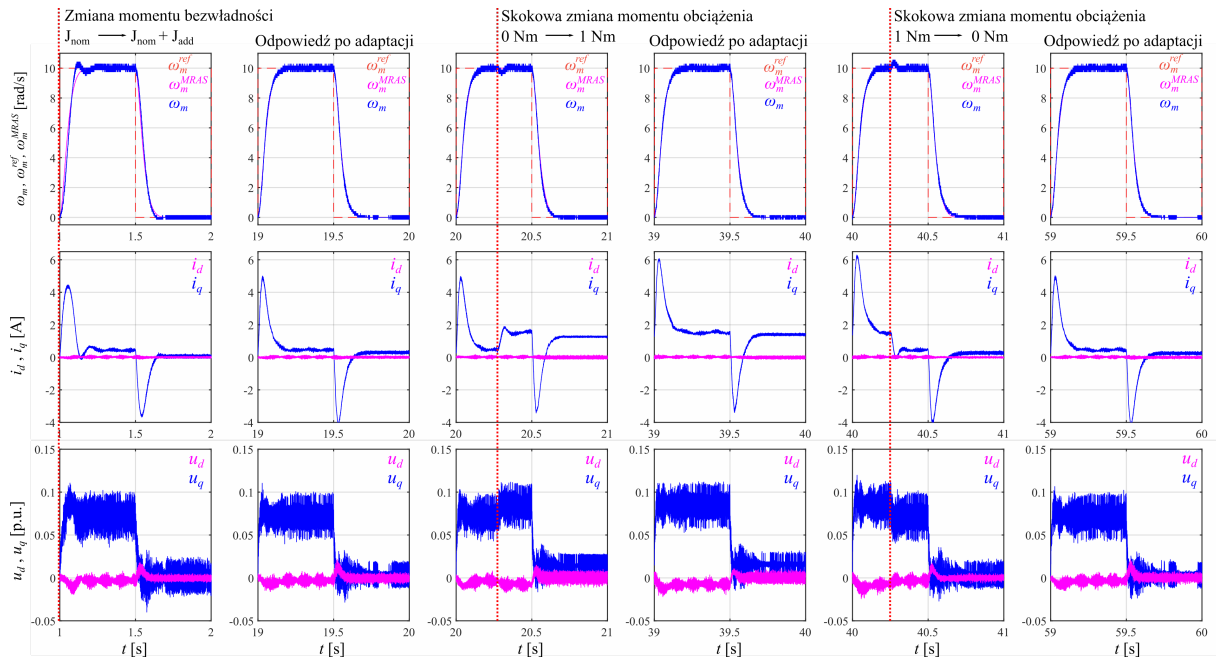
W celu rozwiązania tego problemu przyjęto zasadę zaproponowaną przez autora niniejszej rozprawy doktorskiej w pracy [20]. Prawo sterowania regulatora ze sprzężeniem od wektora zmiennych stanu jest zaimplementowane jako suma początkowych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu ($\mathbf{K}_{initial}$) przemnożonych przez wektor zmiennych stanu oraz zsumowanych aktualizacji współczynników z reguły Widrow'a-Hoff'a ($\Delta\mathbf{K}$) przemnożonych przez wektor zmiennych stanu:

$$\mathbf{u}(t) = \mathbf{K}_{initial}\mathbf{x}(t) + \Delta\mathbf{K}(t)\mathbf{x}(t) \quad (5.2)$$

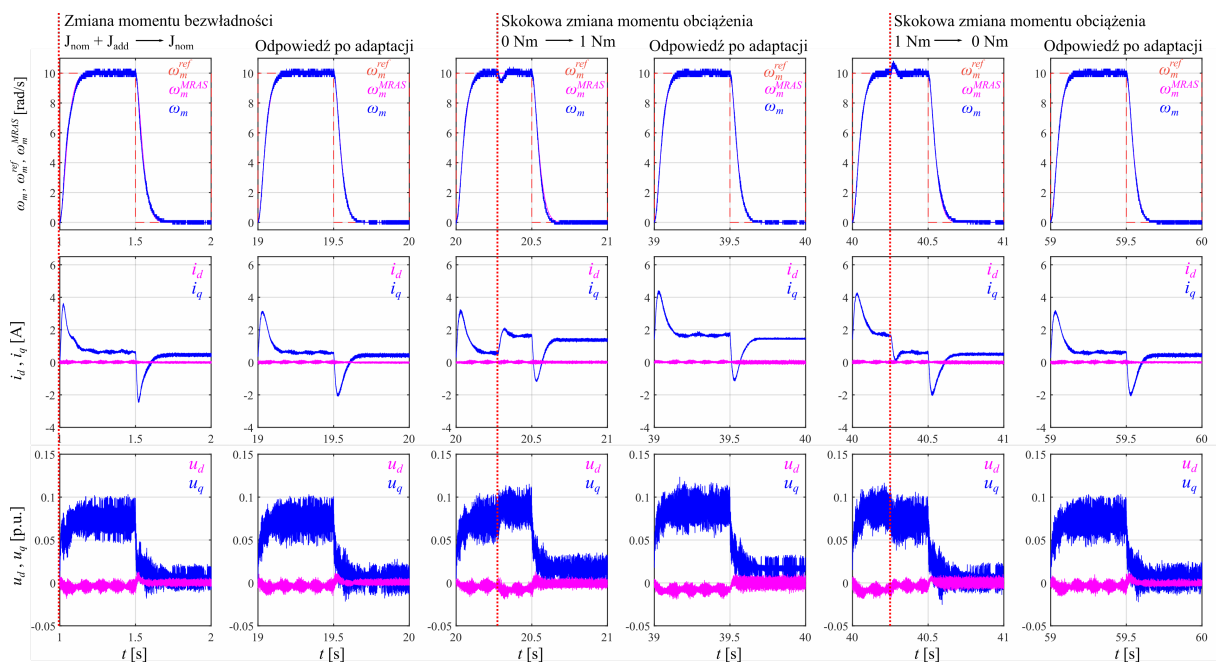
Dzięki temu aktualizacja współczynnika regulatora w każdej pętli regulacji nie jest dodawana do jego początkowej wartości, a jest sumowana w odrębnej zmiennej bez częściowej lub pełnej utraty korekcji.

5.2.1. Rezultaty

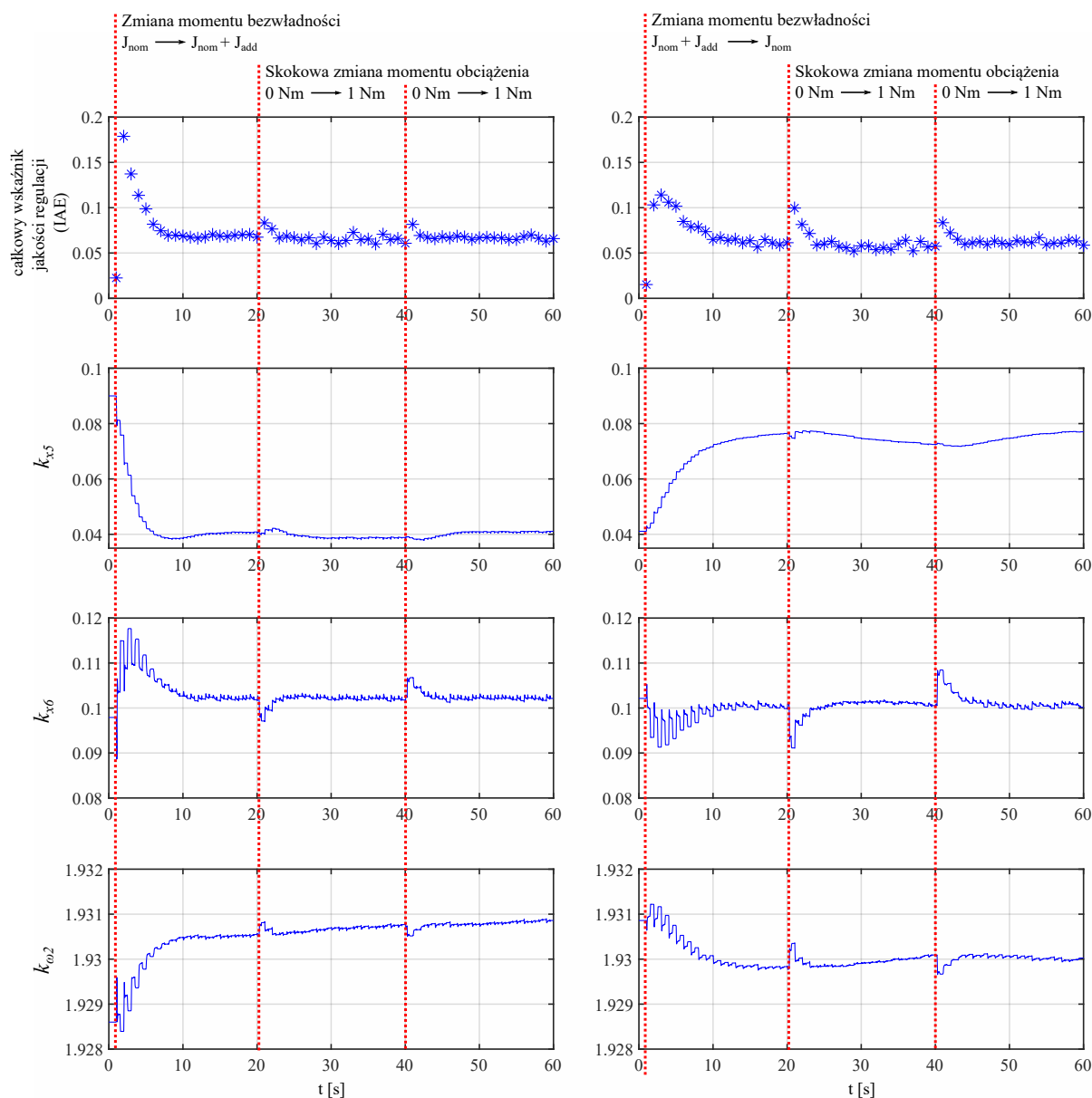
Badania eksperymentalne zostały podzielone na dwa eksperymenty trwające 60 sekund. Sygnał referencyjny jest tożsamy z wykorzystanym w badaniach symulacyjnych, tj. sygnał prostokątny o wartości maksymalnej równej 10 rad/s oraz wartości minimalnej 0 rad/s i częstotliwości 1 Hz. Eksperyment pierwszy rozpoczyna pracę z początkowymi wartościami nastaw regulatora oraz ze zwiększonym momentem bezwładności o J_{add} . Eksperyment drugi rozpoczyna pracę z wartościami nastaw regulatora odpowiadającymi wartościom otrzymanym w eksperymencie pierwszym oraz z nominalnym momentem bezwładności. W obu przypadkach zbadany został wpływ na skokowe zmiany momentu obciążenia - skok do wartości 1 Nm w 20.25 sekundzie oraz do 0 Nm w 40.25 sekundzie. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego zostały przedstawione na rysunkach 5.6-5.7 odpowiednio dla eksperymentu pierwszego oraz drugiego. Przebiegi czasowe całkowego wskaźnika jakości regulacji oraz adaptacyjnych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu zostały przedstawione na rysunku 5.8.



Rysunek 5.6. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego eksperymentu przy zwiększonym momencie bezwładności. Mechanizm adaptacji bazujący na regule Widrw'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$



Rysunek 5.7. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego eksperymentu przy zmniejszonym momencie bezwładności do nominalnej wartości. Mechanizm adaptacji bazujący na regule Widrw'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$



Rysunek 5.8. Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowanych współczynników regulatora dla eksperymentu ze zwiększoną wartością momentu bezwładności oraz ze zmniejszoną do nominalnych wartości. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$

Przedstawione wyniki badań eksperymentalnych są zbieżne z badaniami symulacyjnymi. Reguła Widrow'a-Hoff'a skutecznie wyeliminowała przeregulowanie, które powstało w wyniku skokowej zmiany momentu bezwładności. Otrzymane przebiegi prędkości są wolne od oscylacji, a proces adaptacji trwał kilkanaście sekund po zmianie momentu bezwładności oraz pojedyncze sekundy przy wystąpieniu skokowej zmiany momentu obciążenia. Warto zaznaczyć, że zastosowanie reguły Widrow'a-Hoff'a pozwoliło również na lepszą kompensację zakłóceń w porównaniu do nieadaptacyjnego regulatora ze sprzężeniem od wektora zmiennych

stanu. Wynika to z faktu, że powstała różnica w odpowiedziach układu i modelu odniesienia skutkowałą adaptacją do nowych warunków pracy napędu w każdej pętli regulacji. Otrzymany całkowity wskaźnik jakości regulacji dla zwiększonego momentu bezwładności oscyluje w okolicy wartości 0.07, a dla nominalnego momentu bezwładności w okolicy 0.06. Różnica pomiędzy tymi wartościami jest związana z wykorzystanym urządzeniem mechatronicznym, które podczas zmiany momentu bezwładności powoduje również zmianę współczynnika tarcia wiskotycznego oraz statycznego, co zostało przedstawione w pracy [80], w której zaproponowano metodę identyfikacji parametrów mechanicznych układu maszynowego napędu elektrycznego.

5.3. Mechanizm adaptacji bazujący na algorytmie optymalizacyjnym

Analogicznie do badań symulacyjnych model odniesienia został zaimplementowany w ten sam sposób, co dla mechanizmu adaptacji bazującego na regule Widrow'a-Hoff'a w celu umożliwienia porównania obu mechanizmów adaptacji. Niemniej jednak zaproponowana PAAO pozwala na zaimplementowanie prostszego modelu, co zostało wykazane w badaniach symulacyjnych. Ze względu na szумы pomiarowe, moment zaczepowy, nieliniowe tarcie i inne aspekty wpływające na całkowite kryterium jakości regulacji automatycznej, analogicznie do metody zaproponowanej przy modelu odniesienia o nieadekwatnym rzędzie, wprowadzono korektę do wartości funkcji celu, korzystając z zależności (4.10). Za $f_{initial}$ przyjęto wartość 0.06, która odpowiada dopasowaniu odpowiedzi układu o początkowych współczynnikach regulatora i nominalnego momentu bezwładności do modelu odniesienia. Pozwoliło to na zastosowania parametrów PAAO wykorzystanych podczas badań symulacyjnych.

Dzięki okienkowaniu funkcji celu dokładność liczb zmiennoprzecinkowych o pojedynczej precyzji nie wpływa na działanie zaproponowanego mechanizmu adaptacji. W przypadku adaptacji współczynników regulatora ze sprzężeniem od wektora zmiennych stanu są one aktualizowane raz na okres sygnału referencyjnego o stosunkowo dużą wartość w porównaniu do aktualizacji wykonywanej przez regułę Widrow'a-Hoff'a. Kwestie związane z problemem implementacji dyskretnej transmitancji drugiego rzędu mogą zostać rozwiązane stosując niższy rząd, z którym PAAO skutecznie sobie radzi lub zastosować metodę zaproponowaną w pracy [81], w której jako model odniesienia przyjęto odpowiedź układu na pierwszy okres

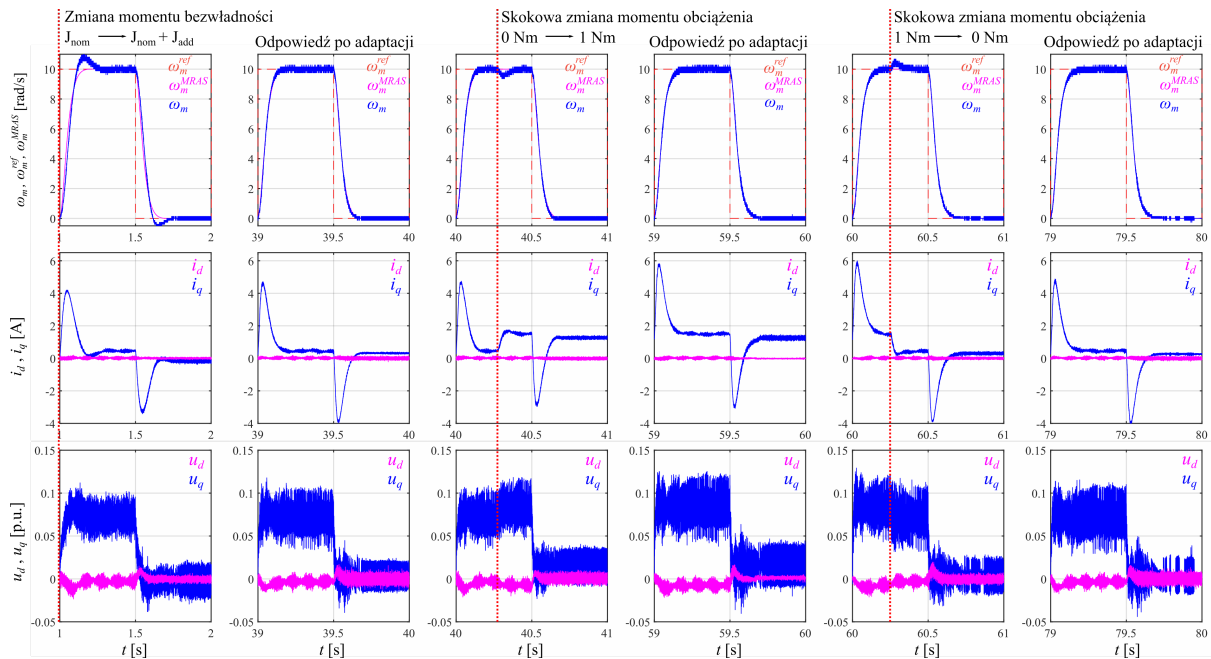
sygnału referencyjnego (tj. dla początkowych współczynników regulatora oraz nominalnych parametrów obiektu) zarchiwizowaną w postaci tablicy w układzie mikroprocesorowym.

Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a wykonuje operacje związane z aktualizacją współczynników w każdej pętli regulacji. PAAO jest uruchamiana tylko podczas rozpoczynającego się nowego okresu sygnału referencyjnego, czyli podczas zbocza narastającego sygnału referencyjnego w tym szczególnym przypadku. W trakcie trwania okresu sygnału referencyjnego wyznaczana jest funkcja celu zdefiniowana jako całkowity wskaźnik jakości regulacji jakim jest całka z wartości bezwzględnej uchybu. W związku z generowanym modelem odniesienia z częstotliwością 1 kHz, występującymi błędami kwantyzacji podczas wyznaczania prędkości oraz szumami pomiarowymi funkcja celu jest również aktualizowana z tą samą częstotliwością o wartość średnią z 22 pomiarów prędkości (ze względu na częstotliwość próbkowania wynoszącą 22 kHz).

5.3.1. Rezultaty - algorytm optymalizacji rojem cząstek

Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego zostały przedstawione na rysunkach 5.9-5.10 odpowiednio dla eksperymentu pierwszego oraz drugiego. Przebiegi czasowe całkowego wskaźnika jakości regulacji oraz adaptacyjnych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu zostały przedstawione na rysunku 5.11. Ze względu na dłuższy czas zbieżności algorytmu optymalizacji rojem cząstek, skokowe zmiany momentu obciążenia zostały wyzwalone w 40.25 i 60.25 sekundzie, a cały eksperyment został wydłużony do 80 sekund. Mechanizm adaptacji skutecznie zminimalizował różnicę pomiędzy odpowiedzią układu a odpowiedzią modelu odniesienia wynikającą ze zwiększenia, a następnie zmniejszenia momentu bezwładności. Występujące skokowe zmiany momentu obciążenia powodowały ponowną inicjalizację algorytmu, lecz w trzech przypadkach już podczas inicjalizacji testowane współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu uzyskały dokładność odwzorowania na zakładanym poziomie całkowego wskaźnika jakości regulacji. Dla eksperymentu z przywróconą wartością nominalną momentu bezwładności skokowa zmiana momentu obciążenia z 1 Nm do 0 Nm wyzwoliła adaptację, która trwała 14 sekund. Różne reakcje i czasy zbieżności wynikają z wykorzystania generatora liczb pseudolosowych, który jest wykorzystywany do ponownej inicjalizacji roju oraz aktualizacji prędkości cząstek (równania (4.9) oraz (4.7)). W przypadku skokowej

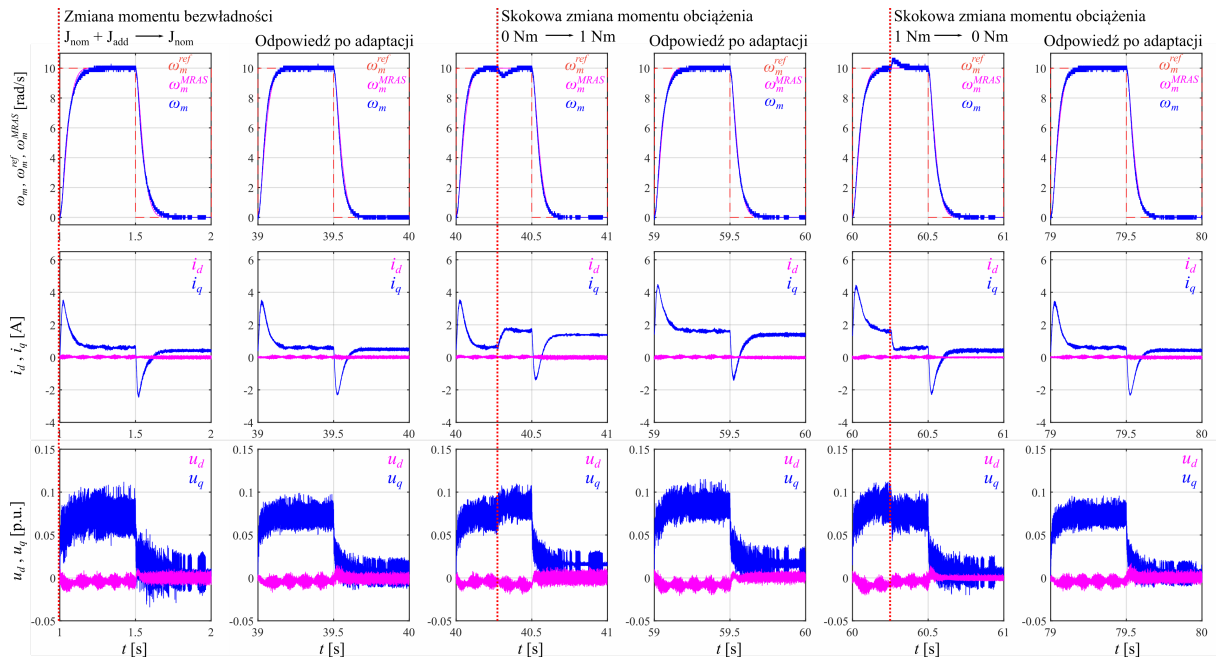
zmiany momentu bezwładności algorytm zakończył proces adaptacji po 25 oraz 28 sekundach odpowiednio dla eksperymentu pierwszego oraz drugiego.



Rysunek 5.9. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zwiększonym momencie bezwładności. Mechanizm adaptacji bazujący algorytmie optymalizacji rojem cząstek.

5.3.2. Rezultaty - algorytm przeszukiwania wzorcem

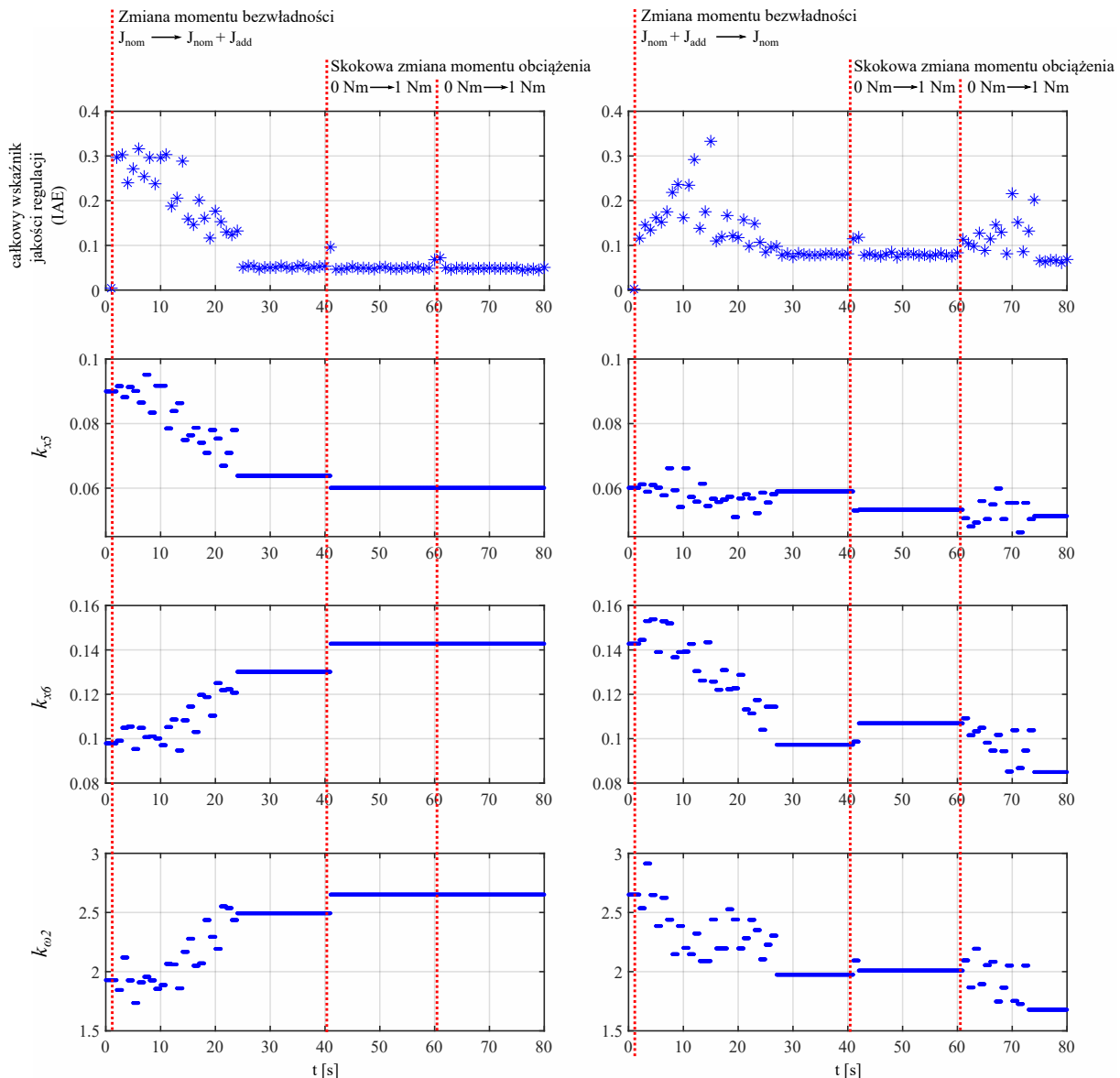
Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego zostały przedstawione na rysunkach 5.12-5.13 odpowiednio dla eksperymentu pierwszego oraz drugiego. Przebiegi czasowe całkowego wskaźnika jakości regulacji oraz adaptacyjnych współczynników regulatora ze sprzężeniem od wektora zmiennych stanu zostały przedstawione na rysunku 5.14. Przeregulowanie i wydłużenie się czasu regulacji podczas zwiększenia momentu bezwładności oraz zbyt wysoka dynamika podczas zmniejszenia momentu bezwładności do wartości nominalnej zostały skutecznie zniwelowane w badanym horyzoncie czasowym. Wystąpienie skokowej zmiany momentu obciążenia wywołało proces adaptacji, lecz w każdym badanym przypadku algorytm zakończył pracę w czasie poniżej 10 sekund i uzyskał dopasowanie odpowiedzi obiektu do odpowiedzi modelu odniesienia o zakładanej dokładności. Całkowy wskaźnik jakości regulacji po adaptacji dla wszystkich przeprowadzonych testów mieścił się w zakresie 0.5-0.8. Warto zauważyć, że podczas powrotu



Rysunek 5.10. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego eksperymentu przy zmniejszonym momencie bezwładności do nominalnej wartości. Mechanizm adaptacji bazujący algorytmie optymalizacji rojem cząstek.

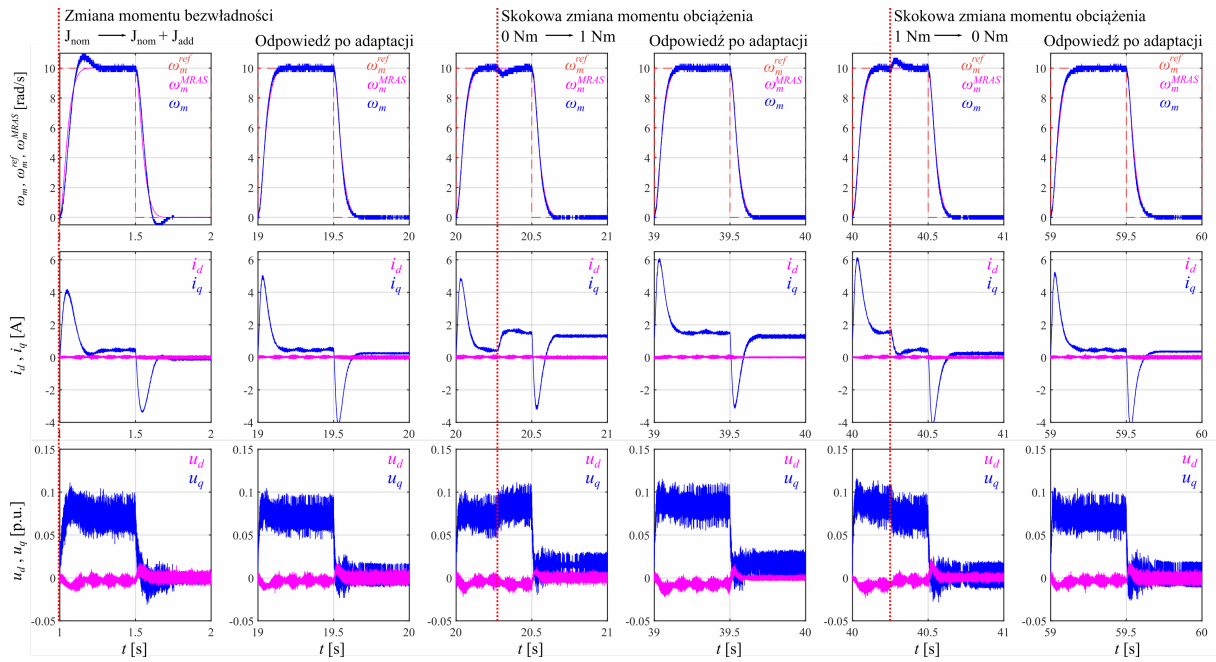
do nominalnego momentu bezwładności i przy skokowej zmianie momentu obciążenia do 1 Nm całkowity wskaźnik jakości regulacji jest znacznie niższy niż w pozostałych sytuacjach. Wynika to z losowości algorytmu optymalizacyjnego, który znalazł rozwiązania zdecydowanie lepsze niż oczekiwana dokładność, która zatrzymuje adaptację. Ta sama losowość wpływa na czas adaptacji, co zostało przedstawione w badaniach symulacyjnych poprzez dziesięciokrotne uruchomienie eksperymentu.

Porównanie współczynników regulatora ze sprzężeniem od wektora zmiennych stanu oraz całkowitego wskaźnika jakości regulacji dla wszystkich badanych struktur regulacji adaptacyjnej po każdym z etapów testu pierwszego oraz drugiego zostały przedstawione w tabeli 5.2. Wartości całkowitego wskaźnika jakości regulacji dla każdego momentu bezwładności oraz warunków pracy napędu są zbliżone dla każdego z zastosowanych mechanizmów adaptacji. Potencjalne różnice nie są wyższe niż niepewność całkowitego wskaźnika jakości wynikająca z szumów pomiarowych. Pozwala to wyciągnąć wniosek, że każdy z mechanizmów adaptacji skutecznie dopasował odpowiedź układu do odpowiedzi modelu referencyjnego. W przypadku współczynników regulatora ze sprzężeniem od wektora zmiennych stanu różnice pomiędzy poszczególnymi mechanizmami adaptacji są bardzo dobrze widoczne. Reguła Widrow'a-Hoff'a zmieniała współczynniki w najmniejszym stopniu,

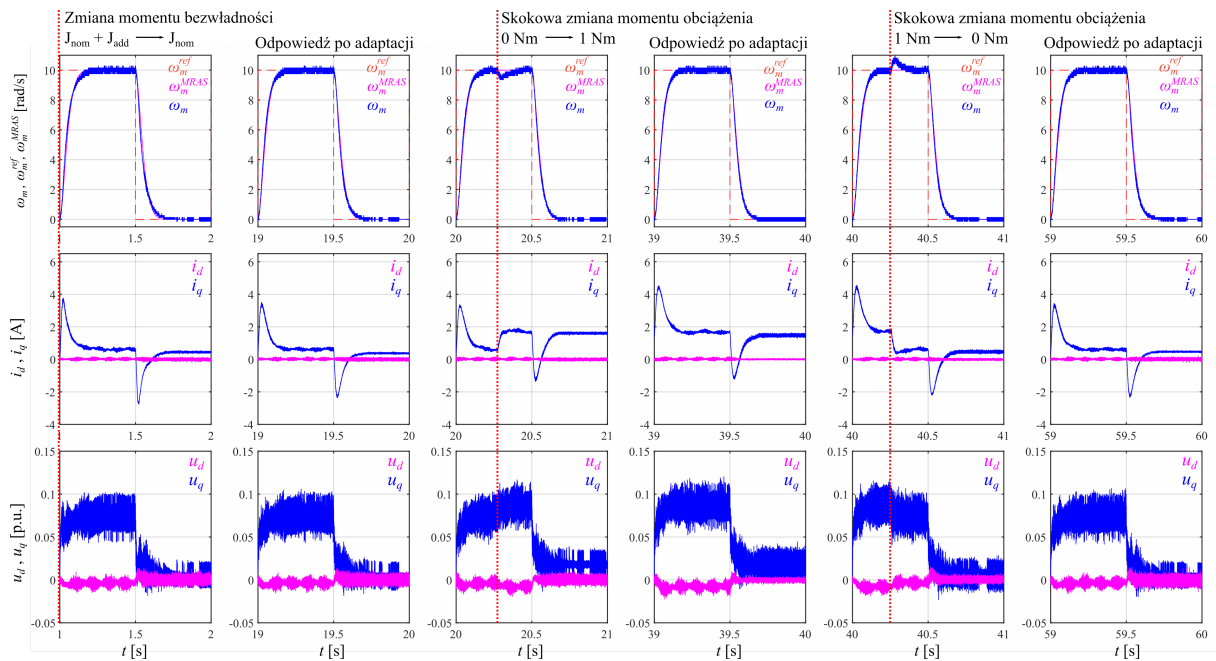


Rysunek 5.11. Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowanych współczynników regulatora dla eksperymentu ze zwiększoną wartością momentu bezwładności oraz ze zmniejszoną do nominalnych wartości. Mechanizm adaptacji bazujący algorytmie optymalizacji rojem cząstek.

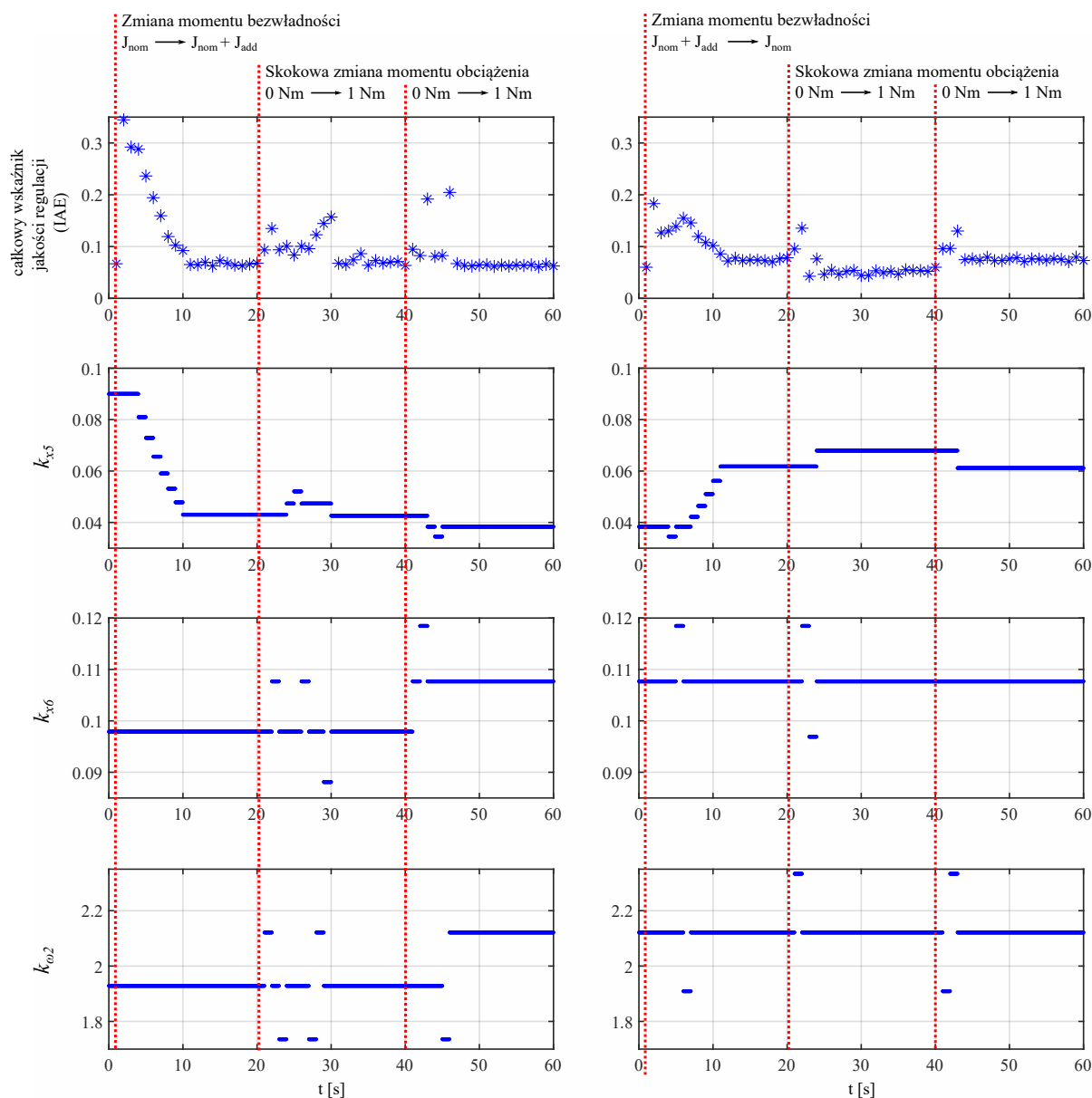
a zaproponowana PAAO z algorytmem optymalizacji rojem cząstek w największym. Na przykład k_{w2} przyjmowała wartości w zakresie $\langle 1.9286, 1.9309 \rangle$ dla reguły Widrow'a-Hoff'a oraz $\langle 1.6788, 2.6509 \rangle$ dla optymalizacji rojem cząstek. Współczynniki zoptymalizowane przez algorytm przeszukiwania wzorcem są zbliżone do tych znalezionych przez regułę Widrow'a-Hoff'a. Jest to spowodowane zasadą działania algorytmów optymalizacyjnych. Optymalizacja rojem cząstek w największym stopniu bazuje na liczbach pseudolosowych i wykorzystuje agentów do eksploracji przestrzeni rozwiązań.



Rysunek 5.12. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego eksperymentu przy zwiększonym momencie bezwładności. Mechanizm adaptacji bazujący algorytmie optymalizacyjnym przeszukiwania wzorcem.



Rysunek 5.13. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych d - q dla kluczowych okresów sygnału referencyjnego eksperymentu przy zmniejszonym momencie bezwładności do nominalnej wartości. Mechanizm adaptacji bazujący algorytmie optymalizacyjnym przeszukiwania wzorcem.



Rysunek 5.14. Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowanych współczynników regulatora dla eksperymentu ze zwiększoną wartością momentu bezwładności oraz ze zmniejszoną do nominalnych wartości. Mechanizm adaptacji bazujący algorytmie optymalizacyjnym przeszukiwania wzorcem.

Powoduje to zmiany we wszystkich współczynnikach równocześnie. Algorytm przeszukiwania wzorcem modyfikuje aktualne rozwiązanie tylko w jednym wymiarze jednocześnie i jeżeli nowe rozwiązanie nie jest lepsze, to powraca do poprzedniego, tj. rozwiązanie nie jest aktualizowane trwale.

Tabela 5.2. Otrzymany całkowity wskaźnik jakości regulacji (IAE) oraz współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu dla wszystkich badanych struktur regulacji adaptacyjnej po każdym z etapów testu pierwszego oraz drugiego

Wartości początkowe				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	–	0.0900	0.0979	1.9286
PAAO-PSO	–	0.0900	0.0979	1.9286
PAAO-PS	–	0.0900	0.0979	1.9286
Skokowa zmiana momentu bezwładności $J_{nom} \rightarrow J_{nom} + J_{add}$				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	0.0675	0.0407	0.1022	1.9305
PAAO-PSO	0.0533	0.0638	0.1301	2.4934
PAAO-PS	0.0678	0.0430	0.0979	1.9286
Skokowa zmiana momentu obciążenia 0 Nm \rightarrow 1 Nm				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	0.0608	0.0387	0.1020	1.9308
PAAO-PSO	0.0679	0.0601	0.1428	2.6509
PAAO-PS	0.0634	0.0426	0.0979	1.9286
Skokowa zmiana momentu obciążenia 1 Nm \rightarrow 0 Nm				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	0.0659	0.0409	0.1017	1.9309
PAAO-PSO	0.0507	0.0601	0.1428	2.6509
PAAO-PS	0.0625	0.0384	0.1077	2.1215
Skokowa zmiana momentu bezwładności $J_{nom} + J_{add} \rightarrow J_{nom}$				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	0.0614	0.0762	0.1000	1.9298
PAAO-PSO	0.0812	0.0590	0.0973	1.9745
PAAO-PS	0.0783	0.0618	0.1077	2.1215
Skokowa zmiana momentu obciążenia 0 Nm \rightarrow 1 Nm				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	0.0574	0.0726	0.1005	1.9301
PAAO-PSO	0.0835	0.0533	0.1069	2.0091
PAAO-PS	0.0600	0.0679	0.1077	2.1215
Skokowa zmiana momentu obciążenia 1 Nm \rightarrow 0 Nm				
Mechanizm adaptacji	IAE	k_{x5}	k_{x6}	$k_{\omega 2}$
Reguła Widrow'a-Hoff'a	0.0587	0.0771	0.1007	1.9300
PAAO-PSO	0.0684	0.0514	0.0850	1.6788
PAAO-PS	0.0730	0.0612	0.1077	2.1215

5.4. Podsumowanie

Wszystkie mechanizmy adaptacji skutecznie zniwelowały wpływ zmiany momentu bezwładności na odpowiedź układu oraz zapewniły niezmienną charakterystykę odpowiedzi układu. Mechanizm bazujący na regule Widrow'a-Hoff'a wymagał zdecydowanie większych

nakładów pracy podczas implementacji w układzie mikroprocesorowym ze względu na konieczność obliczeń na liczbach zmiennoprzecinkowych o pojedynczej precyzji. Brak analizy wpływu niższej precyzji na jakość generowanego modelu odniesienia skutkowałaby przeuczeniem się algorytmu i doprowadzeniem do oscylacyjnego charakteru odpowiedzi układu. Ponadto aktualizacja współczynników regulatora wprost, tj. sumowania początkowych wartości i aktualizacji w każdej pętli regulacji, powodowałaby utratę informacji ze względu na dokładność liczb zmiennoprzecinkowych o pojedynczej precyzji. Rezultatem byłyby konieczność ponownego doboru współczynnika adaptacji lub wydłużenie czasu niezbędnego na uzyskanie zbieżności procesu adaptacji. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a uwzględniający wszystkie możliwe problemy z dokładnością obliczeń w układzie mikroprocesorowym, wyborem modelu odniesienia o adekwatnym rzędzie oraz wybór współczynnika adaptacji, uwzględniając oczekiwany nieoscylacyjny charakter odpowiedzi i stosunkowo krótki czas adaptacji, pozwolił na uzyskanie bardzo dobrych parametrów procesu adaptacji.

Zaproponowana PAAO również skutecznie doprowadziła do zachowania charakterystyki odpowiedzi układu podczas zmian momentu bezwładności. Zgodnie z przedstawionymi wnioskami w pracy "*No free lunch theorems for optimization*" [76], wybór algorytmu optymalizacyjnego ma bardzo duży wpływ na jakość rozwiązania problemu optymalizacyjnego. Algorytm optymalizacji rojem cząstek potrzebował więcej czasu na zbieżność niż w przypadku badań symulacyjnych. Powodem może być dokładność liczb zmiennoprzecinkowych, słabej jakości generator liczb pseudolosowych zastosowany w układzie mikroprocesorowym lub szumy pomiarowe powodujące niepewność całkowego wskaźnika jakości regulacji. Niemniej algorytm optymalizacji rojem cząstek osiągnął oczekiwaną dokładność odwzorowania modelu odniesienia w każdym badanych warunkach pracy napędu elektrycznego z silnikiem PMSM. Mechanizm adaptacji bazujący na algorytmie przeszukiwania wzorcem uzyskał wyniki zbliżone do badań symulacyjnych, ponieważ występująca losowość dotyczy tylko wyboru kolejnej z sześciu możliwości korekcji współczynników regulatora, które są deterministyczne. Uzyskane wyniki dla algorytmu optymalizacyjnego przeszukiwania wzorcem i reguły Widrow'a-Hoff'a są zbliżone, podczas gdy zaproponowana PAAO nie wymagała dodatkowych nakładów pracy związanych z zapewnieniem poprawności działania na układzie mikroprocesorowym.

6. Wnioski

Celem niniejszej rozprawy było opracowanie struktury adaptacyjnej bazującej na algorytmach optymalizacyjnych inspirowanych przyrodą w celu zapewnienia niezmiennej odpowiedzi silnika PMSM o zmiennym momencie bezwładności. W tym celu, opracowano autorską procedurę adaptacji dla algorytmów optymalizacyjnych (PAAO), która umożliwia zastosowanie większości algorytmów optymalizacyjnych, w tym inspirowanych przyrodą, do adaptacji współczynników regulatora ze sprzężeniem od wektora zmiennych stanu. Przedstawione wyniki badań symulacyjnych oraz eksperymentalnych pokazują, że postawiony cel został osiągnięty.

W początkowej części pracy przedstawiono model matematyczny napędu elektrycznego z silnikiem PMSM, syntezę regulatora ze sprzężeniem od wektora zmiennych stanu oraz dokonano analizy stosowanych w literaturze struktur regulacji adaptacyjnej w napędach elektrycznych. Wybór regulatora ze sprzężeniem od wektora zmiennych stanu był motywowany jego odpornością i tolerancją na nieliniowości występujące w obiekcie, lepszymi właściwościami dynamicznymi, lepszą kompensacją zakłóceń oraz możliwością sterowania nieliniowym obiektem. Niniejsza rozprawa doktorska skupiała się na zastosowaniu sterowania adaptacyjnego z modelem odniesienia, które pozwala na zapewnienie stałości odpowiedzi układu niezależnie od parametrów obiektu oraz warunków pracy.

Bazując na powyższych założeniach, opracowano autorską PAAO. Dzięki cyklicznej ewaluacji najlepszego rozwiązania możliwe jest wykrycie zmian parametrów obiektu lub warunków pracy bez zastosowania dodatkowych estymatorów, tj. estymator momentu obciążenia, estymator momentu bezwładności. Wykryta zmiana wyzwala proces adaptacji, za który odpowiada algorytm optymalizacyjny inspirowany przyrodą lub klasyczny. Bazując na przeglądzie literatury, w badaniach symulacyjnych porównano algorytm inspirowany przyrodą, który jest optymalizatorem globalnym, z algorytmem optymalizacji lokalnej. Przedstawicielem algorytmów inspirowanych przyrodą był algorytm optymalizacji rojem

częstek, ze względu na jego popularność w literaturze naukowej oraz bardzo dobre właściwości optymalizacyjne. W przypadku algorytmu optymalizacji lokalnej, głównym założeniem był brak wymaganego wyliczania gradientu funkcji, co spowodowało, że wybrano algorytm przeszukiwania wzorcem. Zaproponowana PAAO z wybranymi algorytmami optymalizacyjnymi została szczegółowo omówiona, a następnie przebadana w modelu symulacyjnym opracowanym w środowisku MATLAB/SIMULINK. Przebadano wpływ zarówno parametrów samej PAAO, jak i algorytmów optymalizacyjnych.

W celu porównania opracowanej struktury sterowania adaptacyjnego ze strukturą klasyczną został szczegółowo omówiony mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a. W badaniach symulacyjnych zostały wykazane potencjalne problemy z podejściem klasycznym. Pierwszym jest konieczność stosowania modelu odniesienia o adekwatnym rzędzie, ponieważ w przypadku braku możliwości idealnego odwzorowania odpowiedzi modelu odniesienia przez odpowiedź układu dochodzi do tzw. przeuczania, co prowadzi do oscylacyjnego charakteru odpowiedzi. Drugim jest dobór współczynnika uczenia. Wysokie wartości skutkowały krótkim czasem adaptacji, lecz powodowały oscylacyjny charakter odpowiedzi, który przyjęto za niedopuszczalny.

Badania eksperymentalne uzupełniły weryfikację symulacyjną, wykorzystując stanowisko badawcze wyposażone w zespół maszynowy z badanym silnikiem oraz silnikiem obciążającym, prototypowy falownik, komercyjny napęd do sterowania silnikiem obciążającym oraz urządzenie mechatroniczne pozwalające na zmianę momentu bezwładności. Podczas implementacji mechanizmów adaptacji w układzie mikroprocesorowym zweryfikowano kolejne potencjalne problemy z podejściem klasycznym – regułą Widrow'a-Hoff'a. Kontynuując numerację z testów symulacyjnych trzecim problemem reguły Widrow'a-Hoff'a jest kwestia dokładności liczb zmiennoprzecinkowych o pojedynczej precyzji. W badanym napędzie przyjęto częstotliwość pętli regulacji równą 22 kHz, co poskutkowało, że implementacja modelu odniesienia z tą samą częstotliwością próbkowania utrzymywała niewielki uchyb w stanie ustalonym, który w dłuższym horyzoncie pracy napędu mógłby prowadzić do oscylacyjnego charakteru odpowiedzi prędkości kątowej. Ponadto aktualizacja współczynników regulatora w każdej pętli regulacji powoduje, że współczynnik uczenia jest stosunkowo niewielki i wartość aktualizacji współczynnika częściowo lub w całości jest tracona ze względu na epsilon maszynowy dla liczb zmiennoprzecinkowych o pojedynczej precyzji. Implementacja reguły

Widrow'a-Hoff'a w układzie mikroprocesorowym wymagała dodatkowych nakładów pracy w celu osiągnięcia oczekiwanych rezultatów.

W przypadku implementacji PAAO w układzie mikroprocesorowym jedyna modyfikacja polegała na uwzględnieniu wpływu szumów pomiarowych na całkowity wskaźnik jakości regulacji poprzez odjęcie stałej wartości równej wskaźnikowi jakości regulacji dla początkowych współczynników regulatora oraz nominalnego momentu bezwładności. Dzięki temu żaden parametr nie musiał być modyfikowany w stosunku do badań symulacyjnych. Zgodnie z badaniami symulacyjnymi algorytm optymalizacji lokalnej wykazał się krótszym czasem adaptacji. Niemniej oba algorytmy, tj. optymalizacja rojem cząstek i przeszukiwanie wzorcem, skutecznie zaadaptowały współczynnik regulatora ze sprzężeniem od wektora zmiennych stanu niezależnie od momentu bezwładności i warunków pracy napędu. Ze względu na zaproponowane okienkowanie funkcji celu efekt tzw. przeuczania nie występuje w zaproponowanej strukturze regulacji adaptacyjnej. Ponadto, aktualizacja nastaw regulatora z każdym okresem sygnału referencyjnego pozwala na wyeliminowanie problemów z dokładnością liczb zmiennoprzecinkowych.

Przedstawione w niniejszej rozprawie wyniki badań potwierdzają słuszność postawionej tezy, że uzyskanie odporności napędu elektrycznego z silnikiem PMSM na zmianę momentu bezwładności możliwe jest poprzez zastosowanie sterowania adaptacyjnego z modelem odniesienia bazującego na algorytmach optymalizacyjnych inspirowanych przyrodą.

Za najważniejsze osiągnięcia własne autora uważa się:

- opracowanie struktury regulacji adaptacyjnej bazującej na regule Widrow'a-Hoff'a dla napędu z silnikiem PMSM oraz uzyskanie bardzo dobrych własności procesu adaptacji dzięki wnikliwej analizie w jej działanie w układzie mikroprocesorowym,
- opracowanie PAAO umożliwiającej zastosowanie większości algorytmów optymalizacyjnych, w tym inspirowanych przyrodą, do adaptacji współczynników regulatora ze sprzężeniem od wektora zmiennych stanu,
- opracowanie modeli symulacyjnych badanych adaptacyjnych struktur regulacji dla napędu elektrycznego z silnikiem PMSM o zmiennym momencie bezwładności,
- uruchomienie stanowiska badawczego, jego integrację z urządzeniem mechatronicznym pozwalającym na zmianę momentu bezwładności oraz implementację opracowanych struktur regulacji adaptacyjnych,

— eksperymentalną weryfikację wybranych wyników badań symulacyjnych potwierdzającą poprawne działanie opracowanych struktur regulacji adaptacyjnych.

Autor uważa, że zaproponowana struktura regulacji automatycznej może znaleźć zastosowanie na liniach produkcyjnych z cyklicznie zmieniającymi się elementami, gdzie występujące przeregulowanie i zmiana czasu regulacji, na przykład przenośnika taśmowego, niekorzystnie wpływają na jakość procesu produkcji.

Bibliografia

- [1] S. Morimoto, “Trend of permanent magnet synchronous machines,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 2, no. 2, pp. 101–108, 2007.
- [2] R. Krishnan, *Electric motor drives: modeling, analysis, and control*, vol. 626. Prentice Hall New Jersey, 2001.
- [3] K. Pietruszewicz, “Multi-degree of freedom robust control of the cnc xy table pmsm-based feed-drive module,” *Archives of Electrical Engineering*, pp. 15–31, 2012.
- [4] W. Lu, Q. Li, K. Lu, Y. Lu, L. Guo, W. Yan, and F. Xu, “Load adaptive pmsm drive system based on an improved adrc for manipulator joint,” *IEEE Access*, vol. 9, pp. 33369–33384, 2021.
- [5] C. Mi and M. A. Masrur, *Hybrid electric vehicles: principles and applications with practical perspectives*. John Wiley & Sons, 2017.
- [6] K. Sun, Y. Lu, Y. Xing, and L. Huang, “Modeling and analysis of leakage currents in pwm-vsi-fed pmsm drives for air-conditioners with high accuracy and within a wide frequency range,” *Journal of Power Electronics*, vol. 16, no. 3, pp. 970–981, 2016.
- [7] M. Tursini, F. Parasiliti, and D. Zhang, “Real-time gain tuning of PI controllers for high-performance PMSM drives,” *IEEE Transactions on Industry Applications*, vol. 38, pp. 1018–1026, July 2002.
- [8] M. Preindl and S. Bolognani, “Model predictive direct speed control with finite control set of pmsm drive systems,” *IEEE Transactions on Power Electronics*, vol. 28, no. 2, pp. 1007–1015, 2012.
- [9] M. Safonov and M. Athans, “Gain and phase margin for multiloop lqg regulators,” *IEEE Transactions on Automatic Control*, vol. 22, no. 2, pp. 173–179, 1977.
- [10] T. Tarczewski and L. M. Grzesiak, “Constrained state feedback speed control of pmsm based on model predictive approach,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3867–3875, 2015.
- [11] K. J. Åström, “Theory and applications of adaptive control—a survey,” *Automatica*, vol. 19, no. 5, pp. 471–486, 1983.
- [12] M. Kaminski, “Grey wolf optimizer in design process of stable neural controller – theoretical background and experiment,” in *Advanced, Contemporary Control* (A. Bartoszewicz, J. Kabziński, and J. Kacprzyk, eds.), (Cham), pp. 701–712, Springer International Publishing, 2020.

- [13] W. Sun, Y. Zhang, Y. Huang, H. Gao, and O. Kaynak, “Transient-performance-guaranteed robust adaptive control and its application to precision motion control systems,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6510–6518, 2016.
- [14] G. Tarchała and T. Orłowska-Kowalska, “Discrete sliding mode speed control of induction motor using time-varying switching line,” *Electronics*, vol. 9, no. 1, p. 185, 2020.
- [15] J. Zhai and Z. Song, “Adaptive sliding mode trajectory tracking control for wheeled mobile robots,” *International Journal of Control*, vol. 92, no. 10, pp. 2255–2262, 2019.
- [16] E. Alcalá, V. Puig, J. Quevedo, and T. Escobet, “Gain-scheduling l_pv control for autonomous vehicles including friction force estimation and compensation mechanism,” *IET Control Theory & Applications*, vol. 12, no. 12, pp. 1683–1693, 2018.
- [17] M. Waszak and R. Łangowski, “An automatic self-tuning control system design for an inverted pendulum,” *IEEE Access*, vol. 8, pp. 26726–26738, 2020.
- [18] S. Kim and C. K. Ahn, “Offset-free proportional-type self-tuning speed controller for permanent magnet synchronous motors,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7168–7176, 2019.
- [19] Y. Shang, “Resilient consensus of switched multi-agent systems,” *Systems & Control Letters*, vol. 122, pp. 12–18, 2018.
- [20] Szczepanski, R and Tarczewski, T and Grzesiak, LM, “PMSM drive with adaptive state feedback speed controller,” *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 68, no. 5, pp. 1009–1017, 2020.
- [21] M. Kaminski and T. Orłowska-Kowalska, “Adaptive neural speed controllers applied for a drive system with an elastic mechanical coupling—a comparative study,” *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 152–167, 2015.
- [22] R. A. Fisher, *The genetical theory of natural selection: a complete variorum edition*. Oxford University Press, 1999.
- [23] M. Dorigo and G. Di Caro, “Ant colony optimization: a new meta-heuristic,” in *Proceedings of the IEEE Congress on Evolutionary Computation-CEC99*, vol. 2, pp. 1470–1477, 1999.
- [24] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995.
- [25] R. Szczepanski, K. Erwinski, and M. Paprocki, “Accelerating PSO based feedrate optimization for NURBS toolpaths using parallel computation with OpenMP,” in *Proceedings of the IEEE 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 431–436, Aug 2017.

- [26] W. Yu, B. Li, H. Jia, M. Zhang, and D. Wang, "Application of multi-objective genetic algorithm to optimize energy efficiency and thermal comfort in building design," *Energy and Buildings*, vol. 88, pp. 135–143, 2015.
- [27] M. Elhoseny, X. Yuan, Z. Yu, C. Mao, H. K. El-Minir, and A. M. Riad, "Balancing energy consumption in heterogeneous wireless sensor networks using genetic algorithm," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2194–2197, 2015.
- [28] S. Chatterjee, S. Sarkar, S. Hore, N. Dey, A. S. Ashour, and V. E. Balas, "Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings," *Neural Computing and Applications*, vol. 28, no. 8, pp. 2005–2016, 2017.
- [29] P. K. Sahu, T. Shah, K. Manna, and S. Chattopadhyay, "Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 300–312, 2014.
- [30] S. Bououden, M. Chadli, and H. R. Karimi, "An ant colony optimization-based fuzzy predictive control approach for nonlinear processes," *Information Sciences*, vol. 299, pp. 143–158, 2015.
- [31] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report, Erciyes University*, p. , 01 2005.
- [32] X.-S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.
- [33] X. Yang and S. Deb, "Cuckoo search via levy flights," in *Proceedings of the IEEE World Congress on Nature Biologically Inspired Computing (NaBIC)*, pp. 210–214, Dec 2009.
- [34] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Science*, vol. 5792, pp. 169–178, 2009.
- [35] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.
- [36] T. Tarczewski and L. M. Grzesiak, "An application of novel nature-inspired optimization algorithms to auto-tuning state feedback speed controller for PMSM," *IEEE Transactions on Industry Applications*, vol. 54, pp. 2913–2925, May 2018.
- [37] M. Zamani, M. Karimi-Ghartemani, N. Sadati, and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," *Control Engineering Practice*, vol. 17, no. 12, pp. 1380–1387, 2009.
- [38] H. Boubertakh, M. Tadjine, P. Glorennec, and S. Labiod, "Tuning fuzzy PID controllers using ant colony optimization," in *Proceedings of the IEEE 17th Mediterranean Conference on Control and Automation*, pp. 13–18, June 2009.

- [39] S. Debbarma, L. C. Saikia, and N. Sinha, "Automatic generation control using two degree of freedom fractional order PID controller," *International Journal of Electrical Power & Energy Systems*, vol. 58, pp. 120–129, 2014.
- [40] H. Mesloub, M. T. Benchouia, R. Boumaaraf, A. Goléa, N. Goléa, and M. Becherif, "Design and implementation of dtc based on aflc and pso of a pmsm," *Mathematics and Computers in Simulation*, vol. 167, pp. 340–355, 2020.
- [41] X. Sun, C. Hu, G. Lei, Y. Guo, and J. Zhu, "State feedback control for a pm hub motor based on gray wolf optimization algorithm," *IEEE Transactions on Power Electronics*, vol. 35, pp. 1136–1146, Jan 2020.
- [42] S. B. C. Debnath, P. C. Shill, and K. Murase, "Particle swarm optimization based adaptive strategy for tuning of fuzzy logic controller," *International Journal of Artificial Intelligence & Applications*, vol. 4, no. 1, p. 37, 2013.
- [43] S. Bououden, M. Chadli, F. Allouani, and S. Filali, "A new approach for fuzzy predictive adaptive controller design using particle swarm optimization algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 9, pp. 3741–3758, 2013.
- [44] M. Kamiński, "Zastosowanie algorytmu bat w optymalizacji obliczeń adaptacyjnego regulatora stanu układu dwumasowego," *Przegląd Elektrotechniczny*, vol. 93, no. 1, pp. 300–304, 2017.
- [45] X. Wang, B. Ufnalski, and L. M. Grzesiak, "Adaptive speed control in the PMSM drive for a non-stationary repetitive process using particle swarms," in *Proceedings of the IEEE 10th International Conference on Compatibility, Power Electronics and Power Engineering*, pp. 464–471, June 2016.
- [46] R. Szczepanski, T. Tarczewski, and L. M. Grzesiak, "Adaptive state feedback speed controller for pmsm based on artificial bee colony algorithm," *Applied Soft Computing*, vol. 83, p. 105644, 2019.
- [47] L. Grzesiak, A. Kaszewski, and B. Ufnalski, *Sterowanie napędów elektrycznych: Analiza, modelowanie, projektowanie*. Polskie Wydawnictwo Naukowe, 2016.
- [48] S. Vaez-Zadeh, *Control of permanent magnet synchronous motors*. Oxford University Press, 2018.
- [49] T. Tarczewski, L. J. Niewiara, and L. M. Grzesiak, "Gain-scheduled state feedback speed control of synchronous reluctance motor," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pp. 559–565, IEEE, 2021.
- [50] T. Tarczewski, Ł. J. Niewiara, M. Skiwski, and L. M. Grzesiak, "Gain-scheduled constrained state feedback control of dc–dc buck power converter," *IET Power Electronics*, vol. 11, no. 4, pp. 735–743, 2018.
- [51] I. Boldea, "Control issues in adjustable speed drives," *IEEE Industrial Electronics Magazine*, vol. 2, no. 3, pp. 32–50, 2008.

- [52] J. R. Dominguez, A. Navarrete, M. A. Meza, A. G. Loukianov, and J. Canedo, "Digital sliding-mode sensorless control for surface-mounted pmsm," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 137–151, 2013.
- [53] H. H. Choi, H. M. Yun, and Y. Kim, "Implementation of evolutionary fuzzy pid speed controller for pm synchronous motor," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 540–547, 2013.
- [54] T. Pajchrowski, K. Zawirski, and K. Nowopolski, "Neural speed controller trained online by means of modified rprop algorithm," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 560–568, 2014.
- [55] L. M. Grzesiak and T. Tarczewski, "Pmsm servo-drive control system with a state feedback and a load torque feedforward compensation," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, 2013.
- [56] T. Tarczewski, M. Skiwski, L. Niewiara, and L. Grzesiak, "High-performance pmsm servo-drive with constrained state feedback position controller," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 66, no. No 1, pp. 49–58, 2018.
- [57] T. Tarczewski and L. M. Grzesiak, "Constrained state feedback speed control of PMSM based on model predictive approach," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 3867–3875, June 2016.
- [58] R. Szczepanski, M. Kaminski, and T. Tarczewski, "Auto-tuning process of state feedback speed controller applied for two-mass system," *Energies*, vol. 13, p. 3067, 2020.
- [59] R. Szczepanski, T. Tarczewski, K. Erwinski, and L. M. Grzesiak, "Comparison of constraint-handling techniques used in artificial bee colony algorithm for auto-tuning of state feedback speed controller for pmsm.," in *ICINCO (1)*, pp. 279–286, 2018.
- [60] I. Semykina and A. Tarnetskaya, "Magnet synchronous machine of mine belt conveyor gearless drum-motor," in *E3S Web of Conferences*, vol. 41, p. 03014, EDP Sciences, 2018.
- [61] I. Semykina and A. Tarnetskaya, "The design of control system for gearless synchronous drum-motor of the underground mine belt conveyor," in *E3S Web of Conferences*, vol. 105, p. 03004, EDP Sciences, 2019.
- [62] J. Ju, W. Li, Y. Wang, M. Fan, and X. Yang, "Dynamics and nonlinear feedback control for torsional vibration bifurcation in main transmission system of scraper conveyor direct-driven by high-power pmsm," *Nonlinear Dynamics*, vol. 93, no. 2, pp. 307–321, 2018.
- [63] F. Khorrami, P. Krishnamurthy, and H. Melkote, *Modeling and adaptive nonlinear control of electric motors*. Springer Science & Business Media, 2003.

- [64] M. S. Zaky, E. M. Ismaeil, and M. M. Khater, "Gain scheduling adaptive proportional-integral controller for a field-oriented control of hybrid stepper motor drives," *Electric Power Components and Systems*, vol. 40, no. 7, pp. 777–791, 2012.
- [65] H. Hannoun, M. Hilairret, and C. Marchand, "High performance current control of a switched reluctance machine based on a gain-scheduling pi controller," *Control Engineering Practice*, vol. 19, no. 11, pp. 1377–1386, 2011.
- [66] L. Wang, Y. Zhang, S. Hao, M. Hao, and B. Song, "Adaptive gain scheduling with feedforward control system based on system model for pmsm," *International Journal of Electronics*, vol. 104, no. 10, pp. 1626–1645, 2017.
- [67] T. Tarczewski, L. Niewiara, and L. Grzesiak, "Artificial bee colony based state feedback position controller for pmsm servo-drive—the efficiency analysis," *Bulletin of the Polish Academy of Sciences. Technical Sciences*, vol. 68, no. 5, 2020.
- [68] K. M. Ravi Eswar, K. Venkata Praveen Kumar, and T. Vinay Kumar, "Enhanced predictive torque control with auto-tuning feature for induction motor drive," *Electric Power Components and Systems*, vol. 46, no. 7, pp. 825–836, 2018.
- [69] X. Liu, H. Yu, J. Yu, and L. Zhao, "Combined speed and current terminal sliding mode control with nonlinear disturbance observer for pmsm drive," *IEEE access*, vol. 6, pp. 29594–29601, 2018.
- [70] E. Kilic, H. R. Ozcalik, and S. Yilmaz, "Efficient speed control of induction motor using rbf based model reference adaptive control method," *automatika*, vol. 57, no. 3, pp. 714–723, 2016.
- [71] B. Widrow and M. E. Hoff, "Adaptive switching circuits," tech. rep., Stanford Univ Ca Stanford Electronics Labs, 1960.
- [72] T. Pajchrowski and K. Zawirski, "Application of artificial neural network for adaptive speed control of pmsm drive with variable parameters," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, 2013.
- [73] R. Szczepanski, T. Tarczewski, and L. Grzesiak, "Pmsm drive with adaptive state feedback speed controller," *Bulletin of the Polish Academy of Sciences. Technical Sciences*, vol. 68, no. 5, 2020.
- [74] T. Pajchrowski and K. Zawirski, "Adaptive neural speed controller for pmsm servodrive with variable parameters," in *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*, pp. LS6b–3, IEEE, 2012.
- [75] R. Tadeusiewicz and M. Szaleniec, *Leksykon sieci neuronowych*. Projekt Nauka. Fundacja na rzecz promocji nauki polskiej, 2015.
- [76] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

- [77] V. Granville, M. Krivánek, and J.-P. Rasson, "Simulated annealing: A proof of convergence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 652–656, 1994.
- [78] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on optimization*, vol. 7, no. 1, pp. 1–25, 1997.
- [79] A. Sahu, S. K. Panigrahi, and S. Pattnaik, "Fast convergence particle swarm optimization for functions optimization," *Procedia Technology*, vol. 4, pp. 319–324, 2012.
- [80] R. Szczepanski, T. Tarczewski, L. J. Niewiara, and D. Stojic, "Identification of mechanical parameters in servo-drive system," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pp. 566–573, IEEE, 2021.
- [81] R. Szczepanski, T. Tarczewski, and L. M. Grzesiak, "Application of optimization algorithms to adaptive motion control for repetitive process," *ISA transactions*, vol. 115, pp. 192–205, 2021.

Spis tabel

2.1. Parametry napędu oraz współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu	29
4.1. Parametry PAAO oraz algorytmów optymalizacyjnych	66
5.1. Parametry znamionowe badanego silnika	81
5.2. Otrzymany całkowity wskaźnik jakości regulacji (IAE) oraz współczynniki regulatora ze sprzężeniem od wektora zmiennych stanu dla wszystkich badanych struktur regulacji adaptacyjnej po każdym z etapów testu pierwszego oraz drugiego	96

Spis rysunków

2.1.	Schemat blokowy kaskadowej struktury regulacji napędu z silnikiem PMSM	23
2.2.	Schemat blokowy napędu z silnikiem PMSM i regulatorem ze sprzężeniem od wektora zmiennych stanu	25
2.3.	Schemat blokowy napędu z silnikiem PMSM bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu o zredukowanej liczbie współczynników	26
2.4.	Model symulacyjny napędu elektrycznego z silnikiem PMSM o zmiennym momencie bezwładności opracowany w programie MATLAB/SIMULINK	27
2.5.	Model symulacyjny przekształtnika energoelektronicznego opracowany w programie MATLAB/SIMULINK	28
2.6.	Model symulacyjny silnika PMSM o zmiennym momencie bezwładności opracowany w programie MATLAB/SIMULINK	28
2.7.	Wpływ zmiany momentu bezwładności na odpowiedź skokową napędu elektrycznego z silnikiem PMSM bazującego na regulatorze ze sprzężeniem od wektora zmiennych stanu	30
3.1.	Schemat blokowy maszyny klasyfikującej AdaLiNe	36
3.2.	Schemat blokowy sterowania adaptacyjnego z modelem odniesienia	37
3.3.	Wyrowadzenie transmitancji układu zamkniętego z regulatorem stanu dla prędkości kątowej napędu z silnikiem PMSM	39
3.4.	Regulator ze sprzężeniem od wektora zmiennych stanu odpowiedzialnego za wyznaczenie sygnału sterującego w osi q wirującego układu współrzędnych	41
3.5.	Schemat blokowy adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego na regule Widrow'a-Hoff'a	43
3.6.	Model symulacyjny adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na regule Widrow'a-Hoff'a opracowany w programie MATLAB/SIMULINK	44
3.7.	Model symulacyjny mechanizmu adaptacji współczynników k_{x5} (a), k_{x5} (b) oraz $k_{\omega 2}$ (c) regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na regule Widrow'a-Hoff'a opracowany w programie MATLAB/SIMULINK	46

3.8.	Wpływ współczynnika adaptacji na dynamikę mechanizmu adaptacji bazującego na regule Widrow'a-Hoff'a	47
3.9.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$	49
3.10.	Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowane współczynniki regulatora oraz wybrane przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ ukazujące reakcję układu na zmianę momentu bezwładności przy nieadekwatnym rzędzie modelu odniesienia. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$	50
4.1.	Wizualizacja okienkowanej funkcji celu w porównaniu do aktualizacji w każdym okresie próbkowania regulatora	54
4.2.	Schemat blokowy adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu dla napędu z silnikiem PMSM w trybie regulacji prędkości kątowej bazującego na procedurze adaptacji wykorzystującej algorytmy optymalizacyjne	55
4.3.	Schemat blokowy procedury adaptacji dla algorytmów optymalizacyjnych	57
4.4.	Wizualizacja procesu adaptacji bazującej na PAAO	59
4.5.	Przykładowa iteracja algorytmu przeszukiwania wzorcem dla dwuwymiarowego problemu optymalizacyjnego	62
4.6.	Model symulacyjny adaptacyjnego regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na procedurze adaptacji dla algorytmów optymalizacyjnych opracowany w programie MATLAB/SIMULINK	64
4.7.	Model symulacyjny mechanizmu adaptacji współczynników k_{x5} , k_{x5} oraz $k_{\omega 2}$ regulatora ze sprzężeniem od wektora zmiennych stanu bazującego na PAAO opracowany w programie MATLAB/SIMULINK	65
4.8.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego. Zaproponowany mechanizm adaptacji bazujący na algorytmie optymalizacyjnym rojem cząstek	67
4.9.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego. Zaproponowany mechanizm adaptacji bazujący na algorytmie optymalizacyjnym przeszukiwania wzorcem	68

4.10. Przebiegi całkowego wskaźnika jakości oraz współczynników regulatora ze sprzężeniem od wektora zmiennych stanu dla mechanizmu adaptacji bazującego na algorytmach optymalizacyjnych	69
4.11. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla okresów sygnału referencyjnego z najgorszymi całkowitymi wskaźnikami jakości regulacji w trakcie trwania procesu adaptacji.	70
4.12. Wpływ parametrów związanych z wykrywaniem zmian obiektu oraz tolerowanej różnicy pomiędzy modelem odniesienia a rzeczywistą odpowiedzią układu	71
4.13. Wpływ parametru odpowiedzialnego za rozmiar przeszukiwanej przestrzeni rozwiązań w okół aktualnego najlepszego rozwiązania.	72
4.14. Wpływ parametrów ψ_1 oraz ψ_2 na proces adaptacji.	75
4.15. Porównanie czasu adaptacji oraz końcowego dopasowania w 10 uruchomieniach algorytmu optymalizacji rojem cząstek i algorytmu przeszukiwania wzorcem	76
4.16. Przebiegi czasowe całkowego wskaźnika jakości regulacji i adaptowane współczynniki regulatora oraz wybrane przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ ukazujące reakcję układu na zmianę momentu bezwładności przy nieadekwatnym rzędzie modelu odniesienia. Mechanizm adaptacji bazujący na algorytmie optymalizacyjnym	78
5.1. Zdjęcie zespołu maszynowego stanowiska laboratoryjnego: a) badany silnik PMSM, b) krążek stalowy zwiększający moment bezwładności zespołu, c) urządzenie mechatroniczne o zmiennym momencie bezwładności, d) silnik służący jako obciążenie układu, e) sprzęgła	80
5.2. Zdjęcie urządzenia mechatronicznego dla różnego rozstawu ramion	81
5.3. Zdjęcie urządzenia mechatronicznego dla różnego rozstawu ramion	82
5.4. Zdjęcie stanowiska laboratoryjnego	84
5.5. Wpływ częstotliwości próbkowania na odpowiedź modelu referencyjnego generowanego w układzie mikroprocesorowym o pojedynczej precyzji liczb zmiennoprzecinkowych	85
5.6. Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zwiększonym momencie bezwładności. Mechanizm adaptacji bazujący na regule Widrw'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$	87

5.7.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zmniejszonym momencie bezwładności do nominalnej wartości. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$.	87
5.8.	Przebiegi czasowe całkowego wskaźnika jakości jakości regulacji i adaptowanych współczynników regulatora dla eksperymentu ze zwiększoną wartością momentu bezwładności oraz ze zmniejszoną do nominalnych wartości. Mechanizm adaptacji bazujący na regule Widrow'a-Hoff'a ze współczynnikiem uczenia wynoszącym $\mu = 0.05T_s$	88
5.9.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zwiększonym momencie bezwładności. Mechanizm adaptacji bazujący algorytmie optymalizacji rojem cząstek.	91
5.10.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zmniejszonym momencie bezwładności do nominalnej wartości. Mechanizm adaptacji bazujący algorytmie optymalizacji rojem cząstek.	92
5.11.	Przebiegi czasowe całkowego wskaźnika jakości jakości regulacji i adaptowanych współczynników regulatora dla eksperymentu ze zwiększoną wartością momentu bezwładności oraz ze zmniejszoną do nominalnych wartości. Mechanizm adaptacji bazujący algorytmie optymalizacji rojem cząstek.	93
5.12.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zwiększonym momencie bezwładności. Mechanizm adaptacji bazujący algorytmie optymalizacyjnym przeszukiwania wzorcem.	94
5.13.	Przebiegi czasowe prędkości kątowej, prądów i sygnałów sterujących w wirującym układzie współrzędnych $d-q$ dla kluczowych okresów sygnału referencyjnego eksperymentu przy zmniejszonym momencie bezwładności do nominalnej wartości. Mechanizm adaptacji bazujący algorytmie optymalizacyjnym przeszukiwania wzorcem.	94

5.14. Przebiegi czasowe całkowego wskaźnika jakości jakości regulacji i adaptowanych współczynników regulatora dla eksperymentu ze zwiększoną wartością momentu bezwładności oraz ze zmniejszoną do nominalnych wartości. Mechanizm adaptacji bazujący algorytmie optymalizacyjnym przeszukiwania wzorcem.	95
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

Załącznik 1: Kod źródłowy wybranych funkcji

```
1 // Funkcja obliczająca całkowity wskaźnik jakości regulacji
2 // wyzwalana w każdej pętli regulacji
3 void paa0_ObjectiveFunctionCalculationRoutine(
4     float y_ref, float y, float y_mras)
5 {
6     static float y_ref_prev = 0.0f;
7     static float temp_fval = 0.0f;
8     static uint32_t i = 0;
9     static float y_mean = 0.0f;
10
11     if( fval_isReady == true )
12         return;
13
14     if( y_ref > 1 && y_ref_prev < 1 )
15     {
16         fval = temp_fval;
17         temp_fval = 0.0f;
18         fval_isReady = true;
19         i = 0;
20         y_mean = 0.0f;
21     }
22     y_ref_prev = y_ref;
23
24     y_mean += y;
25     if( ++i >= 22 )
26     {
27         y_mean /= (float)i;
28         if( y_mean > y_mras )
29             temp_fval += (Ts*22) * (y_mean - y_mras);
30         else
31             temp_fval += (Ts*22) * (y_mras - y_mean);
32
```

```

33     i = 0;
34     y_mean = 0.0f;
35 }
36 }

1 // Funkcja inicjalizująca strukturę PAAO
2 void paa_initialize( void )
3 {
4     PAAO.TheBestSolutionExaminationPeriod = 30;
5     PAAO.Alpha = 0.8;
6     PAAO.Ch_th = 0.02;
7     PAAO.ChP_th = 10;
8     PAAO.conv_th = 0.001;
9     PAAO.step_max = 0.1;
10
11     PAAO.step = 0;
12     PAAO.prevFitness = 0;
13 }

1 // Funkcja wyzwalana po ewaluacji dotychczasowego najlepszego
2 // rozwiązania w celu kontroli procesu adaptacji
3 paa_nextStage_e paa_Routine(float actualFitness, float conv)
4 {
5     paa_nextStage_e nextStage = PAAO_FLAG_STOP_OPTIMIZATION;
6
7     PAAO.step *= PAAO.Alpha;
8
9     if( actualFitness < 1e-5 )
10         actualFitness = 1e-5;
11
12     if( PAAO.prevFitness == -1 )
13         PAAO.prevFitness = actualFitness;
14
15     float Ch = PAAO.prevFitness - actualFitness;
16     if( Ch < 0 ) Ch *= (-1.0f);
17
18     float ChP = 100 * Ch / PAAO.prevFitness;
19
20     if( ChP > PAAO.ChP_th && Ch > PAAO.Ch_th && actualFitness > PAAO.Ch_th )
21     {

```



```

22     PAAO.prevFitness = actualFitness;
23     if( conv < (PAAO.step_max / 2) )
24     {
25         PAAO.step = PAAO.step_max;
26         nextStage = PAAO_FLAG_REINITIALIZATION;
27     }
28     else
29     {
30         nextStage = PAAO_FLAG_CONTINUE_OPTIMIZATION;
31     }
32 }
33 else
34 {
35     if( conv <= 1e-10 )
36     {
37         nextStage = PAAO_FLAG_STOP_OPTIMIZATION;
38     }
39     else if( conv <= PAAO.conv_th || actualFitness <= PAAO.Ch_th )
40     {
41         PAAO.prevFitness = actualFitness;
42         nextStage = PAAO_FLAG_STOP_OPTIMIZATION;
43     }
44     else
45     {
46         nextStage = PAAO_FLAG_CONTINUE_OPTIMIZATION;
47     }
48 }
49
50 PAAO.lastStage = nextStage;
51 return nextStage;
52 }

1 // Mechanizm adaptacji bazujący na algorytmie przeszukiwania wzorcem.
2 // Proces uruchamiany w głównej pętli programu (najniższy priorytet)
3 void rs_AdaptationMechanism_PS(void)
4 {
5     paa_nextStage_e nextStage;
6
7     paa_initialize();

```

```

8
9  PS.Solution[0] = K_SFC[1][1];
10 PS.Solution[1] = K_SFC[1][2];
11 PS.Solution[2] = K_SFC[1][3];
12
13 PS.CurrentlyExaminedSolution[0] = PS.Solution[0];
14 PS.CurrentlyExaminedSolution[1] = PS.Solution[1];
15 PS.CurrentlyExaminedSolution[2] = PS.Solution[2];
16
17 while(1)
18 {
19 while(false == fval_isReady);
20
21 fval -= 0.06f;
22 if( fval < 0.0f ) fval = 0.0f;
23
24 if( fval < PAAO.Ch_th && PS.CurrentState != PS_STATE_PAAO)
25 {
26 PS.SolutionFitness = fval;
27 PS.Solution[0] = PS.CurrentlyExaminedSolution[0];
28 PS.Solution[1] = PS.CurrentlyExaminedSolution[1];
29 PS.Solution[2] = PS.CurrentlyExaminedSolution[2];
30 PS.CurrentState = PS_STATE_PAAO;
31 }
32
33 switch( PS.CurrentState )
34 {
35 case PS_STATE_INITIALIZATION:
36 PS.Optimization_State = 0;
37 PS.Optimization_Counter = 0;
38 PS_Initialize(fval);
39 break;
40
41 case PS_STATE_OPTIMIZATION:
42 PS_OptimizationStep(fval);
43 break;
44
45 case PS_STATE_PAAO:
46 nextStage = paao_Routine(fval, PS.ActualStepSize);

```

```

47
48     switch( nextStage )
49     {
50         case PAAO_FLAG_STOP_OPTIMIZATION:
51             PS.ActualStepSize = 0;
52             PS_BeginTheBestSolutionExamination();
53             break;
54
55         case PAAO_FLAG_REINITIALIZATION:
56             PS_BeginReinitialization(fval);
57             break;
58
59         case PAAO_FLAG_CONTINUE_OPTIMIZATION:
60             PS_BeginOptimization(false);
61             break;
62     }
63     break;
64 }
65
66 K_SFC[1][1] = PS.CurrentlyExaminedSolution[0];
67 K_SFC[1][2] = PS.CurrentlyExaminedSolution[1];
68 K_SFC[1][3] = PS.CurrentlyExaminedSolution[2];
69
70 fval_isReady = false;
71 }
72 }

1 // Funkcja inicjalizująca algorytm przeszukiwania wzorcem
2 void PS_Initialize(float fitness)
3 {
4     PS.SolutionFitness = fitness;
5     PS.ActualStepSize = 0;
6
7     PS_BeginTheBestSolutionExamination();
8 }

1 // Funkcja rozpoczynająca kolejną iterację algorytmu przeszukiwania wzorcem
2 void PS_BeginOptimization(bool solutionImproved)
3 {
4     PS.CurrentState = PS_STATE_OPTIMIZATION;

```

```

5
6  if( PS.ActualStepSize > PAA0.step )
7      PS.ActualStepSize = PAA0.step;
8
9  float prevSolutionChange[3] = {
10     PS.SolutionChange[PS.Optimization_State][0],
11     PS.SolutionChange[PS.Optimization_State][1],
12     PS.SolutionChange[PS.Optimization_State][2]
13 };
14
15 PS.SolutionChange[0][0] = PS.ActualStepSize*PS.Solution[0];
16 PS.SolutionChange[0][1] = 0.0f;
17 PS.SolutionChange[0][2] = 0.0f;
18 PS.SolutionChange[1][0] = -PS.ActualStepSize*PS.Solution[0];
19 PS.SolutionChange[1][1] = 0.0f;
20 PS.SolutionChange[1][2] = 0.0f;
21 PS.SolutionChange[2][0] = 0.0f;
22 PS.SolutionChange[2][1] = PS.ActualStepSize*PS.Solution[1];
23 PS.SolutionChange[2][2] = 0.0f;
24 PS.SolutionChange[3][0] = 0.0f;
25 PS.SolutionChange[3][1] = -PS.ActualStepSize*PS.Solution[1];
26 PS.SolutionChange[3][2] = 0.0f;
27 PS.SolutionChange[4][0] = 0.0f;
28 PS.SolutionChange[4][1] = 0.0f;
29 PS.SolutionChange[4][2] = PS.ActualStepSize*PS.Solution[2];
30 PS.SolutionChange[5][0] = 0.0f;
31 PS.SolutionChange[5][1] = 0.0f;
32 PS.SolutionChange[5][2] = -PS.ActualStepSize*PS.Solution[2];
33
34 PS_RandomPermutationOfSolutionChange();
35
36 if( true == solutionImproved )
37     if( prevSolutionChange[0] != 0.0f || prevSolutionChange[1] != 0.0f
38         || prevSolutionChange[2] != 0.0f )
39     {
40         for(uint32_t i = 0; i < 6; i++)
41             for(uint32_t j = 0; j < 3; j++)
42                 if( (prevSolutionChange[j] > 0 && PS.SolutionChange[i][j] > 0)
43                     || (prevSolutionChange[j] < 0 && PS.SolutionChange[i][j] < 0) )

```

```

44     PS_SwapSolutionChangeRows(i, 0);
45     for(uint32_t i = 0; i < 6; i++)
46         for(uint32_t j = 0; j < 3; j++)
47             if( (prevSolutionChange[j] > 0 && PS.SolutionChange[i][j] < 0)
48                 || (prevSolutionChange[j] < 0 && PS.SolutionChange[i][j] > 0) )
49                 PS_SwapSolutionChangeRows(i, 5);
50     }
51
52     PS.Optimization_State = 0;
53     PS.CurrentlyExaminedSolution[0] = PS.Solution[0]
54     + PS.SolutionChange[PS.Optimization_State][0];
55     PS.CurrentlyExaminedSolution[1] = PS.Solution[1]
56     + PS.SolutionChange[PS.Optimization_State][1];
57     PS.CurrentlyExaminedSolution[2] = PS.Solution[2]
58     + PS.SolutionChange[PS.Optimization_State][2];
59 }

1 // Funkcja przygotowująca ewaluacja najlepszego rozwiązania
2 void PS_BeginTheBestSolutionExamination(void)
3 {
4     PS.CurrentState = PS_STATE_PAAO;
5     PS.Optimization_Counter = 0;
6     PS.CurrentlyExaminedSolution[0] = PS.Solution[0];
7     PS.CurrentlyExaminedSolution[1] = PS.Solution[1];
8     PS.CurrentlyExaminedSolution[2] = PS.Solution[2];
9 }

1 // Funkcja wykonująca pojedynczy krok algorytmu przeszukiwania wzorcem
2 void PS_OptimizationStep(float fitness)
3 {
4     PS.Optimization_Counter++;
5
6     if( fitness < PS.SolutionFitness )
7     {
8         PS.SolutionFitness = fitness;
9         PS.Solution[0] = PS.CurrentlyExaminedSolution[0];
10        PS.Solution[1] = PS.CurrentlyExaminedSolution[1];
11        PS.Solution[2] = PS.CurrentlyExaminedSolution[2];
12        if( PS.Optimization_Counter > PAAO.TheBestSolutionExaminationPeriod )
13        {

```

```

14     PS_BeginTheBestSolutionExamination();
15     return;
16 }
17 else
18 {
19     PS_BeginOptimization(true);
20     return;
21 }
22 }
23
24 PS.Optimization_State++;
25 if( PS.Optimization_State >= 6 )
26 {
27     PS.ActualStepSize = PS.ActualStepSize / 2.0f;
28     if( PS.Optimization_Counter > PAAO.TheBestSolutionExaminationPeriod )
29     {
30         PS_BeginTheBestSolutionExamination();
31         return;
32     }
33     else
34     {
35         PS_BeginOptimization(false);
36         return;
37     }
38 }
39 else
40 {
41     PS.CurrentlyExaminedSolution[0] = PS.Solution[0]
42     + PS.SolutionChange[PS.Optimization_State][0];
43     PS.CurrentlyExaminedSolution[1] = PS.Solution[1]
44     + PS.SolutionChange[PS.Optimization_State][1];
45     PS.CurrentlyExaminedSolution[2] = PS.Solution[2]
46     + PS.SolutionChange[PS.Optimization_State][2];
47 }
48 }

1 // Mechanizm adaptacji bazujący na algorytmie optymalizacji rojem cząstek.
2 // Proces uruchamiany w głównej pętli programu (najniższy priorytet)
3 void rs_AdaptationMechanism_PSO(void)

```

```

4 {
5   paa0_nextStage_e nextStage;
6
7   paa0_Initialize();
8
9   PSO.gbest[0] = K_SFC[1][1];
10  PSO.gbest[1] = K_SFC[1][2];
11  PSO.gbest[2] = K_SFC[1][3];
12
13  PSO.CurrentlyExaminedSolution[0] = PSO.gbest[0];
14  PSO.CurrentlyExaminedSolution[1] = PSO.gbest[1];
15  PSO.CurrentlyExaminedSolution[2] = PSO.gbest[2];
16
17  while(1)
18  {
19    while(false == fval_isReady);
20
21    fval -= 0.06f;
22    if( fval < 0.0f ) fval = 0.0f;
23
24    if( fval < PAA0.Ch_th && PSO.CurrentState != PSO_STATE_PAA0 )
25    {
26      PSO.fgbest = fval;
27      for(int j = 0; j < PSO_DIMENSIONS; j++)
28      {
29        PSO.gbest[j] = PSO.CurrentlyExaminedSolution[j];
30      }
31      PSO.CurrentState = PSO_STATE_PAA0;
32    }
33
34    switch( PSO.CurrentState )
35    {
36      case PSO_STATE_INITIALIZATION:
37        PSO_InitializationStep(fval);
38        break;
39
40      case PSO_STATE_OPTIMIZATION:
41        PSO_OptimizationStep(fval);
42        break;

```

```

43
44     case PSO_STATE_PAAO:
45         nextStage = paaoroutine(fval, PSO_GetConvergence());
46         switch( nextStage )
47         {
48             case PAAO_FLAG_STOP_OPTIMIZATION:
49                 for(int i = 0; i < PSO_POPULATION; i++)
50                 {
51                     for(int j = 0; j < PSO_DIMENSIONS; j++ )
52                     {
53                         PSO.Particles[i][j] = PSO.gbest[j];
54                         PSO.Velocities[i][j] = 0.0f;
55                         PSO.pbest[i][j] = PSO.Particles[i][j];
56                     }
57
58                     PSO.fpbest[i] = PSO.fgbest;
59                     PSO.ParticlesFunctionValues[i] = PSO.fgbest;
60                 }
61                 PSO_BeginTheBestSolutionExamination();
62                 break;
63
64             case PAAO_FLAG_REINITIALIZATION:
65                 PSO_BeginReinitialization();
66                 break;
67
68             case PAAO_FLAG_CONTINUE_OPTIMIZATION:
69                 PSO_BeginOptimization();
70                 break;
71         }
72         break;
73     }
74
75     K_SFC[1][1] = PSO.CurrentlyExaminedSolution[0];
76     K_SFC[1][2] = PSO.CurrentlyExaminedSolution[1];
77     K_SFC[1][3] = PSO.CurrentlyExaminedSolution[2];
78
79     fval_isReady = false;
80 }
81 }

```



```

1 // Funkcja rozpoczynająca ponowną inicjalizację algorytmu optymalizacji
2 // rojem cząstek
3 void PSO_BeginReinitialization(void)
4 {
5     for(int i = 0; i < PSO_POPULATION; i++)
6     {
7         for( int j = 0; j < PSO_DIMENSIONS; j++)
8         {
9             PSO.Velocities[i][j] = PSO.Velocities[i][j] + PSO_AddCraziness(j);
10        }
11
12        PSO_BoxConstrainedVelocities(i);
13
14        for( int j = 0; j < PSO_DIMENSIONS; j++)
15        {
16            PSO.Particles[i][j] = PSO.Particles[i][j] + PSO_AddCraziness(j);
17        }
18
19        PSO_BoxConstrainedPositions(i);
20    }
21
22    for(int i = 0; i < PSO_POPULATION; i++)
23    {
24        for( int j = 0; j < PSO_DIMENSIONS; j++)
25        {
26            PSO.pbest[i][j] = PSO.Particles[i][j];
27        }
28        PSO.fpbest[i] = 1e100; // inf
29    }
30
31    PSO.fgbest = 1e100; // inf
32
33    PSO.InitializePhase_Counter = 0;
34    PSO.InitializePhase_State = PSO_INITIALIZATION_STATE_EVALUATION;
35    PSO.CurrentState = PSO_STATE_INITIALIZATION;
36    for( int j = 0; j < PSO_DIMENSIONS; j++)
37    {
38        PSO.CurrentlyExaminedSolution[j] =
39            PSO.Particles[PSO.InitializePhase_Counter][j];

```

```

40 }
41 }

1 // Funkcja rozpoczynająca optymalizację algorytmem optymalizacji rojem
2 // cząstek
3 void PSO_BeginOptimization(void)
4 {
5     PSO.CurrentState = PSO_STATE_OPTIMIZATION;
6     PSO.OptimizationPhase_State = PSO_OPTIMIZATION_STATE_PREPARATION;
7     PSO_OptimizationStep(0);
8 }

1 // Funkcja przygotowująca ewaluacja najlepszego rozwiązania
2 void PSO_BeginTheBestSolutionExamination(void)
3 {
4     PSO.PAAO_Counter = 0;
5     PSO.CurrentState = PSO_STATE_PAAO;
6     for( int j = 0; j < PSO_DIMENSIONS; j++)
7     {
8         PSO.CurrentlyExaminedSolution[j] = PSO.gbest[j];
9     }
10 }

1 // Funkcja wykonująca jeden krok podczas inicjalizacji algorytmu
2 // optymalizacji rojem cząstek
3 void PSO_InitializationStep(float fitness)
4 {
5     uint32_t i;
6     switch(PSO.InitializePhase_State)
7     {
8     case PSO_INITIALIZATION_STATE_PREPARATION:
9         for(int i = 0; i < PSO_POPULATION; i++)
10            {
11                for(int j = 0; j < PSO_DIMENSIONS; j++ )
12                    {
13                        PSO.Particles[i][j] = PSO.gbest[j];
14                        PSO.Velocities[i][j] = 0.0f;
15                        PSO.pbest[i][j] = PSO.Particles[i][j];
16                    }
17

```

```

18     PSO.fpbest[i] = 1e100; // inf;
19     PSO.ParticlesFunctionValues[i] = 1e100; // inf
20     PSO.fgbest = 1e100; // inf
21 }
22 PSO_BeginTheBestSolutionExamination();
23 break;
24
25 case PSO_INITIALIZATION_STATE_EVALUATION:
26     i = PSO.InitializePhase_Counter;
27     PSO.ParticlesFunctionValues[i] = fitness;
28     PSO.fpbest[i] = fitness;
29     PSO.PAAO_Counter++;
30     if( i == (PSO_POPULATION-1u) )
31     {
32         if (PSO.PAAO_Counter >= PAAO.TheBestSolutionExaminationPeriod)
33         {
34             PSO_BeginTheBestSolutionExamination();
35         }
36         else
37         {
38             PSO_BeginOptimization();
39         }
40     }
41     else
42     {
43         PSO.InitializePhase_Counter++;
44         for( int j = 0; j < PSO_DIMENSIONS; j++)
45         {
46             PSO.CurrentlyExaminedSolution[j] =
47                 PSO.Particles[PSO.InitializePhase_Counter][j];
48         }
49     }
50 }
51 }

1 // Funkcja wykonująca jeden krok podczas optymalizacji
2 // algorytmu optymalizacji rojem cząstek
3 void PSO_OptimizationStep(float fitness)
4 {

```

```

5  uint32_t i;
6
7  switch( PSO.OptimizationPhase_State )
8  {
9      case PSO_OPTIMIZATION_STATE_PREPARATION:
10         PSO.OptimizationPhase_Counter = 0;
11         PSO_UpdateParticle( PSO.OptimizationPhase_Counter );
12         for( int j = 0; j < PSO_DIMENSIONS; j++)
13         {
14             PSO.CurrentlyExaminedSolution[j] =
15                 PSO.Particles[PSO.OptimizationPhase_Counter][j];
16         }
17         PSO.OptimizationPhase_State = PSO_OPTIMIZATION_STATE_EVALUATION;
18         break;
19
20     case PSO_OPTIMIZATION_STATE_EVALUATION:
21         i = PSO.OptimizationPhase_Counter;
22
23         PSO.ParticlesFunctionValues[i] = fitness;
24         if( PSO.fpbest[i] > fitness )
25         {
26             PSO.fpbest[i] = fitness;
27             for( int j = 0; j < PSO_DIMENSIONS; j++)
28             {
29                 PSO.pbest[i][j] = PSO.Particles[i][j];
30             }
31         }
32         if( PSO.fgbest > fitness )
33         {
34             PSO.fgbest = fitness;
35             for( int j = 0; j < PSO_DIMENSIONS; j++)
36             {
37                 PSO.gbest[j] = PSO.Particles[i][j];
38             }
39         }
40
41         PSO.PAAO_Counter++;
42         if( i == (PSO_POPULATION-1u) )
43         {

```

```

44     if( PSO.PAAO_Counter >= PAAO.TheBestSolutionExaminationPeriod )
45     {
46         PSO.OptimizationPhase_State = PSO_OPTIMIZATION_STATE_PREPARATION;
47         PSO_BeginTheBestSolutionExamination();
48     }
49     else
50     {
51         PSO_BeginOptimization();
52     }
53 }
54 else
55 {
56     PSO.OptimizationPhase_Counter++;
57     PSO_UpdateParticle( PSO.OptimizationPhase_Counter );
58     for( int j = 0; j < PSO_DIMENSIONS; j++)
59     {
60         PSO.CurrentlyExaminedSolution[j] =
61             PSO.Particles[PSO.OptimizationPhase_Counter][j];
62     }
63 }
64 break;
65 }
66 }

```

```

1 // Funkcja aktualizująca i-tą cząstkę w algorytmie optymalizacji
2 // rojem cząstek
3 void PSO_UpdateParticle(uint32_t i )
4 {
5     for( int j = 0; j < PSO_DIMENSIONS; j++)
6     {
7         PSO.Velocities[i][j] = PSO_OMEGA * PSO.Velocities[i][j]
8             + PSO_C1 * frand() * (PSO.pbest[i][j] - PSO.Particles[i][j])
9             + PSO_C2 * frand() * (PSO.gbest[j] - PSO.Particles[i][j]);
10    }
11
12    PSO_BoxConstrainedVelocities(i);
13
14    for( int j = 0; j < PSO_DIMENSIONS; j++)
15    {

```

```

16     PSO.Particles[i][j] += PSO.Velocities[i][j];
17 }
18
19 PSO_BoxContrainedPositions(i);
20 }

1 // Funkcja ograniczająca wartości prędkości i-tej cząstki
2 // w algorytmie optymalizacji rojem cząstek
3 void PSO_BoxContrainedVelocities(uint32_t i)
4 {
5     for( int j = 0; j < PSO_DIMENSIONS; j++)
6     {
7         PSO.Velocities[i][j] = MAX(PSO.Velocities[i][j],
8             -PAAO.step * PSO.gbest[j]);
9         PSO.Velocities[i][j] = MIN(PSO.Velocities[i][j],
10            PAAO.step * PSO.gbest[j]);
11     }
12 }

1 // Funkcja ograniczająca wartości pozycji i-tej cząstki
2 // w algorytmie optymalizacji rojem cząstek
3 void PSO_BoxContrainedPositions(uint32_t i)
4 {
5     for( int j = 0; j < PSO_DIMENSIONS; j++)
6     {
7         PSO.Particles[i][j] = MAX(PSO.Particles[i][j],
8             PSO.gbest[j] - PAAO.step * PSO.gbest[j]);
9         PSO.Particles[i][j] = MIN(PSO.Particles[i][j],
10            PSO.gbest[j] + PAAO.step * PSO.gbest[j]);
11     }
12 }

1 // Funkcja generująca wartość prędkości cząstki w j-tym wymiarze przy
2 // ponownej inicjalizacji algorytmu optymalizacji rojem cząstek
3 float PSO_AddCraziness(uint32_t j)
4 {
5     return (PAAO.step * PSO.gbest[j]) * frand();
6 }

```