



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Technologie inteligentnego sterowania

### Przedmiot

Kierunek studiów

automatyka i robotyka

Studia w zakresie (specjalność)

Inteligentne systemy automatyki

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1 / 2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

15

Laboratoria

30

Inne (np. online)

Ćwiczenia

Projekty/seminaria

### Liczba punktów ECTS

3

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Konrad Urbański

Odpowiedzialny za przedmiot/wykładowca:

email: Konrad.Urbanski@put.poznan.pl

tel. 48 61 665 2810

Wydział Automatyki, Robotyki i Elektrotechniki

ul. Piotrowo 3A 60-965 Poznań

### 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

rozumie metodykę projektowania specjalizowanych analogowych i cyfrowych systemów elektronicznych;

### Umiejętności

potrafi posługiwać się technikami informacyjno-komunikacyjnymi;

### Kompetencje społeczne

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 rozwijania dorobku zawodowego;

## 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 narzędzi programistycznych: instalacja i konfiguracja wybranego systemu opartego na linuxie, 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
łączy nakład pracy	90	3,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	1,5
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do egzaminu) <sup>1</sup>	45	1,5

<sup>1</sup> niepotrzebne skreślić lub dopisać inne czynności