



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Technologie inteligentnego sterowania [S2AiR2-ISA>TIS]

Przedmiot

Kierunek studiów

Automatyka i robotyka

Rok/Semestr

1/2

Studia w zakresie (specjalność)

Inteligentne systemy automatyki

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

3,00

Koordynatorzy

dr hab. inż. Konrad Urbański

konrad.urbanski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać wiedzę z automatyki i robotyki odpowiadającą 6 poziomowi Polskiej Ramy Kwalifikacji, w szczególności wiedzę z zakresu podstaw automatyki, operacji na macierzach oraz umiejętność programowania. Powinien również rozumieć konieczność poszerzania swoich kompetencji i być gotowy do podjęcia współpracy w zespole.

Cel przedmiotu

Zapoznanie studentów z metodami programowania, symulowania i analizy wybranych metod i struktur sterowania w wybranych systemach operacyjnych i środowiskach programowania. Zapoznanie z metodami konfiguracji i podstawowymi funkcjami oraz możliwościami wykorzystywanego systemu i środowisk programowania.

Przedmiotowe efekty uczenia się

Wiedza:

ma specjalistyczną wiedzę w zakresie systemów zdalnych, rozproszonych, systemów czasu rzeczywistego oraz technik sieciowych; rozumie metodykę projektowania specjalizowanych analogowych i cyfrowych systemów elektronicznych; ma uporządkowaną i pogłębioną wiedzę w zakresie specjalizowanych

systemów mikroprocesorowych przeznaczonych do układów sterowania i układów kontrolno-pomiarowych;

Umiejętności:

potrafi dobrać i zintegrować elementy specjalizowanego systemu pomiarowo-sterującego w tym: jednostkę sterującą, układ wykonawczy, układ pomiarowy oraz moduły peryferyjne i komunikacyjne; potrafi skonstruować algorytm rozwiązania złożonego i nietypowego zadania inżynierskiego i prostego problemu badawczego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym dla wybranych systemów operacyjnych; potrafi skonstruować algorytm rozwiązania złożonego i nietypowego zadania pomiarowego i obliczeniowo-sterującego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym na platformie mikroprocesorowej;

Kompetencje społeczne:

posiada świadomość konieczności profesjonalnego podejścia do zagadnień technicznych, skrupulatnego zapoznania się z dokumentacją oraz warunkami środowiskowymi, w których urządzenia i ich elementy mogą funkcjonować;

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykład: egzamin

Laboratorium: sprawdzanie umiejętności tworzenia struktur sterowania, doboru i wyznaczania parametrów modułów sterujących oraz analizy ich działania

Treści programowe

Przygotowanie i obsługa środowiska dla wykorzystywanych narzędzi badawczych

Sposoby doboru parametrów sterowników i analiza jakości działania wybranych struktur regulacji

Tematyka zajęć

Przygotowanie narzędzi programistycznych: instalacja i konfiguracja wybranego systemu opartego na Linuksie, instalacja i konfiguracja środowiska programowania dla języka Python (programy pomocnicze, moduły programowe, biblioteki: komunikacja, arytmetyczne, sterowanie, wizualizacja, inteligencja obliczeniowa, m.in. TensorFlow, itp.). Modelowanie oraz uruchamianie dostarczonych w modułach wybranych struktur regulatorów, tworzenie modeli obiektów regulacji. Uruchamianie struktur regulacji, analiza poprawności ich działania.

Zagadnienia:

dobór nastaw regulatorów wg określonych kryteriów

IMC - internal model control

SP - predyktor Smith'a

MPC - model predictive control

KF - filtr Kalmana, implementacja w układach sterowania

SSN - sztuczne sieci neuronowe - implementacja

TF, Keras i sieci głębokie - uczenie i implementacja

wpływ opóźnień w torze sterowania

sterowanie urządzeniami z użyciem j. Python

Metody dydaktyczne

-wykład z prezentacją multimedialną (w tym: rysunki, zdjęcia, animacje, dźwięk, filmy) uzupełniany przykładami podawanymi na tablicy

-wykład prowadzony w sposób interaktywny z formułowaniem pytań do grupy studentów

-przedstawianie nowego tematu poprzedzone przypomnieniem treści powiązanych, znanych studentom z innych przedmiotów

laboratoria:

-praca w zespołach

-eksperymenty obliczeniowe

Literatura

Podstawowa:

1. Internetowe tutoriale dla aktualnej wersji Pythona 3.x
2. Dokumentacja (internet) wybranych modułów języka Python dla wersji 3.x
3. Dokumentacja (internet) bibliotek Keras, TensorFlow
4. PID Controllers : Theory, Design, and Tuning, 2nd Edition, K.J. Astrom, T. Hagglund, 1995
5. Control system design guide, G. Ellis, Elsevier 2004

Uzupełniająca:

1. Automatyzacja nudnych zadań z Pythonem, A. Sweigart, wydanie jak najnowsze
2. Python: wprowadzenie, M. Lutz, Helion, wydanie jak najnowsze
3. Python dla każdego. Podstawy programowania, M. Dawson, wydanie jak najnowsze
4. Deep Learning - Praca z językiem Python i biblioteką Keras, F. Chollet, Helion 2019

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	90	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	30	1,00