



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Inżynieria oprogramowania w robotyce [N2AiR1-SSiR>IOwR]

Przedmiot

Kierunek studiów

Automatyka i robotyka

Rok/Semestr

1/1

Studia w zakresie (specjalność)

Systemy sterowania i robotyki

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

niestacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

10

Laboratorium

20

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

3,00

Koordynatorzy

dr hab. inż. Wojciech Kowalczyk

wojciech.kowalczyk@put.poznan.pl

dr inż. Bartłomiej Krysiak

bartlomiej.krysiak@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu projektowania układów robotycznych, analizy i implementacji układów regulacji automatycznej w kontekście podstawowych zadań sterowania, planowania i tworzenia map przestrzennych środowiska. Student powinien posiadać wiedzę na temat budowy, wykorzystania i własności regulatorów, umiejętność opisu dynamiki systemów w przestrzeni stanu i w postaci wejściowo-wyjściowej. Student powinien posiadać umiejętności wykorzystania języka programowania C++ na poziomie podstawowych z wykorzystaniem technik obiektowych. Ponadto student powinien umieć korzystać z podstawowych technik informacyjno-komunikacyjnych, umieć pozyskiwać informacje ze wskazanych źródeł, a także powinien wykazywać gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

Zapoznanie się z metodologią projektowania systemów informatycznych wykorzystywanych w sterowaniu i zarządzaniu robotami z wykorzystaniem wybranych platform programistycznych. W zakresie przedmiotu zostaną omówione zagadnienia związane z sterowaniem robotów w zakresie komunikacji, zarządzania i synchronizacji procesów, implementacji układów regulacji oraz modeli obiektów. Ponadto w ramach przedmiotu będą poruszane zagadnienia dotyczące przetwarzania danych sensorycznych w tym zagadnienia filtrowania i fuzji. W dalszej kolejności zostaną przedstawione zagadnienia wizualizacji środowiska wraz z jego obiektami w celu weryfikacji poprawności implementacji oprogramowania. Student będzie zdobywał umiejętność świadomego i konstruktywnego wykorzystania poznanych metod projektowych do różnych zadań w robotyce.

Przedmiotowe efekty uczenia się

Wiedza

1. Podstawowa znajomość wybranej architektury środowiska programistycznego wspierającego implementację oprogramowania dla robotów. Przegląd podstawowych bibliotek i narzędzi umożliwiających rozwój oprogramowania do aplikacji robotycznych. Ogólny schemat zarządzania oprogramowaniem do aplikacji robotycznych w ramach wybranych bibliotek i programów ramowych zarządzających pracą systemów robotycznych. [K2_W10]
2. Znajomość podstawowych technik akwizycji i wizualizacji danych w oprogramowaniu wykorzystywanym w aplikacjach robotycznych. Znajomość podstawowych technik przetwarzania tych danych do wnioskowania poprawności pracy układów robotycznych. [K2_W10]
3. Znajomość technik zarządzania oprogramowaniem w kontekście wersjonowania oprogramowania, archiwizacji i zespołowego implementacji systemów informatycznych w robotyce. [K2_W3], [K2_W12]
4. Podstawowe zagadnienia dotyczące komunikacji pomiędzy jednostkami przetwarzania danych w kontekście współpracy wielu układów robotycznych. [K2_W3]

Umiejętności

1. Umiejętność wykorzystania wybranej platformy programistycznej w aplikacjach robotycznych. [K2_U2], [K2_U22]
2. Potrafi dokonać wyboru odpowiednich bibliotek oprogramowania stosowanie do rozważanych zadań w sterowaniu robotów. [K2_U1]
3. Potrafi wykorzystać architekturę zarządzania modułami programistycznymi w ramach wybranych programów ramowych wykorzystywanych w oprogramowaniu robotów. [K2_U12]
4. Umie zaimplementować proste zadania programistyczne dla wybranych procesów realizowanych w sterowaniu robotów. [K2_U9], [K2_U24], [K2_U25]
5. Potrafi zarządzać oprogramowaniem w kontekście archiwizacji i wersjonowania. [K2_U9], [K2_U24]

Kompetencje społeczne

1. Umiejętność pracy w zespole oraz świadomość pozatechnicznych skutków podejmowanych decyzji projektowych w obszarze automatycznych układów regulacji. [K2_K2], [K2_K3]
2. Świadomość konieczności profesjonalnego podejścia do zagadnień technicznych i krytycznej oceny dokonywanych wyborów projektowych. [K2_K5], [K2_K1]

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:

- A) W zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest poprzez ocenę wiedzy studentów podczas pisemnego zaliczenia końcowego w formie pisemnej.
- B) W zakresie zajęć laboratoryjnych weryfikowanie założonych efektów kształcenia realizowane jest poprzez bieżącą ocenę postępów w nauce, a także przez końcową ocenę jakości działania zaprojektowanego oprogramowania, ocenę pisemnego raportu z realizacji zadań, a także ocenę odpowiedzi na pytania merytoryczne związane z wykonanym zadaniem.

Treści programowe

Program przedmiotu obejmuje następujące zagadnienia: architektura oprogramowania ramowego wraz z podziałem na moduły programistyczne, porty wymiany danych, wiadomości komunikacyjne i parametry inicjalizacyjne; zarządzanie implementacją z poziomu linii komend wybranego środowiska programistycznego; tworzenie pakietów programistycznych z wykorzystaniem gotowych formatów

projektowych oraz skryptów uruchamiających pakiety programowe; wykorzystanie wybranych bibliotek programistycznych do tworzenia oprogramowania; symulacja i wizualizacja procesów w sterowaniu robotami; wykorzystanie bibliotek obliczeniowych do transformacji geometrycznych w zadaniach obliczania położenia, prędkości i przyspieszeń robotów; integracja wybranych modułów programistycznych w celu realizacji złożonych zadań w robotyce.

Metody dydaktyczne

A) Wykłady: prezentacja multimedialna (slajdy) dodatkowo ilustrowana przykładami podawanymi i analizowanymi na tablicy.

B) Ćwiczenia laboratoryjne: wykonywanie zadań programistycznie w tematyce podanym przez prowadzącego.

Literatura

Podstawowa

[1] Control system design, G. C. Goodwin, S. F. Graebe, M. E. Salgado, Prentice Hall, 2001

[2] Programowanie. Teoria i praktyka z wykorzystaniem C++, Bjarne Stroustrup, Helion, 2010

[3] Programming Robots with ROS, William D. Smart, O'Reilly Media, Inc, USA, 2015.

Uzupełniająca

[4] Introduction to Ubuntu Linux, Vimal Kumar v., Vimal Kumar V., 2020

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	45	1,00