



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Analiza systemów sterowania [S1AiR1E>PO5-ASysSter]

Przedmiot

Kierunek studiów

Automatyka i robotyka/Automatic Control and Robotics

Rok/Semestr

4/7

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

5,00

Koordynatorzy

dr hab. inż. Konrad Urbański

konrad.urbanski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Zna i rozumie na poziomie zaawansowanym teorie obwodów prądu przemiennego i stałego (w tym elektrotechnikę trójfazową). Zna i rozumie w stopniu zaawansowanym teorię i metody zasad działania podstawowych urządzeń elektronicznych, podzespołów analogowych i cyfrowych, wybranych układów i układów elektronicznych, zna i rozumie w stopniu zaawansowanym podstawowe kryteria syntezy i metod strojenia sterowników, narzędzi oraz techniki automatycznego doboru nastaw regulatorów i identyfikacji obiektów sterujących.

Cel przedmiotu

Zapoznanie studentów z metodami programowania, symulacji i analizy układów sterowania w wybranych systemach operacyjnych i środowiskach programistycznych. Nauczenie metod konfiguracji oraz podstawowych funkcji i możliwości systemu i środowiska programistycznego.

Przedmiotowe efekty uczenia się

W zakresie wiedzy:

Zna i rozumie typowe technologie inżynierskie, zasady oraz techniki konstruowania prostych systemów

automatyki i robotyki; zna i rozumie zasady doboru układów wykonawczych, jednostek obliczeniowych oraz elementów i urządzeń pomiarowo-kontrolnych [K1_W20 (P6S_WG)].

Orientuje się w aktualnym stanie oraz najnowszych trendach rozwojowych obszaru automatyki i robotyki [K1_W21 (P6S_WG)].

Zna i rozumie fundamentalne dylematy współczesnej cywilizacji powiązane z rozwojem automatyki i robotyki [K1_W28 (P6S_WK)].

W zakresie umiejętności:

Potrafi zaplanować, przygotować i przeprowadzić symulację działania prostych układów automatyki i robotyki [K1_U10 (P6S_UW)].

Potrafi dobrać rodzaj i parametry układu pomiarowego, jednostki sterującej oraz modułów peryferyjnych i komunikacyjnych dla wybranego zastosowania oraz dokonać ich integracji w postaci wynikowego systemu pomiarowo-sterującego [K1_U22 (P6S_UW)].

Potrafi opracować rozwiązanie prostego zadania inżynierskiego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym na komputerze klasy PC dla wybranych systemów operacyjnych [K1_U26 (P6S_UW)].

W zakresie kompetencji społecznych:

Posiada świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżynierskiej, w tym jej wpływ na środowisko i związaną z tym odpowiedzialność za podejmowane decyzje; jest gotów do dbałości o dorobek i tradycje zawodu [K1_K2 (P6S_KR)].

Posiada świadomość konieczności profesjonalnego podejścia do zagadnień technicznych, skrupulatnego zapoznania się z dokumentacją oraz warunkami środowiskowymi, w których urządzenia i ich elementy mogą funkcjonować; jest gotów do przestrzegania zasad etyki zawodowej i wymagania tego od innych, poszanowania różnorodności poglądów i kultur [K1_K5 (P6S_KR)].

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza zdobyta na wykładzie jest weryfikowana podczas kolokwium przeprowadzonego na ostatnim wykładzie. Studenci będą mieli dostęp do listy zagadnień obowiązujących na kolokwium. Umiejętności zdobyte w ramach laboratorium weryfikowane są na bieżąco podczas zajęć.

Treści programowe

1. Narzędzia i środowiska programistyczne dla języka Python.
2. Numeryczne modelowanie opóźnień.
3. Modelowanie opóźnienia podczas linearyzacji modelu.
4. Modelowanie numeryczne systemu pierwszego i drugiego rzędu.
5. Struktury sterujące 2DOF.
6. Struktury sterowania z predyktorem Smitha.
7. Struktury sterowania z IMC (Internal Model Control).
8. Budowa, zasada działania i zastosowania filtra Kalmana.
9. Struktury kontroli MPC (Model Predictive Control).
10. Sztuczne sieci neuronowe jako regulator.
11. Dyskretyzacja modeli.
12. Wybrane metody całkowania numerycznego.
13. Wpływ opóźnień na torze głównym i pomiarowym układu sterowania.
14. Podstawowe funkcje biblioteki opencv (computer vision).
15. Linux + Python: procedura dostępu do portu szeregowego.
16. Detektor Viola-Jones.
17. Obliczenia równoległe oraz obliczenia oparte na GPU.

Tematyka zajęć

brak

Metody dydaktyczne

Zastosowane metody szkoleniowe:

- wykład z prezentacją multimedialną (zawierającą: rysunki, fotografie, animacje, dźwięk, filmy) uzupełnioną przykładami podanymi na tablicy

- wykład prowadzony w sposób interaktywny z formułowaniem pytań do grupy studentów
 - prezentacja nowego tematu poprzedzona przypomnieniem powiązanych treści znanych studentom z innych przedmiotów
- laboratoria:
- praca w zespołach
 - eksperymenty obliczeniowe i wykonanie zadań podanych przez prowadzącego.

Literatura

Podstawowa

1. Internetowe tutoriale języka Python 3.x
2. Dokumentacja paczek języka Python
3. Dokumentacja biblioteki opencv

Uzupełniająca

1. Automatyzacja nudnych zadań z Pythonem, A. Sweigart
2. Python: wprowadzenie, M. Lutz, Helion, wydanie jak najnowsze
3. Python dla każdego. Podstawy programowania, M. Dawson
4. Control system design guide, G. Ellis, Elsevier 2004

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	120	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	60	2,50