

## EFEKTYWNE KSZTAŁCENIE INŻYNIERÓW Z WYKORZYSTANIEM LABORATORYJNYCH SYSTEMÓW MECHATRONICZNYCH I STOSOWANIEM METODOLOGII SZYBKIEGO PROTOTYPOWANIA ALGORYTMÓW STEROWANIA

Dawid KNAPIK, Krzysztof KOŁEK, Maciej ROSÓŁ, Andrzej TURNAU

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie  
tel.: 126341568 e-mail: {knapik, kko, mr, atu}@agh.edu.pl

**Streszczenie:** W pracy przedstawiono metodykę nauczania opartą na wykorzystaniu laboratoryjnych systemów mechatronicznych oraz narzędzi do szybkiego prototypowania algorytmów sterowania. Stosowana metoda jest odpowiedzią na zapotrzebowanie rynku pracy na inżynierów z praktyką przy jednoczesnym uwzględnieniu finansowych możliwości uczelni wyższych. Istotną kwestią w procesie edukacji na studiach technicznych jest równowaga pomiędzy nauczaniem teorii a przekazywaniem umiejętności praktycznych. Przedstawiana metoda umożliwia zrealizowanie wymagań stawianych współczesnym laboratoriom technicznym.

**Słowa kluczowe:** studia inżynierskie, systemy mechatroniczne, szybkie prototypowanie algorytmów, laboratorium sterowania.

### 1. WSTĘP

W artykule przedstawiono metodykę nauczania z wykorzystaniem eksperymentów praktycznych stosowaną na studiach technicznych. Głównymi cechami omawianego rozwiązania jest wykorzystanie oprogramowania wspomagającego, w szczególności technik szybkiego prototypowania algorytmów oraz laboratoryjnych systemów mechatronicznych.

Istotną kwestią w procesie edukacji technicznej jest znalezienie równowagi pomiędzy przekazywaniem wiedzy teoretycznej a kształceniem umiejętności praktycznych [1,2]. Jako zauważono w [1] właściwy dobór proporcji pomiędzy teorią a praktyką jest trudny z wielu powodów. Z jednej strony zbyt duży nacisk na nauczanie czysto teoretyczne pozbawia studentów intuicji i praktyki inżynierskiej. Gąszcz matematycznych równań może doprowadzić do zagubienia idei prezentowanych wzorami. Student biegle w matematyce może mniej sprawnie poruszać się w jej zastosowaniach. Z drugiej strony mnogość praktycznych przykładów pozwala na wrywkowe przedstawienie teorii. Bez teorii student nie odnajdzie powiązań na podstawie doświadczeń. Nauczanie praktyczne (eksperymentalne) i teoretyczne (wykorzystujące formuły matematyczno-fizyczne) wymagają odpowiedniej kadry naukowej. O ile przy nauczaniu teoretycznym jest to podstawowe i najważniejsze wymaganie, o tyle w przypadku nauczania praktycznego lista wymagań jest znacznie dłuższa. Nawiązując do [3] dobry eksperyment laboratoryjny powinien:

- demonstrować istotne problemy teoretyczne,
- odzwierciedlać zagadnienia spotykane w praktyce,

- być interesujący wizualnie i akustycznie,
- mieścić się w ramach czasowych,
- nie stwarzać zagrożeń,
- mieć akceptowalny koszt,
- być łatwy do zrozumienia i przedstawienia.

Niniejsza praca stara się być pomocna we wskazaniu technik i technologii, które pomagają spełnić powyższe założenia. Praca jest adresowana do kadry naukowej uczącej: automatyki, mechatroniki, algorytmów sterowania, metrologii i pokrewnych dziedzin uczelni technicznych,

Praca jest zorganizowana w następujący sposób. W rozdziale drugim przedstawiono narzędzia programowe wspomagające nauczanie. Główny nacisk kładzie się na środowisko programowe MATLABa wraz z Simulinkiem ze względu na jego popularność zarówno w ośrodkach akademickich jak i w firmach badawczo-rozwojowych. Przedstawiono potencjał symulacyjny tego środowiska, wyrażając opinię na temat czystej symulacji bez eksperymentów rzeczywistych jako niewłaściwego kierunku rozwoju edukacji. W kolejnym rozdziale pokazano ścieżkę szybkiego prototypowania algorytmów oraz automatyczną generację kodu. Omówiono jak metoda szybkiego prototypowania algorytmów, mająca często u podstaw eksperymenty symulacyjne i prace na modelu, w połączeniu z automatyczną generacją kodu, pozwala rozwiązywać problemy bez wnikania w szczegóły realizacji. W rozdziale czwartym odniesiono się do współczesnych kart pomiarowo-sterujących, platform edukacyjnych i mikrokontrolerów, puentując jak postęp w dziedzinie elektroniki wpływa na procesy kształcenia. W rozdziale piątym przedstawiono wybrane systemy mechatroniczne spełniające najważniejsze wymogi edukacyjno-badawcze. Na koniec podsumowano zalety i wady proponowanych podejść.

Na czym polega zatem efektywne kształcenie inżynierów zapowiedziane w tytule pracy? Odpowiedź odnajdujemy pośrednio w tematyce przedstawionej w kolejnych rozdziałach. Jaki jest zatem model inżyniera naszych czasów? Inżynier przede wszystkim ma posiadać szeroką, ale i specjalistyczną wiedzę popartą doświadczeniem praktycznym. Wykształcony inżynier w dziedzinie automatyka i robotyka oraz w dziedzinach pokrewnych musi radzić sobie z projektowaniem i odpowiednią konfiguracją układów sterowania dla rzeczywistych obiektów. W kolejnych rozdziałach tej pracy proponuje się metodologię kształcenia pozwalającą na

edukację inżynierów odpowiadających na współczesne zapotrzebowanie.

## 2. NAUCZANIE WSPOMAGANE KOMPUTEROWO

Oprogramowanie wspomagające obliczenia inżynierskie stało się nieodzownym narzędziem pracy w środowisku akademickim. Jego wykorzystanie w zastosowaniach przemysłowych staje się coraz bardziej popularne [4].

Wśród przykładów takiego oprogramowania warto wymienić oprogramowanie MATLAB/Simulink, a także, konkurencyjne, bezpłatne rozwiązania takie jak GNU Octave czy Scilab.

Podstawową funkcją ww. pakietów jest wspomaganie obliczeń numerycznych i wizualizacja wyników. Zbiór funkcji matematycznych, składnia umożliwiająca obliczenia macierzowe oraz prosty język skryptowy powodują, że programy te są popularne w szczególności w środowisku akademickim [5]. Dalszy, intensywny rozwój wspomnianego oprogramowania znacznie zwiększył jego możliwości, które aktualnie dostarczają specjalistycznej funkcjonalności dostosowanej do różnych dziedzin (wśród nich możemy wymienić obliczenia symboliczne, sztuczne sieci neuronowe, obliczenia statystyczne). Dodatkowo istnieje możliwość przeprowadzania eksperymentów symulacyjnych na podstawie modeli zbudowanych w postaci diagramów blokowych programu Simulink (lub jego odpowiednika pakietu Scilab - Xcos). Dzięki intuicyjnemu interfejsowi, bazie standardowych funkcji (bloków) oraz zaawansowanym solverom oprogramowanie jest chętnie używane. Zalety oprogramowania wspomagającego są oczywiste, przede wszystkim pozwalają na skupieniu się na rozwiązywaniu postawionego problemu bez konieczności pracochłonnej implementacji.

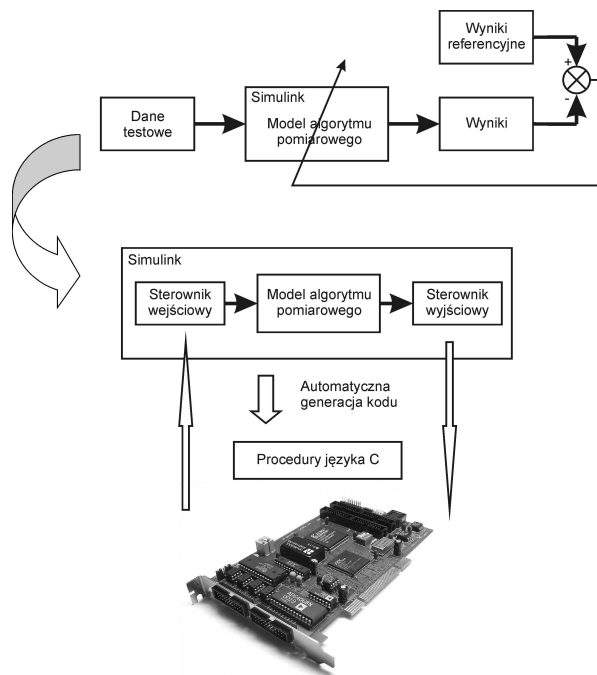
Jedną z możliwości symulacji komputerowych jest modelowanie procesów i urządzeń spotykanych w świecie rzeczywistym. W związku z tym pojawia się pytanie czy istnieje możliwość zastosowania symulacji komputerowych w zastępstwie pracy ze sprzętem. Zaletą takiego rozwiązania jest zastąpienie kosztownego sprzętu przez programy komputerowe. Należy jednak zauważyć, że nawet najlepsza symulacja nie oddaje w 100% rzeczywistości. Zawsze opiera się na pewnych założeniach upraszczających i idealizacjach, a więc nigdy nie zastąpi fizycznego urządzenia. Co więcej student powinien zetknąć się ze zjawiskami: szumów pomiarowych, niepewności modelowania, ograniczeń w sterowaniu, itp. [6, 7].

Warto w rozdziale o projektowaniu wspomaganym komputerowo spytać na koniec, czy zalety oprogramowania wspomagalnego można wykorzystać w pracy z układami rzeczywistymi? Czy korzysta się z narzędzi usprawniających identyfikację, dobór regulatorów i symulację działania w celu przyspieszenia prac?

## 3. SZYBKIE PROTOTYPOWANIE ALGORYTMÓW STEROWANIA

Główną zaletą metod szybkiego prototypowania algorytmów jest skupienie się na rozwiązaniu problemu pomijając przy tym szczegóły implementacyjne [8]. Podstawowym narzędziem stosowanym w tej metodzie jest automatyczna generacja kodu. Jak pokazano na rysunku 1. punktem wyjściowym jest opis algorytmu w postaci schematu blokowego. Schemat określa sposób przetwarzania

danych, począwszy od sygnałów wejściowych, a skończywszy na sygnałach wyjściowych. Do jego budowy używa się funkcji bibliotecznych. Model przepływu danych jest wzbogacony o sterowniki urządzeń wejścia/wyjścia. Należą do nich: przetworniki A/C i C/A, interfejsy komunikacyjne, wejścia/wyjścia cyfrowe. Na podstawie schematu generowany jest kod C/C++, który następnie jest kompilowany na platformę docelową. Może być to komputer PC z oprogramowaniem umożliwiającym uruchomienie skompilowanej aplikacji jako zadania czasu rzeczywistego lub platforma sprzętowa przystosowana do takiego działania.



Rys. 1. Schemat generacji kodu w technice szybkiego prototypowania

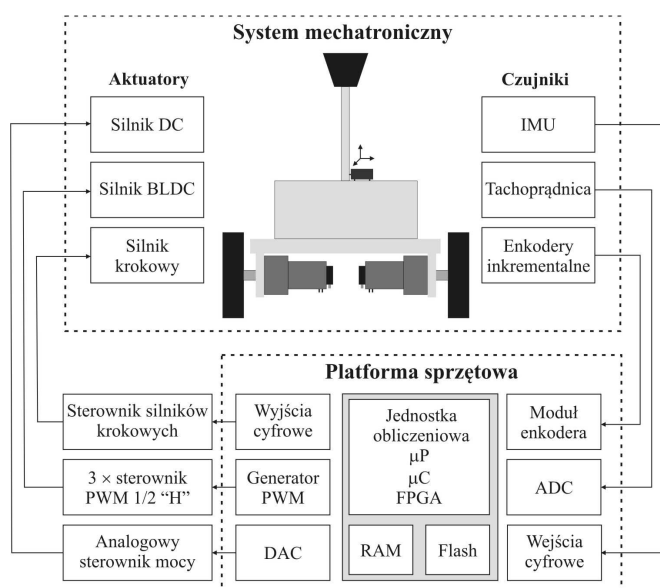
Również pakiet MATLAB/Simulink może być użyty do automatycznego generowania kodu m.in. na podstawie diagramu programu Simulink (istnieją specjalne przybory takie jak *Simulink Coder* i *Embedded Coder*, które realizują to zadanie). Oznacza to, że oprogramowanie, które wykorzystywano do obliczeń inżynierskich czy do modelowania i symulacji, może również posłużyć do realizacji algorytmu regulatora, czy algorytmu przetwarzania danych. Podejście, w którym modelowania matematyczne i symulacje modelowe doprowadzają do zaprojektowania i weryfikacji algorytmu, a następnie jego sprzętowej realizacji określane jest angielskim terminem Model Based Design (MBD). To co wyróżnia podejście MBD od standardowego podejścia jest uniknięcie pracowej implementacji algorytmu wymagającej ręcznego programowania na rzecz automatycznej generacji kodu.

## 4. PLATFORMY SPRZĘTOWE

Jak wspomniano w rozdziałach 2. i 3., w procesie dydaktycznym związanym z nauczaniem metod tworzenia oprogramowania dla systemów mechatronicznych, istotne jest wykorzystanie nowoczesnych środowisk programowych takich jak MATLAB/Simulink czy LabVIEW. Pozwalają one na wykorzystanie gotowych bibliotek do tworzenia algorytmów pomiarowo-sterujących oraz udostępniają narzędzia umożliwiające automatyczną generację kodu dla

docelowych platform sprzętowych, podłączonych do badanego obiektu. Należy jednak zaznaczyć, że gotowe rozwiązania, rozumiane jako udostępnienie przez środowisko programowe pełnej ścieżki realizacji aplikacji sterującej obiektem (od powstania graficznej reprezentacji aplikacji do gotowego kodu uruchamianego na sprzęcie) jest zapewniane jedynie dla ograniczonej liczby sterowników sprzętowych. Stąd przy planowaniu zajęć dydaktycznych, których celem jest praktyczna nauka tworzenia systemów sterowania, bardzo ważny jest wybór platformy sprzętowej, która będzie podłączona do czujników i elementów wykonawczych sterowanego obiektu mechatronicznego. W założeniu należy rozważyć te rozwiązania, które są w pełni wspierane przez ww. środowiska programowe. Wsparcie to może być oferowane przez producenta sprzętu lub producenta środowiska programowego.

W dalszej części tego rozdziału przedstawiono wybrane platformy sprzętowe, które są w pełni wspierane przez środowisko MATLAB/Simulink oraz mogą być wykorzystane do sterowania obiektami mechatronicznymi. Ogólną ideę połączenia sterownika sprzętowego do obiektu przedstawiono na rysunku 2. Można na nim wyróżnić system mechatroniczny wyposażony w czujniki i elementy wykonawcze, sterowniki mocy dla urządzeń wykonawczych oraz platformę sprzętową. Platforma sprzętowa zawiera jednostkę obliczeniową, pamięci programu i danych oraz wymagane do utworzenia pętli sprzężenia zwrotnego, interfejsy wejścia/wyjścia.



Rys. 2. Schemat połączenia sterownika sprzętowego do obiektu

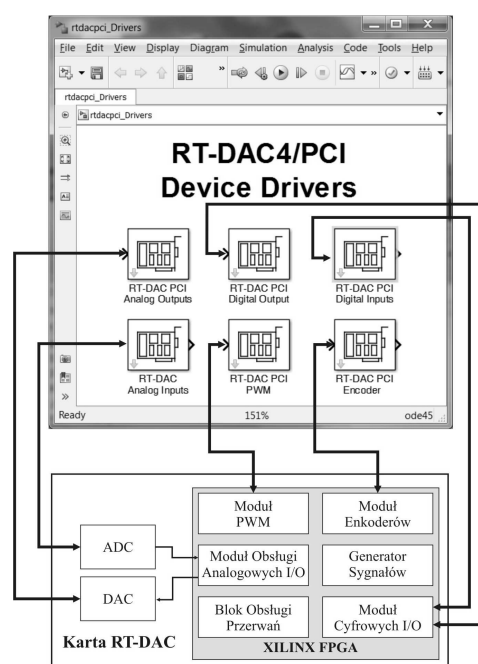
Przy selekcji prezentowanego sprzętu założono, że musi on posiadać następujące interfejsy wejścia/wyjścia:

- Przetworniki ADC o rozdzielczości, minimum, 10 – bitowej (podłączenie sygnałów z tachoprądnic, czujników MEMS z wyjściem analogowym),
- Cyfrowe kanały pomiarowe dla enkoderów inkrementalnych,
- Przetworniki DAC o rozdzielczości, minimum, 10 – bitowej (analogowe sterowniki mocy),
- Generatory sygnału PWM o rozdzielczości, minimum, 8 – bitowej (sterowniki mocy dla obciążeń indukcyjnych typu: zawory, pompy, silniki DC i BLDC).

Należy zaznaczyć, że nowoczesne czujniki pomiarowe i sterowniki urządzeń wykonawczych coraz częściej wyposażane są w szeregowo interfejsy komunikacji, takie jak SPI, I2C. Stąd ich obecność na wybranym sterowniku sprzętowym może okazać się niezbędną.

#### 4.1. Karta RT-DAC

Karta RT-DAC (ang. Real-Time Data Acquisition Board) jest wielofunkcyjną kartą wejść/wyjść analogowych i cyfrowych [9]. Produkowana jest w wersjach z następującymi interfejsami komputerowymi: PCI (zdjęcie z rysunku 1.), USB 2.0 i PCI Express. Najważniejszym elementem tej karty jest rekonfigurowalny układ FPGA firmy Xilinx, umożliwiający zmianę jej funkcjonalności. Oznacza to, że urządzenie RT-DAC można dostosować do konkretnego systemu mechatronicznego. Przykładowo do sterowania, opisanym w rozdziale 5., serwonapędem prądu stałego wykorzystano: 2 analogowe kanały pomiarowe, 1 wejście dla enkodera inkrementalnego, 1 generator sygnału PWM oraz precyzyjny sygnał timera do odmierzenia czasu. Należy zaznaczyć, że wszystkie kanały pomiarowe i sterujące są w pełni konfigurowalne. Producent udostępnia sterowniki programowe dla środowiska MATLAB/Simulink, obsługujące wszystkie dostępne układy peryferijne karty. Ogólny schemat budowy RT-DAC wraz z odniesieniem do bloków sterowników programowych Simulinka, obsługujących podstawowe interfejsy wejścia/wyjścia karty, przedstawiono na rysunku 3.



Rys. 3. Schemat budowy RT-DAC z odniesieniem do bloków sterowników programowych Simulinka

#### 4.2. Sterownik STM32F4 Discovery

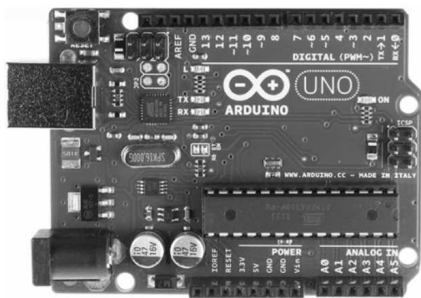
System uruchomieniowy STM32F4 Discovery to produkt firmy STMicroelectronics przeznaczony do testów tzw. sterowników wbudowanych [10]. Najważniejszym jego elementem jest jednocukładowy mikroprocesor STM32F407VGT6. Posiada on 32-bitowy rdzeń ARM Cortex-M4 z koprocesorem FPU (ang. Floating- Point Unit), wejścia/wyjścia analogowe, wejścia dla dwóch enkoderów inkrementalnych, 1MB pamięci Flash oraz 192kB RAM. W skład STM32F4 Discovery wchodzi również m.in. programator STLink, 3-osiowy akcelerometr, diody LED i

przyciski użytkownika, wyprowadzenia portów I/O oraz złącze micro-USB.

Wsparcie dla procesora STM32F4 zapewnione jest przez producenta układu – STMicroelectronics (*STM32 embedded target for MATLAB and Simulink*) [11] oraz firmę Aimagin (*Waijung Blockset*) [12]. W obu przypadkach udostępnione są sterowniki programowe do obsługi wejść/wyjść procesora oraz wybranych interfejsów komunikacyjnych (SPI, I2C, RS232). Do generacji aplikacji wymagany jest przyborek *Embedded Coder* firmy MathWorks.

#### 4.3. Arduino

Arduino stanowi jeden z najbardziej znanych systemów procesorowych powstałych jako projekt typu OpenHardware. Stosuje się go przede wszystkim do nauczania robotyki, automatyki i programowania na kierunkach technicznych. Aktualnie istnieje około 20 standardowych modeli Arduino, przy czym najbardziej znane to: Uno, Due, Leonardo, Mega, Mini, Micro i Nano. Typowa płyta PCB systemu Arduino zawiera: 8-bitowy mikrokontroler AVR ATmega firmy Atmel, cyfrowe i analogowe wyprowadzenia wejść/wyjść mikrokontrolera oraz interfejsy UART/USB, I2C, SPI. Wygląd ogólny Arduino Uno pokazano na rysunku 4.



Rys. 4. Wygląd ogólny Arduino Uno

Funkcjonalność Arduino można rozszerzyć wykorzystując dostępne na rynku płyty rozszerzeń nazywane, z języka angielskiego, *shield*. Płyty te podłącza się do standardowych złączy zamontowanych na płycie bazowej systemu. Najczęściej stosowane są rozszerzenia oferujące połączenia Ethernet, Bluetooth, ZigBee oraz sterowniki silników (DC, BLDB, krokowych).

Realizacja aplikacji dla Arduino odbywa się najczęściej za pośrednictwem środowiska Arduino IDE (dedykowany język programowania bazujący na C/C++). Do praktycznej nauki sterowania systemami mechatronicznymi, na poziomie wyższej szkoły technicznej, zaleca się jednak wykorzystanie środowiska MATLAB/Simulink. Producent zapewnia bibliotekę bloków Simulinka do konfiguracji oraz dostępu do czujników, elementów wykonawczych i interfejsów komunikacyjnych docelowej platformy Arduino. Dodatkowo możliwe jest monitorowanie zmiennych procesowych oraz strojenie algorytmów sterowania z użyciem tego samego modelu Simulinka, na podstawie którego wygenerowano kod programu (praca w tzw. trybie *External*).

#### 4.4. Xilinx Zynq

Interesującą platformą dla systemów mechatronicznych są układy typu SoC (System on Chip), czyli układy łączące w jednej obudowie rdzenie mikroprocesorowe, pamięć, przetworniki ADC i DAC oraz specjalizowane układy wejścia i wyjścia. Jako przykład może posłużyć rodzina

Zynq firmy Xilinx. Układy rodziny Zynq określane są przez producenta mianem Extensible Processing Platform (EPP). W pojedynczym układzie scalonym umieszczono dwurdzeniowy procesor ARM Cortex-A9 oraz matrycę rekonfigurowaną FPGA. Rozszerzalność (ang. extensibility) platformy Zynq, związana jest z obecnością matrycy FPGA, w której zaimplementowana może zostać część zadań pomiarowo-sterujących, w szczególności zadań wymagających reakcji układu liczonej w nanosekundach.

W przypadku układów Zynq wspomniana w rozdziale 3. metoda szybkiego prototypowania może dotyczyć zarówno aplikacji czasu rzeczywistego wykonywanej przez procesor (lub procesory) jak i konfiguracji matrycy FPGA. Podobnie jak dla trzech, opisanych wcześniej sterowników sprzętowych, układ typu Zynq 7000 posiada wsparcie dla automatycznej generacji aplikacji w środowisku MATLAB/Simulink. Firma MathWorks udostępnia bibliotekę bloków dla układów peryferyjnych Zynq 7000 oraz możliwość automatycznej generacji kodu VHDL (przyborek *HDL Coder*) dla układu FPGA i C/C++ dla procesora ARM Cortex-A9 (przyborek *Embedded Coder*) [13]. Najczęściej stosowane płyty rozwojowe z układem Zynq 7000 to: ZedBoard firmy Avnet, Xilinx ZC702 Evaluation Kit, Xilinx ZC706 Evaluation Kit i Digilent ZYBO Zynq.

### 5. EDUKACYJNE SYSTEMY MECHATRONICZNE

Wymagania stawiane współczesnym laboratoriom technicznym, zarysowane we wcześniejszych rozdziałach, sugerują użycie specjalistycznych urządzeń. W niniejszym rozdziale zaprezentowano kilka wybranych systemów mechatronicznych używanych w Katedrze Automatyki i Inżynierii Biomedycznej, AGH w Krakowie w celach edukacyjnych oraz badawczych. Wśród zaprezentowanych rozwiązań znalazły się serwomechanizm, suwnica bramowa, oraz robot balansujący (Lego Mindstorm i UnTrans).

#### 5.1. Serwomechanizm

Serwomechanizm jest prostym systemem o bardzo szerokich możliwościach edukacyjnych. Urządzenie składa się silnika prądu stałego, na wale którego znajduje się tarcza (koło zamachowe) oraz enkoder mierzący położenie katowe wału, dodatkowo umieszczona jest tarcza z potencjometrem do zadawania sygnału odniesienia. W komplet stanowiska wchodzi również karta pomiarowo-sterująca oraz komputer PC z środowiskiem programowym MATLAB /Simulink.

Ponieważ silniki są najczęściej spotykanym elementem wykonawczym, to zjawiska obserwowane przy pracy z serwomechanizmem (takie jak tarcie, szумы pomiarowe, ograniczenie mocy sterownika) są powszechnie występujące w praktyce inżynierskiej.

Mimo prostej budowy stanowisko pozwala na przedstawienie szerokiej gamy problemów, począwszy od badania właściwości, filtracji pomiarów, przez budowę i strojenie prostych regulatorów konwencjonalnych takich jak PID czy LQ, a skończywszy na inteligentnych metodach sterowania opartych np. o sztuczne sieci neuronowe.

W ramach prowadzonych zajęć studenci opracowują model matematyczny stanowiska, a następnie prowadzą jego identyfikację parametryczną. Kolejnym krokiem jest budowa, strojenie i weryfikacja algorytmów pomiarowo-sterujących dla modelu matematycznego urządzenia. Na tym etapie wykorzystuje się symulacje komputerowe. Następnie generuje się aplikację czasu rzeczywistego na podstawie

schematu zaprojektowanego algorytmu pomiarowo-sterującego w celu weryfikacji algorytmów na rzeczywistym urządzeniu. Dzięki monitorowaniu on-line pracy układu studenci obserwują osiągnięte rezultaty oraz porównują różne strategie pomiarowe i sterujące. Na każdym z powyższych etapów korzysta się z oprogramowania MATLAB/Simulink.

## 5.2. Suwnica bramowa

Laboratoryjna suwnica jest odpowiednikiem przemysłowego rozwiązania wykonanego w skali zmniejszonej (fragment suwnicy z ładunkiem na pierwszym planie przedstawia rysunek 5.). Urządzenie posiada 3 napędy pozwalające na ruch w płaszczyźnie XY oraz podnoszenie i opuszczanie ładunku. Zamontowane 5 czujników położenia informuje o lokalizacji ładunku oraz o jego kącie wychylenia. Dzięki takiej konstrukcji możliwe są zaawansowane badania nad mechaniką ruchu, optymalizacją trajektorii czy sterowaniem adaptacyjnym.

Podobnie jak w stanowisku serwomechanizmu, suwnica jest wyposażona w komputer PC wraz z kartą pomiarowo-sterującą i oprogramowaniem umożliwiając szybkie prototypowanie algorytmów.



Rys. 5. Suwnica bramowa

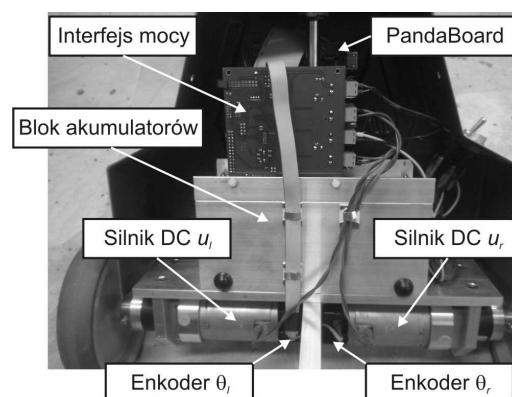
## 5.3. Robot balansujący

Robot balansujący jest przykładem mobilnego, autonomicznego pojazdu o interesującej, z punktu widzenia teorii sterowania, dynamice. Ze względu na dwa koła, zamontowane w jednej osi, robot jest z natury niestabilny, do utrzymania pionowej postawy wymaga regulatora nieprzerwanie korygującego jego położenie. Jedną z realizacji robota balansującego jest NXTway-GS, zbudowany z zestawu klocków LEGO Mindstorms (rys. 6). Do realizacji algorytmu sterującego dla ww. rozwiązania użyto oprogramowania MATLAB/Simulink i projektowania algorytmów korzystając z modelowania - MBD. Dzięki użyciu zestawu klocków LEGO istnieje możliwość wprowadzenia łatwych modyfikacji w konstrukcji fizycznej robota. Zestaw czujników znajdujący się w zestawie LEGO Mindstorms umożliwia realizację algorytmów związanych z autonomicznym poruszaniem robota, omijaniem przeszkód i podążaniem za zadanymi trajektoriami.



Rys. 6. Robot balansujący NXTway-GS [14]

Podobną konstrukcją jest robot UnTrans [15], którego system sterowania korzysta z minikomputera PandaBoard działającej pod kontrolą systemu operacyjnego Linux. Wygląd tego robota z zaznaczonymi czujnikami, elementami wykonawczymi i minikomputerem pokazano na rysunku 7. Sercem sterownika jest układ wyposażony w dwurdzeniowy procesor ARM Cortex-A9. Dzięki znacznie większej mocy obliczeniowej robot umożliwia realizację zaawansowanych algorytmów wymagający wysokiej częstotliwości próbkowania. Minikomputer oferuje również szereg dostępnych interfejsów umożliwiających łączenie dodatkowych czujników i urządzeń, np. kamery. Dzięki systemowi operacyjnemu obsługa peryferiów jest znacznie uproszczona a możliwości rozbudowy platformy niemal nieograniczone.



Rys. 7. Robot balansujący UnTrans

## 6. PODSUMOWANIE

Systemy automatyki, czy szerzej pojęte jako systemy mechatroniki, są wyposażeniem laboratoriów dydaktyczno-badawczych. Każdy z systemów jest sterowany komputerowo. Pokaz działania tych rzeczywistych modeli laboratoryjnych prowadzi się korzystając z przygotowanego oprogramowania demonstracyjnego (tzw. demo). Prawdziwa przygoda dydaktyczna a zarazem badawcza rozpoczyna się w chwili własnej ingerencji. Studenci osiągają ogromną satysfakcję widząc jak zaprojektowane przez nich algorytmy realizują założone rezultaty. Dzięki wymiernemu celowi wzrasta ich motywacja do pracy i zaangażowanie.

Zaproponowana metoda nauczania dobrze spełnia wymagania stawiane współczesnej, praktycznej edukacji na studiach inżynierskich. Co więcej, wykładowca w trakcie swego wykładu może stosować technikę przeplatania wykładu z eksperymentem rzeczywistym. Uzyskuje się to

dzięki obecnym narzędziom wizualizacyjnym i pewnemu wysiłkowi poświęconemu na przygotowanie wykładu z praktycznym doświadczeniem na wybranej platformie sprzętowo-programowej. Na ogół ten wplatany pokaz eksperymentu będzie miał charakter z góry przygotowanego przykładu demonstracyjnego, ale niekoniecznie, bowiem sprawny wykładowca, przygotowany wstępnie do pokazu ma otwarte pole do improwizacji w trakcie eksperymentu i może on-line reagować na życzenia i uwagi studentów obecnych na takim wykładzie-pokazie.

Duże znaczenie ma również aspekt praktyczny zaprezentowanej ścieżki szybkiego prototypowania. Metoda automatycznej generacji kodu postrzegana jest obecnie jako mniej podatna na błędy programistyczne w porównaniu do tradycyjnego pisania oprogramowania. W związku z powyższym inżynier posiadający kompetencje z tego zakresu ma szersze możliwości rozwoju zawodowego oraz większą wartość na rynku pracy.

## 7. BIBLIOGRAFIA

1. Kheir N.A., Åström K.J., Auslander D., Cheok K.C., Franklin G.F., Masten M., Rabins M., Control systems engineering education, *Automatica*, Volume 32, Issue 2, February 1996, Pages 147-166.
2. Kołek K., Turnau A., Hajduk K., Piątek P., Pauluk M., Marchewka D., Piłat A., Rosół M., Gorczyca P., Laboratory real-time systems to facilitate automatic control education and research, *Proceedings of the International Multiconference on Computer Science and Information Technology*, October 18–20, 2010, Wisła, Poland.
3. Balchen, J. G., M. Handlykken and A. Tyss (1981). The need for better laboratory experiments in control engineering education In *Proc. 8th IFAC Triennial World Congress*, Kyoto, Japan.
4. Charles J. Murray, Automakers Opting for Model-Based Design, [http://www.designnews.com/document.asp?doc\\_id=229640&dfpPPParams=ind\\_184](http://www.designnews.com/document.asp?doc_id=229640&dfpPPParams=ind_184).
5. Assi H.A., Engineering Education and Research Using MATLAB, *InTech*, October, 2011.
6. Horáček P., Laboratory experiments for control theory courses: A survey, *Annual Reviews in Control*, Volume 24, 2000, pp. 151-162.
7. Dormido Bencomo S., Control learning: present and future, *Annual Reviews in Control*, Volume 28, Issue 1, 2004, pp. 115-136.
8. Kołek K., Piątek K., Rapid algorithm prototyping and implementation for power quality measurement, *EURASIP Journal on Advances in Signal Processing*, 2015.
9. Inteco Sp. z o.o, RT-DAC4/PCI Multi I/O Board. Xilinx version, User's Manual, Kraków, Poland, 2012.
10. STMicroelectronics, Discovery kit for STM32F407/417 lines. User's Manual, January 2014.
11. STMicroelectronics, Code Generation for STM32 MCUs using MATLAB® and Simulink®, Technical Presentation, November 2014.
12. Aimagin Co.,Ltd., „Waijung Blockset,” [Online], Available: <http://waijung.aimagin.com>. [Data uzyskania dostępu: Luty 2016].
13. The MathWorks, Xilinx Zynq Support from MATLAB and Simulink, [Online], Available: [http://www.mathworks.com/hardware-support/zynq.html?s\\_eid=PEP\\_9385](http://www.mathworks.com/hardware-support/zynq.html?s_eid=PEP_9385). [Data uzyskania dostępu: Luty 2016].
14. Yamamoto Y., NXTway-GS Self-Balancing Two-Wheeled Robot Model-Based Design, <http://www.mathworks.com/matlabcentral/fileexchange/19147>, March 2008.
15. Gorczyca P., Knapik D., Kołek K., Rosół M., Rapid development of real-time applications for ARM Cortex A9, Design, development and implementation of real-time systems / sci. eds. Leszek Trybus, Marcin W. Mastalerz; Polish Information Processing Society. Chapter 11, Warszawa, pp. 145–155, 2013.

## EFFECTIVE ENGINEERING EDUCATION ON THE BASIS OF MECHATRONIC LABORATORY SYSTEMS AND CONTROL ALGORITHMS OBTAINED BY RAPID PROTOTYPING METHODOLOGY

The article demonstrates teaching methodology on the basis of mechatronic laboratory systems and tools for the rapid prototyping of control algorithms. The presented solution is dedicated to nowadays teaching courseware as a methodology to meet the requirements for highly trained and educated engineers. The main difficulty in engineering education is to find a balance between theory and practice. The practical (based on experiments) education requires a lot of effort. First of all, it requires access to the appropriate lab. The methodology presented in the paper is based on cost effective mechatronics systems originated from industrial plants reduced in scale. The following systems, are exemplified: gantry crane, servo and balancing robot. The signals from the sensors and signals to the actuators bring the real world into the computer model. Mechatronic systems are connected to PC computer via interface board. Computer-Aided Engineering which supports Model-Based Design techniques is used for modeling, validation and implementing measurement and control applications. Two main advantages of this solution are well visible: to make the laboratory sets cost effective, and to allow a focus on the problem while omitting the details of its implementation. The proposed method is addressed mainly to educators of the control engineering, mechanical engineering, metrology and embedded control.

**Keywords:** engineering education, mechatronic, model-based design, control laboratory.