

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

Warszawa, 27 listopada 2018 r.

D z i e k a n a t

Uprzejmie informuję, że na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej odbędzie się w dniu 11 grudnia 2018 r. publiczna obrona rozprawy doktorskiej

Mgr. inż. Patryka Chabra

temat: „Automatyczna generacja kodu algorytmów regulacji predykcyjnej na mikrokontrolery”

promotor – dr hab. inż. Maciej Ławryńczuk, prof. Politechniki Warszawskiej

recenzenci:

prof. dr. hab. inż. Roman Śmierzchalski z Politechniki Gdańskiej Wydział Elektrotechniki i Automatyki,

dr. hab. inż. Dariusz Bismor, prof. Politechniki Śląskiej Wydział Automatyki, Elektroniki i Informatyki.

Obrona odbędzie się w dniu 11 grudnia 2018 r. w sali 116 na Wydziale Elektroniki i Technik Informacyjnych – Gmach im. Janusza Groszkowskiego, Warszawa, ul. Nowowiejska 15/19; początek godz. 10.00

Po adresem: www.elka.pw.edu.pl/Wydzial/Rada-Wydzialu/Harmonogram-obron-doktorskich-streszczenia-i-recenzje zapewniony jest na stronie Wydziału dostęp do tekstów streszczenia rozprawy i recenzji, jak również do tekstu rozprawy umieszczonej w Bazie Wiedzy Politechniki Warszawskiej.

Dziekan



prof. dr hab. inż. Krzysztof Zaremba

Doktorant: mgr inż. Patryk Chaber

Promotor: dr hab. inż. Maciej Ławryńczuk, prof. PW

Tytuł rozprawy: **Automatyczna generacja kodu algorytmów regulacji predykcyjnej na mikrokontrolery**

Streszczenie

Przedmiotem niniejszej rozprawy jest projekt oraz badania autorskiego systemu do automatycznej generacji kodu algorytmów regulacji predykcyjnej AutoMATiC, z przeznaczeniem na jednostki o istotnie ograniczonych zasobach oraz stosunkowo niskiej mocy obliczeniowej – w szczególności mikrokomputery jednoukładowe. Obecnie praktycznie nie istnieją rozwiązania, które pozwalają na automatyczną i szybką generację algorytmów regulacji predykcyjnej, na wybraną platformę sprzętową, jako gotowego programu. Ewentualnie istniejące rozwiązania obciążone są poważnymi ograniczeniami, tj. nie spełniają oczekiwań w zakresie zajętości pamięci programu wynikowego, czasu jego wykonania lub oferowanej funkcjonalności.

Praca podzielona jest na sześć rozdziałów. Pierwszy rozdział stanowi wprowadzenie do rozprawy, określając jej dziedzinę oraz problem, którego dotyczy teza pracy. W drugim rozdziale omówione są podstawy teorii regulacji, w tym algorytmy regulacji predykcyjnej wykorzystane w systemie AutoMATiC. Trzeci rozdział zawiera opis funkcjonalny oraz implementacyjny proponowanego rozwiązania – poddane dyskusji są założenia projektu systemu oraz proponowany proces projektowania regulatora z wykorzystaniem niniejszego systemu. W rozdziale czwartym opisane zostały wyniki badań przeprowadzonych z wykorzystaniem proponowanego systemu, w szczególności zużycie zasobów, czas wykonania programu oraz możliwość implementacji regulatorów o zwiększonej niezawodności. Piąty rozdział poświęcony został porównaniu proponowanego rozwiązania z obecnie rozwijanymi narzędziami (m.in. MATLAB Coder, ACADO, Multi-Parametric Toolbox 3) pod kątem wydajności oraz funkcjonalności. Szósty rozdział stanowi podsumowanie rozprawy.

Na podstawie realizowanych prac stwierdzono, że autorskie rozwiązanie AutoMATiC stanowi wartościowe uzupełnienie istniejących rozwiązań. Umożliwia ono automatyczną generację kodu wydajnych algorytmów regulacji predykcyjnej, pozwalając tym samym na implementację wynikowego programu na mikrokomputerach jednoukładowych. Generacja przeprowadzana z wykorzystaniem mechanizmu transkompilacji pozwala na wprowadzenie zarówno modułowości jak i ułatwienie dalszego rozwoju narzędzia, zachowując prostotę zapisu wynikającą z użycia w tym celu języka MATLAB. System oferuje dwa algorytmy regulacji predykcyjnej: DMC (bazujący na modelu odpowiedzi skokowej) i GPC (bazujący na równaniu różnicowym), w dwóch wersjach: analitycznej i numerycznej (z cykliczną optymalizacją kwadratową). Wydajność pamięciowa kodu generowanego przez system AutoMATiC jest porównywalna z innymi rozwiązaniami, a efektywna implementacja algorytmów regulacji sprawia, że czas wykonania programu wynikowego jest niższy niż gdyby stosować rozwiązania konkurencyjne i wyrażony jest w milisekundach (algorytmy numeryczne) lub nawet setkach mikrosekund (algorytmy analityczne).

Słowa kluczowe: automatyczna generacja kodu, transkompilacja, regulacja predykcyjna, mikrokontroler.



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTROTECHNIKI
I AUTOMATYKI

Prof. dr hab. inż. Roman Śmierzchalski
tel. (+58) 348 63 27
e-mail: roman.smierzchalski@pg.edu.pl

Gdańsk, 2 listopada 2018r.

**RECENZJA ROZPRAWY DOKTORSKIEJ DLA RADY
WYDZIAŁU ELEKTRONIKI I TECHNIK INFORMACYJNYCH
POLITECHNIKI WARSZAWSKIEJ**

Tytuł rozprawy: **Automatyczna generacja kodu algorytmów regulacji predykcyjnej na mikrokontrolery.**

Autor rozprawy: **mgr inż. Patryk Chaber**

1. Jakie zagadnienie naukowe jest rozpatrzone w pracy /teza rozprawy/ i czy zostało ono dostatecznie jasno sformułowane przez autora? Jaki charakter ma rozprawa (teoretyczny, doświadczalny, inny)?

Rozwój technik komputerowych spowodował w okresie ostatnich lat powstanie nowych metod obliczeniowych i programów, w tym również w zakresie sterowania predykcyjnego. Sterowanie predykcyjne znajduje coraz większe zastosowanie w rozwiązaniach i aplikacjach przemysłowych. Obecnie istnieje trend tworzenia aplikacji pozwalających na generowanie programów w sposób automatyczny. Występuje szereg modułów pozwalających generowanie programów sterowania. Jednak nie istnieją rozwiązania, które pozwalają na generację algorytmów regulacji predykcyjnej, ze wskazaniem platformy sprzętowej. Ponadto istniejące rozwiązania nie są zoptymalizowane pod względem zajętości pamięci i czasu obliczeń.

Rozprawa doktorska wypełnia lukę w tym zakresie i obejmuje zaprojektowanie oraz badanie systemu automatycznie generującego kod algorytmów sterowania predykcyjnego o nazwie „AutoMATiC”. Istotną zaletą systemu to generowanie kodu na jednostki o ograniczonych zasobach oraz niskiej mocy obliczeniowej, głównie z przeznaczeniem na mikrokomputery jednoukładowe. W systemie zaimplementowano dwa algorytmy regulacji predykcyjnej: Dynamic Matrix Control DMC bazujący na modelu odpowiedzi skokowej i Generalized Predictive Control GPC - oparty na równaniu różnicowym, w dwóch wersjach: analitycznej i numerycznej. Do generacji został wykorzystany mechanizm transkompilacji, co pozwala na elastyczne modyfikacje systemu, wprowadzenie nowych modułów i rozbudowę narzędzia. Podstawą tworzenia kodu jest przygotowanie skryptu w języku Matlab.

Tematyka pracy dotyczy algorytmów sterowania predykcyjnego i projektowania systemów generujących automatycznie oprogramowanie i swoim zakresem obejmuje dyscyplinę naukową automatyka i robotyka.

W pracy postawiono tezę: „Autorskie narzędzie do automatycznej generacji kodu (AutoMATiC) umożliwia efektywne opracowanie implementacji programowej algorytmów regulacji predykcyjnej o dobrej jakości oraz niewielkim zapotrzebowaniu na pamięć i czas jednostki obliczeniowej, pozwalając na szybką implementację tych algorytmów na mikrokomputerach jednoukładowych”. W celu udowodnienia tezy Autor konstruuje system do generacji kodu (AutoMATiC) i przeprowadza eksperymenty rozwiązania przykładowych zadań sterowania predykcyjnego w różnych wersjach wykorzystując to narzędzie. Zaprojektowany i opracowany przez Autora system jest propozycją w stosunku do istniejących rozwiązań, szczególnie gdy wymagane jest zachowanie warunków zoptymalizowania czasu obliczeń a także ograniczeń wynikających z zastosowania określonych rozwiązań sprzętowych. Uwzględniając powyższe praca ma charakter przede

wszystkim projektowy, od strony teoretycznej przedstawiono wybrane metody MPC, eksperymentalna część potwierdza efektywność projektu. Zdaniem recenzenta główny nacisk Autor skierował na opracowanie określonego narzędzia oraz implementację algorytmów regulacji predykcyjnej przy wykorzystaniu mikrokomputerów jednoukładowych. Wykazał się bardzo dużą umiejętnością w zakresie projektowania systemów generacji kodów. Biorąc pod uwagę zakres pracy od strony programistycznej Autor opracował samodzielnie narzędzie, które zasadniczo powinno być wykonywane przez zespół. Świadczy to o dużym zaangażowaniu w przedstawiony system o znacznej złożoności obliczeniowej.

2. Czy w rozprawie przeprowadzono w sposób właściwy analizę źródeł / w tym literatury światowej, stanu wiedzy i zastosowań w przemyśle /świadczący o dostatecznej wiedzy autora. Czy wnioski z przeglądu źródeł sformułowano w sposób jasny i przekonujący?

Autor dokonał przeglądu źródeł, 89 pozycji w tym 81 publikacji obcojęzycznych pod kątem stosowanych w rozprawie algorytmów sterowania predykcyjnego jak i projektowania systemu. Najstarsza publikacja to H.S.Black, Stabilized feedback amplifiers. Bell System Technical Journal z 1934 roku, większość pozycji obejmuje prace po roku 2000. Autor w części teoretycznej na podstawie literatury definiuje podstawowe pojęcia regulacji predykcyjnej, zasady regulacji predykcyjnej i zastosowanych algorytmów DMC oraz GPC reprezentujących rozwiązywanie zadania dla obiektów liniowych. Przedstawiono również inne typy algorytmów oraz zalety regulacji predykcyjnej. Autor znalazł w przywołanych w rozprawie źródłach metody, które jego zdaniem powinny być zastosowane do tworzenia kodu programu regulacji predykcyjnej. Część pracy dotycząca przykładowych eksperymentów odnosi się do rozwiązań platformy testowej, gdzie szczegółowo opisano rozwiązania, w tym stosowany mikrokontroler STM32. Wnioski sformułowane przez Autora na podstawie przeglądu literatury są jednoznaczne, implementacja algorytmów obliczeniowych o dużej złożoności, szczególnie do sterowania predykcyjnego procesami dynamicznymi wymaga aplikacji o znacznie krótszych okresach próbkowania i zastosowania systemów wbudowanych, całość rozprawy świadczy o wiedzy Autora w zakresie sterowania predykcyjnego i projektowania systemów sterowania co obejmuje dyscyplinę naukową automatyka i robotyka. Ponadto Autor cytował w rozprawie 8 publikacji, których był współautorem.

3. Czy autor rozwiązał postawione zagadnienia, czy użył właściwej do tego metody i czy przyjęte założenia są uzasadnione?

Zagadnieniem, jakie Autor postanowił rozwiązać było stworzenie efektywnego systemu do implementacji zaawansowanych algorytmów regulacji predykcyjnej, zoptymalizowanych pod względem czasu obliczeń, zajętości pamięci programu wykonywalnego. System zapewni automatyczne i szybkie prototypowanie algorytmów sterowania predykcyjnego, poprzez generowanie programu do sterowania określonym procesem na wybraną platformę sprzętową w tym wypadku z zastosowaniem do celów sterowania mikrokomputerów jednoukładowych. Efektywność systemu charakteryzuje minimalny wkład projektanta w opracowanie implementacji algorytmu regulacji predykcyjnej poprzez wprowadzenie jedynie parametrów algorytmu regulacji.

Recenzent uważa, że opracowany system na podstawie przedstawionych eksperymentów, mimo pewnych ograniczeń wynikających z zastosowania dwóch algorytmów Dynamie Matrix Control (DMC) i Generalized Predictive Control (GPC) dla modeli liniowych, pozwala na efektywne generowanie programu o przedstawionych powyżej cechach. Oba algorytmy Autor opracował w wersji numerycznej i analitycznej, gdzie dla każdej chwili próbkowania wartości sygnałów sterujących procesem są obliczane w wyniku rozwiązania zadania optymalizacji kwadratowej z uwzględnieniem ograniczeń wartości i szybkości zmian sygnałów sterujących oraz wartości prognozowanych sygnałów wyjściowych. W analitycznej wersji algorytmów problem sprowadza się do określenia prawa sterowania bez uwzględnienia ograniczeń. Przy opracowaniu implementacji programów zoptymalizowano czas obliczeń i metody są właściwe dla rozwiązywanego zagadnienia.

4. Na czym polega oryginalność rozprawy, co stanowi samodzielny i oryginalny dorobek autora, jaka jest pozycja rozprawy w stosunku do stanu wiedzy czy poziomu techniki reprezentowanych przez literaturę światową?

Na samodzielny i oryginalny dorobek Autora pracy składają się:

- zaimplementowanie w ramach systemu AutoMATiC algorytmów regulacji predykcyjnej,
- opracowanie struktur działania systemu z organizacją zależności pomiędzy poszczególnymi składnikami oraz narzędzi wspomagających projektowanie regulatorów systemu w tym transkompilatora, modułu symulacji oraz profilera,
- opracowanie efektywnej metody projektowania algorytmów regulacji predykcyjnej w postaci kodu z wykorzystaniem systemu AutoMATiC,

- przeprowadzenie badań i testów regulacji predykcyjnej dla wybranych obiektów regulacji, potwierdzających efektywność proponowanego systemu,
- analiza porównawcza systemu AutoMATiC z rozwiązaniami systemów generujących kody regulacji automatycznie.

Oprócz prac związanych z projektowaniem, programowaniem i testowaniem systemu AutoMATiC, prowadzono badania w zakresie modelowania neuronowego, optymalizacji struktury modeli dynamicznych i nieliniowej regulacji predykcyjnej ze szczególnym uwzględnieniem efektywności obliczeniowej nieliniowych algorytmów regulacji predykcyjnej. Prace te były pozytywnie ocenione w czasopiśmie oraz na konferencjach i obejmują jedną publikację w czasopiśmie JCR „Nonlinear Dynamics”, cztery rozdziały w publikacjach wydawnictwa Springer International Publishing, cztery artykuły na konferencjach oraz artykuł w czasopiśmie PAK, z czego 8 pozycji jest cytowanych w pracy.

Autor porównał cechy techniczne systemu AutoMATiC w stosunku do narzędzi ACADO i MPT3 stosując odpowiednie kryteria. Analizowano możliwości generowania kodów dla zadań nieliniowych MPC, modeli w czasie ciągłym i dyskretnych, gdzie uwzględniono: sposób minimalizacji, strukturę regulatora, możliwość użycia innych solverów i rozbudowy o nowe moduły algorytmiczne, możliwość określenia horyzontu predykcji, zajętość pamięci, czas obliczeń oraz inne. Autor wykazał istotne pozytywne cechy systemu AutoMATiC w stosunku do systemów ACADO i MPT3, do których należy zaliczyć niskie zużycie pamięci i krótki czas obliczeń wartości sygnałów sterujących dla określonej próbki oraz możliwość zastosowania wygenerowanego kodu na różne platformy obliczeniowe, a w szczególności na mikrokontrolery jednokładowe.

5. Czy autor wykazał umiejętność poprawnego i przekonującego przedstawienia uzyskanych przez siebie wyników /zwięzłość, jasność, poprawność redakcyjna rozprawy?

Praca pod względem układu i kolejności sformułowanych problemów jest prawidłowo skonstruowana. Praca od strony redakcyjnej została przygotowana z dużą starannością. Język poprawny, jednak Autor się nie ustrzegł kilku błędów językowych, które zostaną przedstawione w dalszej części recenzji. Pewna drobna uwaga dotyczy sformułowania celu pracy, Autor stwierdza, że celem pracy było omówienie systemu do automatycznej generacji kodu. Zdaniem recenzenta celem pracy było stworzenie systemu, który został omówiony w rozprawie.

Błędy językowe:

Strona 8 „natomiast regulacją pętli zajmują się klasyczne regulatory PID” złe sformułowanie.

Strona 33 „... które nastawione są właśnie na sterowanie binarne” złe sformułowanie.

Strona 37 „Zastosowanie tej mieszanki języków jako wejściowych danych ...” złe sformułowane stwierdzenie w zdaniu - zdanie niezrozumiałe.

Strona 40 „.. spreparowane sygnały wygenerowane programowo” złe sformułowanie.

Strona 43 „ ograniczy ć” błąd w tekście.

Strona 48 złe oznaczony rysunek c).

Strona 122 „ nie będzie w stanie zdażyć z wyznaczeniem nowych wartości” złe sformułowanie.

W tekście Autor często używa sformułowania „polityki sterowań” co wydaje się niewłaściwe.

W pracy rysunki nr 3.10, 3.11, 3.12 „ Drzewa składni ...” są nieczytelne i trudno określić różnice – praktycznie rysunki nie wnoszą nowych wartości do pracy.

6. Jakie są słabe strony rozprawy i jej główne wady?

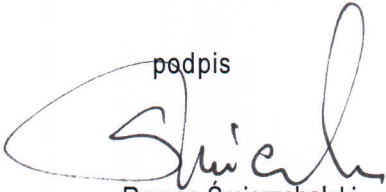
Rozprawa posiada zdaniem recenzenta dyskusyjne słabe strony. Recenzent do pewnego braku zalicza nieopracowanie modułu do rozwiązywania zadań predykcyjnych modeli nieliniowych i porównania efektywności systemu generowania kodu dla tej klasy modeli. Autor podkreślił w pracy, że w dalszym etapie prac nad systemem nastąpi rozszerzenie o moduł predykcyjny dla modeli nieliniowych. Jednak recenzent uważa, że dla tego typu modeli generowanie kodu dla mikrokontrolerów będzie stanowiło znaczną trudność. Wydaje się również, że do testowania obiektów wielowymiarowych zastosowano znacznie uproszczony model MIMO, który nie zawiera sprzężeń skrośnych. Model w postaci prezentowanej w pracy przedstawia dwa modele równoległe SISO. To uproszczenie modelu nie odnosi się całkowicie do modeli MIMO jako wybranych zadań testowych. W tym przypadku wydaje się trudne dokonanie adekwatnych porównań. Jednak należy podkreślić, że uzyskane przez system rozwiązania w postaci generowanego kodu spełniają warunki sterowania predykcyjnego. Zdaniem recenzenta należało podjąć szerszą dyskusję na temat rozwiązań dla innych zadań testowych.

7. Jaka jest przydatność rozprawy dla nauk technicznych?

Analizując recenzowaną pracę, główny nacisk został skierowany na opracowanie systemu generowania kodu dla wybranej klasy zadań predykcji. Pomimo pewnych ograniczeń związanych z predykcją dla modeli liniowych, Autor przedstawił nowe ujęcie zastosowania systemu automatycznie generującego kod algorytmów sterowania predykcyjnego o nazwie „AutoMATiC”. Istotną zaletą systemu to generowanie kodu na jednostki o ograniczonych zasobach oraz niskiej mocy obliczeniowej, gdzie głównym przeznaczeniem będą mikrokomputery jednocukładowe. Autor również wprowadził oryginalne metody transkompilacji i innych operacji przy tworzeniu kodu co pozwala na szybką modyfikację systemu. Autor podjął jednocześnie trudne próby optymalizacji kodu wynikowego co potwierdza duże umiejętności Autora od strony programistycznej. Strojenie licznych parametrów algorytmu wymagało od Autora dużego nakładu czasu i doświadczenia w zakresie testowanych przykładów. Przedstawione w pracy propozycje algorytmu i przeprowadzone symulacje mogą być wykorzystane do rozwiązywania określonej klasy zadań sterowania predykcyjnego, a także mogą być wskazówkami do tworzenia nowych narzędzi rozwiązywania tego typu problemów.

Ważnym faktem oceny osiągnięć Doktoranta było wystąpienie na seminarium wydziałowym na Wydziale Elektrotechniki i Automatyki Politechniki Gdańskiej, gdzie szersze przedstawienie pracy oraz wnioski z dyskusji potwierdzają głęboką znajomość przedstawianej problematyki i wysoki poziom naukowy pracy. Potwierdzeniem bardzo dobrej oceny osiągnięć Doktoranta stanowią również wystąpienia na konferencjach MMAR, przedstawione w publikacjach indeksowanych w WoS jak i jedna publikacja w czasopiśmie JCR „Nonlinear Dynamics”.

8. Do której z następujących kategorii Recenzent zalicza rozprawę:
- a/ nie spełniająca wymagań stawianych rozprawom doktorskim przez obowiązujące przepisy
 - b/ wymagająca wprowadzenia poprawek i ponownego recenzowania
 - c/ spełniająca wymagania
 - d/ spełniająca wymagania z wyraźnym nadmiarem
 - e/ wybitnie dobra, zasługująca na wyróżnienie

podpis

Roman Śmierzchalski



Politechnika
Śląska

DR HAB. INŻ. DARIUSZ BISMOR, PROF. PŚ

Politechnika Śląska, Instytut Automatyki

Zakład Pomiarów i Systemów Sterowania

ul. Akademicka 16, 44-100 Gliwice

tel. 32 237 10 46, e-mail: Dariusz.Bismor@polsl.pl

Gliwice, 8 października 2018

RECENZJA

rozprawy doktorskiej mgr inż. Patryka Chabry

pt.: „*Automatyczna generacja kodu algorytmów regulacji predykcyjnej na mikrokontrolery*”

w dziedzinie **nauk technicznych**, w dyscyplinie **automatyka i robotyka**.

Promotor: dr hab. inż. Maciej Ławryńczuk, prof. PW.

1. Podstawa formalna

Praca została wykonana na zlecenie Dziekana Wydziału Elektroniki i Technik Informatycznych Politechniki Warszawskiej, zgodnie z pismem z dnia 3 września 2018r.

2. Wprowadzenie i ogólna charakterystyka rozprawy

Od wielu lat, dzięki rozwojowi komputerów i oprogramowania, algorytmy sterowania pisze się i uruchamia w środowiskach symulacyjnych, a dopiero później we właściwych środowiskach pracy. Dzięki temu nie jest konieczne czasochłonne i kosztowne strojenie algorytmów i usuwanie ich usterek na rzeczywistym obiekcie regulacji. Jednym z najbardziej popularnych środowisk symulacyjnych jest środowisko Matlab® firmy Mathworks. Środowisko to jednak posługuje się własnym językiem programowania, tzw. językiem M, podczas gdy rzeczywiste urządzenia służące do regulacji (sterowniki i mikrokontrolery) programuje się albo w językach drabinkowych, albo w języku C/C++. Aby zatem móc wykorzystać kod algorytmu sterowania symulowanego w Matlabie, należy dokonać jego translacji, na przykład do języka C. Ważne jest, aby na tym etapie nie wprowadzić do sprawdzonego już w symulacjach kodu błędów, oraz aby proces ten przebiegał szybko, pozwalając na sprawne, wielokrotne testowanie różnych pomysłów i idei. Dlatego najlepiej jest, aby translacja następowała automatycznie, czyli była wykonywana nie przez człowieka, lecz przez specjalne oprogramowanie. O ile tłumaczenie kodu źródłowego na język maszynowy nazywamy kompilacją, tłumaczenie kodu źródłowego w jednym języku na kod źródłowy w innym języku określa się mianem *transkompilacji*. Transkompilacja jest głównym zagadnieniem recenzowanej rozprawy doktorskiej.

Transkompilacja nie jest pojęciem nowym, a wręcz przeciwnie: pierwsze kompilatory języka C++, stworzone w końcu lat 70-tych ubiegłego wieku przez twórcę tego języka, były tak naprawdę transkompilatorami, które tłumaczyły kod klasowy na kod języka C. Warto jednak podkreślić, że w chwili obecnej nie istnieje uniwersalny, zaawansowany transkompilator dla języka M, zaś istniejące rozwiązania (np. Matlab Coder) obciążone są różnymi wadami, uniemożliwiającymi ich szerokie zastosowanie. Fakt ten, oraz przytoczone powyżej przesłanki w pełni uzasadniają wagę zagadnienia podjętego przez doktoranta, jego aktualność oraz

stopień skomplikowania. Warto także podkreślić, że praca zawiera zarówno aspekt poznawczy, jak i praktyczny. Tematyka pracy dotyczy implementacji zaawansowanych algorytmów sterowania (algorytmów predykcyjnych) i ich implementacji na mikrokontrolerach, zwanych często sterownikami dedykowanymi, a zatem mieści się w dyscyplinie *automatyka i robotyka*.

3. Charakterystyka rozprawy

Oceniana praca ma klasyczny układ i zawiera 6 rozdziałów oraz streszczenia w języku polskim i angielskim, spis treści oraz bibliografię. Całość liczy 147 stron. Bibliografia liczy 89 pozycji literaturowych, z których w 10 doktorant jest autorem lub współautorem. Praca jest ilustrowana 51 rysunkami i zdjęciami, wykonanymi w sposób estetyczny i ułatwiającymi zrozumienie omawianych zagadnień. Ponadto, praca zawiera 38 listingów kodu, które są niezbędne do zrozumienia różnych aspektów omawianego w pracy narzędzia do automatycznej generacji kodu.

Pierwszym rozdziałem pracy jest wprowadzenie, w którym Autor omawia zalety regulacji predykcyjnej w porównaniu do klasycznej regulacji PID, posługując się licznymi przykładami z literatury. Doktorant formułuje tutaj także tezę pracy, która mówi, że kody regulatorów predykcyjnych o dobrej jakości dla różnorodnych mikrokontrolerów mogą być automatycznie generowane poprzez opracowane przez Autora narzędzie, nazwane AutoMATiC. Ponieważ dalej Doktorant wyjaśnia szczegółowo, co rozumie przez automatyczną generację kodu oraz jego dobrą jakość, tezę pracy należy ocenić jako jasno i czytelnie sformułowaną. We wprowadzeniu Autor omawia także krótko swój dorobek naukowy, terminologię zastosowaną w pracy, oraz jej układ.

Kolejny rozdział pracy omawia ideę regulacji predykcyjnej, jej historię oraz w szczegółach te algorytmy regulacji predykcyjnej, które znalazły zastosowanie w pracy. W rozdziale tym, jak i w poprzednim, Doktorant bardzo zręcznie operuje odwołaniami do licznych źródeł literaturowych. Niestety, źródła te dotyczą w znakomitej większości regulacji predykcyjnej, a także innych zagadnień teorii i praktyki sterowania. Praca natomiast dotyczy przede wszystkim zagadnień związanych z translacją kodu w jednym języku na inny język programowania (transkompilacją), a patrząc nieco szerzej, zagadnień automatycznej generacji kodu. Chociaż prac na ten temat jest znacznie mniej i są one trudniejsze do znalezienia, powinny one być w pracy przywołane. Za mankament należy uznać, że Doktorant nie odwołuje się w pracy do takich pozycji literaturowych, chociaż z pewnością je zna — o czym świadczą na przykład cytowania w publikacjach Doktoranta. Świadczy o tym także fakt, że Doktorant prawidłowo przeprowadził analizę literatury dotyczącej automatycznej generacji kodu, którą podsumował w postaci tezy.

Rozdział trzeci to najdłuższy z rozdziałów, opisujący zasadniczą część pracy, czyli opracowany system do automatycznej generacji kodu. Na jego początku Autor w sposób syntetyczny opisuje założenia, które przyjął podczas tworzenia systemu, a które są jasne, precyzyjne i uzasadnione. Dalej Doktorant dokonuje podziału zagadnienia projektowania i implementacji regulatora na mikrokontrolerze w poziomie i w pionie. Podział w pionie pozwala wyodrębnić cztery najważniejsze części składowe kodu, zaś podział w poziomie przebiega zgodnie z kolejnymi w czasie etapami projektowania i wdrażania regulatora. Doktorant opisuje także narzędzia, które stworzył dla każdego z tych etapów. Ostatnia, najdłuższa część rozdziału opisuje szczegółowo sposób implementacji systemu transkompilatora oraz założenia i warunki, jakie musi spełniać kod wejściowy (w języku M i C), aby można było wykonać automatyczną generację kodu.

Następny rozdział opisuje badania systemu do automatycznej generacji kodu przeprowadzone przez Doktoranta już po jego stworzeniu. Badania polegały na wygenerowaniu kodu dla czterech algorytmów regulacji predykcyjnej, z których dwa były algorytmami wykorzystującymi optymalizację kwadratową wykonywaną w każdym okresie próbkowania. Warto podkreślić, że badania zostały przeprowadzone na aż trzech obiektach: obiekcie SISO emulowanym programowo przez oddzielny mikrokontroler, serwomechanizmie laboratoryjnym, oraz obiekcie MIMO emulowanym programowo. Na koniec Autor przedstawia także badania układu o podwyższonej niezawodności, do regulacji którego wykorzystano trzy mikrokontrolery, przy czym jeden z nich zawierał układ arbitrażu. Należy zauważyć bardzo duży nakład pracy potrzebny do realizacji takiej liczby eksperymentów oraz fakt, że bez możliwości automatycznej generacji kodu badania takie trwałyby zapewne o wiele miesięcy dłużej.

W przedostatnim rozdziale Doktorant zawarł porównanie stworzonego przez siebie systemu z innymi systemami do automatycznej generacji kodu, dostępnymi dla użytkowników Matlaba. Rozdział rozpoczyna

się od przedstawienia spektakularnej porażki, którą zakończyła się próba wykorzystania rozprowadzanego wraz z Matlabem narzędzia *Matlab Coder*. Dalej Autor opisuje wykorzystanie narzędzi *ACADO Toolkit* oraz *Multi-Parametric Toolkit 3* do generacji kodu regulatora predykcyjnego dla wybranego obiektu MIMO. Autor dokonuje także generacji kodu dla tego samego regulatora i obiektu za pomocą swojego systemu. W wyniku porównania ujawnia się jedna z najważniejszych zalet stworzonego przez Doktoranta systemu: optymalność kodu, która sprawia, że czas wykonywania kodu wygenerowanego przez ten system jest wielokrotnie krótszy od rozwiązań konkurencyjnych. Do innych zalet systemu należą: dobra zajętość pamięci RW mikrokontrolera, możliwość opisu obiektu w postaci ciągłej i dyskretnej, możliwość dowolnego doboru horyzontu sterowania i predykcji, oraz łatwa rozszerzalność systemu o nowe algorytmy regulacji.

W ostatnim rozdziale Doktorant podsumowuje pracę i przedstawia kierunki dalszych badań.

Cała praca napisana jest w sposób zwięzły i jasny, poprawną polszczyzną i z minimalną liczbą błędów redakcyjnych, stylistycznych czy interpunkcyjnych. Z jednym tylko wyjątkiem, wszystkie formułowane przez Autora wypowiedzi można łatwo zrozumieć, co powoduje, że pracę czyta się lekko i z przyjemnością. Prezentowane treści są ponadto ilustrowane rysunkami, wykresami, zdjęciami i fragmentami kodu, które poprawiają jeszcze i tak wysoką jakość przekazu.

4. Najważniejsze osiągnięcia pracy

Rozprawa doktorska mgr inż. Patryka Chabera jest ważnym krokiem na drodze do zautomatyzowania procesu generacji powtarzalnego i o wysokiej jakości kodu regulatorów predykcyjnych na sterowniki wbudowane. Dotychczas istniejące rozwiązania cechują się istotnymi wadami, z których najważniejszą jest generacja kodu o niskiej optymalności, zwłaszcza w sensie czasu wykonywania skompilowanego kodu na mikrokontrolerach. Podjęta tematyka jest ważna ze względu na coraz szersze pola zastosowań regulatorów predykcyjnych, w tym zastosowania wymagające dużych częstotliwości próbkowania. Podjęta tematyka jest aktualna ze względu na powszechność szybkiego prototypowania w rozwoju metod i algorytmów sterowania.

Do najważniejszych osiągnięć pracy zaliczam:

1. wykazanie wad i niedoskonałości istniejących narzędzi do automatycznej generacji kodu,
2. opracowanie sposobu translacji kodu języka Matlaba na kod w języku C z wykorzystaniem analizy leksykalnej i składniowej, z implementacją w środowisku Matlaba,
3. opracowanie sposobu upraszczania drzewa składni kodu, dzięki któremu możliwe jest wygenerowanie optymalnego kodu wynikowego,
4. umożliwienie stopniowego zwiększania udziału sprzętu w trakcie rozwoju oprogramowania: od symulacji całkowicie w Matlabie, poprzez przeniesienie kodu regulatora na kartę mikroprocesora z symulacją pomiarów, aż do całkowitego zawarcia kodu na karcie,
5. opracowanie sposobu przechowywania historycznych danych w zwartej strukturze globalnej, do której dostęp może mieć wiele różnych algorytmów regulacji,
6. opracowanie systemu arbitrażu, pozwalającego na zastosowanie kilku regulatorów jednocześnie w celu zwiększenia odporności układu regulacji na błędy, usterki i awarie sprzętowe.

Należy zauważyć, że translacja kodu z języka Matlab na język C nie należy do zagadnień intuicyjnie prostych. Powodem tego jest przede wszystkim fakt, iż Matlab jest środowiskiem macierzowym, a język C nie. Ponadto język C nie wspiera przeładowania operatorów, co uniemożliwia rozszerzenie tego języka o operacje macierzowe zapisywane w sposób dla ludzi naturalny. Dlatego nawet tak proste wyrażenie w języku Matlaba, jak $x = y * z$, musi w języku C zostać przetłumaczone na wiele linii kodu, zawierających sprawdzanie poprawności rozmiarów, dwie zagnieżdżone pętle, odwołania przez wskaźniki, itd. Dodatkowo, w języku Matlaba przyjęto indeksować wektory i macierze poczynając od 1, zaś w C robi się to poczynając od 0.

Drugą, istotną trudnością jest fakt, iż w Matlabie zmiennych nie trzeba deklarować, natomiast w języku C jest to obowiązkowe. Co więcej, próba odwołania się w języku C do nie istniejącego elementu wektora (tabeli) kończy się zwykle w sposób fatalny dla poprawnego działania programu. Obydwa wymienione aspekty

powodują, że w implementacji transkompilatora nie ma miejsca absolutnie na żaden błąd. Przedstawione w dysertacji przykłady badań z wykorzystaniem napisanego systemu pokazują, że Doktorant takiego błędu nie popełnił.

Ostatnią okolicznością, którą należy zauważyć, jest szerokie opublikowanie rezultatów prowadzonych przez Doktoranta prac na konferencjach krajowych i zagranicznych, pomimo dużego tempa powstawania dysertacji doktorskiej. Doktorant już w roku 2015 opublikował trzy prace, a następnie w roku 2016 cztery. Zwieńczenie tego procesu stanowi publikacja w renomowanym czasopiśmie *Nonlinear Dynamics*, którego *impact factor* wynosi 4,3, a samo czasopismo zajmuje 8 pozycję w rankingu 130 czasopism z dziedziny mechaniki i inżynierii mechanicznej. Publikacja ta ukazała się w roku bieżącym, lecz została zgłoszona w pierwszym kwartale roku 2017.

5. Uwagi krytyczne i pytania

Za zasadniczy mankament pracy uznaję przyjęty do badań i opisany w publikacji sposób próbkowania sygnałów analogowych, a także podejście do modeli dyskretnych powstałych w wyniku takiego procesu dyskretyzacji. Doktorant wykazuje dużą niefrasobliwość w stosunku do filtrów antystroboskopowych. Świadczy o tym chociażby kompletne pominięcie opisu tych filtrów oraz sposobu ich doboru. Wątpliwości budzi także podane na stronie 85, nieuzasadnione stwierdzenie, że filtry antystroboskopowe nie mają wpływu na dynamikę obiektu. Oczywiście, na dynamikę obiektu jako takiego nie mają wpływu, gdyż nie są jego elementem składowym. Natomiast na dynamikę obiektu widzianą przez regulator już mogą mieć, zwłaszcza w przypadku ich niepoprawnego doboru (lub braku takich filtrów). Wpływ taki będzie szczególnie dobrze widoczny, gdy wystąpi zjawisko stroboskopowe, lub gdy pasmo przenoszenia filtrów będzie zbyt wąskie. Podejrzewam, że samo zastosowanie przez Doktoranta filtrów antystroboskopowych zostało wymuszone pojawieniem się podczas badań aliasingu, i jeżeli tak było, to rozwiązanie problemu nazwałbym „inżynierskim”, a nie „naukowym”.

Dalej, opisany na str. 90 sposób akwizycji sygnału analogowego, poprzez wielokrotne uruchomienie konwersji analogowo-cyfrowej z bliżej nie opisaną częstotliwością próbkowania, a następnie przepisywanie (znowu nie opisanej liczby) próbek za pomocą mechanizmu DMA z jednego do innego miejsca w pamięci, gdzie są one dalej przetwarzane, nazwać można w najlepszym przypadku podejściem beztroskim. Podejście to zdało egzamin w warunkach laboratoryjnych, nie znaczy to jednak, że podobnie będzie w innych zastosowaniach.

Kolejnym krokiem wykonywanym podczas odczytu danych z przetworników, opisywanym przez Doktoranta, jest filtracja dolnoprzepustowa i decymacja pozyskanego sygnału. Do filtracji Doktorant posługuje się bardzo prostym filtrem uśredniającym, zwanym inaczej filtrem o średniej ruchomej lub o skończonej odpowiedzi impulsowej (FIR). Literatura anglojęzyczna nazywa ten filtr także filtrem typu *boxcar*. Dobrany przez Doktoranta filtr ma wszystkie współczynniki identyczne, co powoduje, że jego charakterystyka amplitudowa cechuje się niewielką szybkością opadania i stosunkowo dużą wysokością listków bocznych (w paśmie zaporowym). W związku z tym filtr ten (zastosowany w pojedynczej sekcji) słabo nadaje się do roli filtra decymacyjnego. Istnieją natomiast proste zabiegi pozwalające na znaczne polepszenie charakterystyki takiego filtru. Ponadto powinien zostać omówiony dobór długości filtru decymacyjnego, i powinien on korespondować z wybranym współczynnikiem decymacji.

Po pozyskaniu sygnałów wejściowych do regulatora oraz wystąpieniu przerwania zegarowego oznaczającego kolejny okres próbkowania, zaimplementowany przez Doktoranta system wylicza kolejną wartość sterowania, która jest natychmiast przesyłana do przetworników cyfrowo-analogowych. Podejście to nie budzi wątpliwości w przypadku opisywanych analitycznych algorytmów sterowania, których czas obliczeń jest stały i jest wielokrotnie krótszy od przyjętego okresu próbkowania, wynoszącego 200 ms. Jednakże w przypadku algorytmów wykorzystujących optymalizację kwadratową, czasy wykonywania obliczeń różnią się od siebie znacząco w kolejnych krokach, co widać wyraźnie na rysunkach 4.8, 4.10, 4.15 i 4.17. Widziany przez regulator obiekt dyskretny różni się w takim przypadku w kolejnych okresach próbkowania, uniemożliwiając poprawne działanie regulatora predykcyjnego. Dlatego zazwyczaj, po wykonaniu obliczeń wyznaczających kolejną próbkę sterowania, nie wysyła się jej od razu do przetworników wyjściowych, lecz odczeka się do następnej chwili próbkowania sygnału wejściowego.

Co gorsza, w przypadku badań zilustrowanych rysunkami 4.15 i 4.17, czasy wykonywania obliczeń były w kilku przypadkach większe od okresu próbkowania, czyli nastąpiła utrata próbki sygnału wejściowego.

Zachodzi pytanie, dlaczego Doktorant nie zwiększył w tym przypadku okresu próbkowania, jak to się zwykle robi? Zastosowany okres próbkowania, równy 0,25 s, był stosunkowo krótki w porównaniu z dynamiką obiektu (przybliżona reguła mówi o 4–10 próbkach na okres narastania odpowiedzi skokowej, który był w przypadku tego obiektu równy ok. 3 s).

Należy jednak zauważyć, że przedstawiona powyżej krytyka metodologii postępowania podczas sterowania obiektami rzeczywistymi nie umniejsza zasadniczej wartości pracy, którą jest automatyczna generacja kodów regulatorów predykcyjnych na mikrokontrolery. Wskazane błędy, jeżeli w ogóle stanowią część systemu transkompilatora, można banalnie prosto poprawić.

Poniżej przedstawiam pozostałe uwagi o charakterze krytycznym i polemicznym, oraz pytania, które nasunęły się po lekturze pracy. Uwagi te również nie umniejszają jakości pracy, którą oceniam bardzo wysoko, powinny natomiast zostać potraktowane jako wskazówki dla Autora i zostać uwzględnione w dalszych jego pracach.

1. Równania są jednymi z najtrudniejszych do prezentacji w każdej pracy naukowej elementów, dlatego warto podjąć wysiłek ich starannej redakcji. Autor natomiast w monografii zaniechał powszechnie stosowanej w pracach naukowych konwencji pisanie wektorów i macierzy *zawsze* czcionką pogrubioną (lub rzadziej z strzałką, np. $\vec{q}$). Autor nie jest przy tym konsekwentny, co widać wyraźnie np. w równaniu (2.9), gdzie macierz $\mathbf{A}$ jest pogrubiona, a Λ_p już nie. W sposób zasadniczy utrudnia to zrozumienie zapisów równań na stronie 20 i następnych, oraz nieco mniej już w rozdziale 5.
2. Porównanie symbolu kwalifikowanego atrybutem `weak` do mechanizmu polimorfizmu języków wspomagających programowanie orientowane obiektowo (str. 50) jest nieuzasadnione. Polimorfizm bowiem „działa” na etapie kompilacji programu, zaś kwalifikacja (lub jej brak) atrybutem `weak` jest wykorzystywana przez program konsolidujący („linker”). Widać to wyraźnie w przywoływanym przez Doktoranta środowisku Arduino które, chociaż wspiera programowanie orientowane obiektowo, używa także atrybutu `weak`.
3. Typ występujących w strukturze `ArchiveData` zmiennych `number_of_u`, `number_of_y`, `size_of_u`, `size_of_y` określono jako `long`. Zachodzi pytanie dlaczego zmienne te nie mogą być typu `unsigned int`? Pojemność tego ostatniego na platformach 32-bitowych wynosi przeszło 4 mld, co powinno wystarczyć do określenia rozmiaru tablic na ograniczonych sprzętowo mikrokontrolerach.
4. Jedną z bardzo istotnych zalet regulacji predykcyjnej jest możliwość uwzględnienia przyszłych zmian wartości sterowania podczas obliczania sterowania w danej chwili. Zachodzi więc pytanie dlaczego Doktorant zrezygnował z tej, nie wymagającej dużych nakładów pracy, możliwości (str. 54)?
5. Celem Doktoranta było uzyskanie w efekcie transkompilacji kodu dobrej jakości, przez który Doktorant rozumie kod, który wykonuje się szybko. Dlatego jako spory błąd można uznać zastosowanie bufora przesuwanego w funkcji interfejsowej `new_output()` struktury `ArchiveData` (str. 58). Zamiast tego należało zastosować bufor cykliczny.
6. Doktorant w znakomitej większości stosuje angielskie nazwy zmiennych, struktur i funkcji, co jest często spotykaną konwencją. Dlatego zastanawiające jest przejście na nazwy polskie w opisie biblioteki do obliczeń macierzowych (str. 60 i 61). Co więcej, bardzo skrótowy opis funkcji bibliotecznych znacząco odbiega od opisu innych, zamieszczonych w pracy funkcji. Dla przykładu, nie są podawane deklaracje funkcji bibliotecznych, a jedynie zapis pokazujący sposób ich wywołania. Powodem mogła być chęć nie koncentrowania się na opisie biblioteki, stanowiącej niewielki wkład naukowy w pracę. Jednakże nie sposób zauważyć tutaj, że doskonałym sposobem pogodzenia obydwu tych sprzeczności byłoby zamieszczenie w pracy załącznika z dokładniejszym opisem biblioteki. Nie jest to jedyne miejsce w pracy, w którym brak odwołania do załącznika jest zauważalny.
7. Stwierdzenie na stronie 85, że przesył danych trwa istotnie długo w porównaniu z okresem próbkowania wymaga uzasadnienia w obliczu faktu, że UART układu STM32F7 może pracować z maksymalną szybkością dochodzącą do 4,5 Mb/s.
8. Do badań algorytmów regulacji z obiektem SISO emulowanym poprzez układ elektroniczny (str. 92) wybrano transmitancję dwuinercyjną. Jednakże konkretne współczynniki tej transmitancji zostały

dobrane bardzo niefortunnie: tak, że jedna stała czasowa obiektu wynosi około 1/3 s, zaś druga 1/30 s. Tak duży rozrzut stałych czasowych powoduje, że odpowiedź skokowa transmitancji niewiele różni się od odpowiedzi transmitancji pierwszego rzędu z dominującą stałą czasową (1/3 s). Widać to jeszcze lepiej po dyskretyzacji tego obiektu z założonym okresem próbkowania wynoszącym 0,2 s: jeden biegun obiektu zdyskretyzowanego ma wartość $z_1 = 0,549$, zaś drugi $z_2 = 0,0025$, i jest pomijalnie mały.

9. Również niefortunnie została dobrana macierz transmitancji obiektu MIMO użyta do badań opisanych w punkcie 4.5 (str. 102). Macierz ta jest diagonalna, co powoduje, że badany obiekt MIMO zachowuje się jak dwa niezależne (a ponadto identyczne!) obiekty SISO.

6. Uwagi szczegółowe

Poniżej zamieszczam listę uwag szczegółowych o charakterze merytorycznym. Pomijam uwagi dotyczące redakcji tekstu, bo drobne błędy redakcyjne są nieodłącznie związane z pierwszą edycją każdej monografii.

1. Na str. 8 Doktorant niepotrzebnie dosłownie przetłumaczył angielską nazwę *programmable logic controller*, podczas gdy w języku polskim przyjęło się mówić *sterownik programowalny*.
2. Na tej samej stronie, wymieniając literaturę dotyczącą sterowania silników spalinowych, Doktorant podaje pozycję [16], która dotyczy silników elektrycznych.
3. Użyte przez Doktoranta na str. 17 sformułowanie „(...) *zapewniając jednocześnie większą stabilność trajektorii sygnałów*” jest bardzo niefortunne, zwłaszcza w kontekście dziedziny automatyki, w której doktorat jest broniący. Pojęcia: sygnał, trajektoria i stabilność są bowiem w automatyce bardzo precyzyjnie określone.
4. Na str. 23 Doktorant zauważa, że algorytm DMC jest strojony za pomocą modelu odpowiedzi skokowej, która jest bardzo łatwa w pozyskaniu, co pozytywnie odróżnia go od algorytmu PID. Zwraca się uwagę, że istnieje cały szereg metod strojenia algorytmu PID bez konieczności przeprowadzania eksperymentu ze wzmocnieniem granicznym, w tym metody wykorzystujące odpowiedzi skokowe.
5. Na stronie 36 Doktorant pisze, że jednym z założeń podczas projektowania systemu AutoMATiC było, aby do zaprojektowania regulatora nie było potrzebne nic prócz projektowanego systemu i zestawu bibliotek. Doktorant zapomina, że potrzebne jest jeszcze środowisko Matlab, w którym zaimplementowano stworzony system.
6. Na str. 41 Doktorant pisze, że „(...) *jedna iteracja pętli [głównej programu] jest równoznaczna z upłynięciem jednej chwili czasu dyskretnego*”. Taka definicja budzi zastrzeżenia, gdyż program wykonywany w pętli głównej może zawierać instrukcje warunkowe, które spowodują, że poszczególne chwile czasu dyskretnego nie będą od siebie równoodległe. A nawet w przypadku, gdyby takich instrukcji nie było, odmierzenie czasu dyskretnego szybkością wykonania pewnej porcji kodu na jakimś mikroprocesorze nie jest naturalnym podejściem w automatyce.
7. Na str. 45 Doktorant opisuje sposób wywołania transkompilatora, podając nazwę polecenia go uruchamiającego, oraz możliwe parametry wywołania. Do tego momentu nigdzie w monografii nie pada informacja, że transkompilator został zaimplementowany w języku Matlaba (taka informacja pojawia się po raz pierwszy dopiero na stronie 62), co utrudnia zrozumienie tego fragmentu pracy podczas pierwszego jej czytania.
8. Zdanie na str. 47 rozpoczynające się od „*Na rysunkach zaprezentowano dodatkowo...*” jest wielokrotnie złożonym zdaniem o trudnej do zrozumienia treści. Warto byłoby go podzielić na kilka krótszych zdań.
9. Na stronie 52, w opisie funkcji `timeout()`, Doktorant stwierdza: „*Oznacza to, że regulator jest niewystarczająco szybki, aby poprawnie sterować obiektem regulacji*”. Takie stwierdzenie, wobec nie zdefiniowania obiektu regulacji, o którym mowa, jest nieuprawnione. Wystąpieniu przekroczenia czasu pojedynczej iteracji można zapobiec zwiększając okres próbkowania, lub zmieniając algorytm regulacji na prostszy, a nie konieczne zmieniając regulator (sprzęt) na taki o większych możliwościach.

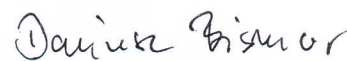
10. Na stronie 53 zdanie „*Aby ułatwić projektantom algorytmów regulacji ... na wartość obecnej chwili czasowej*” jest niedokończone.
11. Na stronie 85 omawiane są przykładowe obiekty regulacji wykorzystywane w badaniach, a na następnych stronach zamieszczone są ich zdjęcia. Szkoda, że Doktorant nie umieścił schematów (ideowych lub chociaż blokowych) stosowanych kombinacji sterownika i obiektów, które znacznie lepiej ilustrowałyby przedstawione badania.
12. Na stronie 92 Doktorant podaje we wzorze (4.1) transmitancję emulowanego obiektu w postaci współczynników wielomianu. Szkoda, że nie podaje także postaci z zerami i biegunami, która jest znacznie łatwiejsza do zrozumienia.
13. Na stronie 113 Doktorant przypisuje istnienie rozbieżności pomiędzy wyjściami „*niedoskonałościom sprzętu pomiarowego*” zapominając, że głównym źródłem błędów są zwykle zakłócenia pomiarowe.
14. Użyte na rysunku 5.1 na stronie 129 oznaczenia y_1 , y_2 , y_1^{zad} i y_2^{zad} nie korespondują z wzorami przytoczonymi na stronie poprzedniej.

7. Wniosek końcowy

W swojej rozprawie doktorskiej mgr inż. Patryk Chaber trafnie zidentyfikował zagadnienie naukowe, które nie zostało jeszcze w sposób zadowalający rozwiązane, i sformułował tezę, że rozwiązanie tego zagadnienia można znacznie poprawić. Następnie udowodnił tą tezę opracowując autorski system do automatycznej generacji kodu regulatorów predykcyjnych na mikrokontrolery, nazwany AutoMATiC. Aby udowodnić poprawność działania opracowanego narzędzia, przeprowadził wiele eksperymentów z wykorzystaniem płyty rozwojowej mikrokontrolera STM32F7 oraz kilku wybranych obiektów regulacji: rzeczywistych lub symulowanych za pomocą innej płyty zawierającej mikrokontroler.

W pracy Autor wykazał się znajomością algorytmów regulacji predykcyjnej, metod optymalizacji, metod numerycznych oraz programowania w języku Matlab oraz C. Uważam, że praca zawiera oryginalne rozwiązanie problemu transkompilacji kodu napisanego w języku wysokiego poziomu (Matlab), przydatnego na etapie symulacji, do kodu, który może zostać skompilowany dla dowolnego mikrokontrolera (kodu w C), i zastosowany w rzeczywistym układzie regulacji.

Stwierdzam, że rozprawa doktorska mgr inż. Patryka Chabera spełnia wymogi obowiązującej *Ustawy z dnia 14 marca 2003 o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki*, stawiam wniosek o przyjęcie rozprawy doktorskiej i dopuszczenie mgr inż. Patryka Chabera do jej publicznej obrony.



dr hab. inż. Dariusz Bismor