



**AGH**

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,  
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA AUTOMATYKI

Praca dyplomowa magisterska

*Algorytm sterowania wykorzystujący sztuczne sieci neuronowe dla  
bezzałogowego statku latającego typu TRICOPTER*

Autor:

*Rafał Włodarz*

Kierunek studiów:

*Automatyka i robotyka*

Opiekun pracy:

*dr hab. Adam Piłat*

Kraków, 2015

*Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.*

*Serdecznie dziękuję ... tu ciąg dalszych podziękowań np. dla promotora, żony, sąsiada itp.*



## Spis treści

<b>1. Wstęp</b>	7
1.1. Cele pracy	7
1.2. Zawartość pracy	7
<b>2. Sztuczne sieci neuronowe</b>	9
2.1. Cele pracy	9
<b>3. Architektura statku latającego typu tricopter</b>	11
3.1. Konstrukcja tricopter-a	11
3.2. Budowa modelu	11
<b>4. Aplikacja sterująca</b>	13
4.1. Konfiguracja beaglebone black	13
4.1.1. System czasu rzeczywistego	13
4.1.2. Przygotowanie systemu operacyjnego	14
4.1.3. Analiza operacji zmiennoprzecinkowych	14
4.2. Architektura systemu sterującego	14
<b>5. Testy systemu sterującego</b>	15
<b>6. Podsumowanie</b>	17



# **1. Wstęp**

## **1.1. Cele pracy**

## **1.2. Zawartość pracy**





## **2. Sztuczne sieci neuronowe**

Rozdział ten zawiera informacje na temat sieci neuronowych, ich architektury, zasady działania oraz algorytmów uczenia.

### **2.1. Cele pracy**



### **3. Architektura statku latającego typu tricopter**

Poniższy rozdział przedstawia zbiór podstawowych zagadnień związanych z konstrukcją wirnikowca typu tricoper oraz zawiera informacje na temat zasad sterowania układem.

#### **3.1. Konstrukcja tricopter-a**

#### **3.2. Budowa modelu**



## 4. Aplikacja sterująca

W niniejszym rozdziale przedstawiono informacje na temat aplikacji sterującej. Zaprezentowano kolejne etapy przygotowania platformy sprzętowej, które wymagane są do weryfikacji działania systemu czasu rzeczywistego o ostrych ograniczeniach czasowych (ang. hard real-time).

### 4.1. Konfiguracja beaglebone black

#### 4.1.1. System czasu rzeczywistego

System czasu rzeczywistego (ang. real-time system) to system, który przetwarza każdy rodzaj informacji i który musi reagować na sygnały wejściowe - bodźce generowane z zewnątrz w skończonym i określonym czasie. Jego poprawne działanie zależy zarówno od prawidłowych rezultatów logicznych, jak również od czasu reakcji. Na podstawie tych kryteriów są one dzielone na:

- Systemy o ostrych wymaganiach czasowych (ang. hard real-time) - wymagania czasowe muszą być skrupulatnie przestrzegane, naruszenie ram czasowych może wpłynąć na życie ludzkie, środowisko czy też sam system,
- Systemy o słabych wymaganiach czasowych (ang. soft real-time) – głównym kryterium oceny tych jest średni czas odpowiedzi. Sporadyczne opóźnienie nie powoduje zagrożenia lecz jedynie wpływa negatywnie na ocenę całego systemu,
- Systemy o solidnych wymaganiach czasowych (ang. firm real-time) - są one kombinacją systemów o wymaganiach ostrych oraz słabych. Naruszenie kryterium czasowych może pojawiać się okazjonalnie. Często dla lepszej oceny systemu stosuje się ograniczenia czasowe o charakterze słabym- krótsze, których przekroczenie nie powoduje katastrofy oraz ostrym- dłuższe, których naruszenie oznacza nieprawidłowe działanie systemu.

Bez względu na to które z powyższych kryteriów są spełniane przez system konieczne jest, aby każdy z nich charakteryzował się następującymi cechami:

- Ciągłość działania - powinny działać nieprzerwanie w okresie od uruchomienia systemu do jego wycofania,

- Zależność od otoczenia - zachowanie opiera rozpatruje się w kontekście otoczenia. Prowadzone obliczenia zależą od zdarzeń oraz danych pochodzących z zewnątrz układu,
- Współbieżność - struktura systemu narzuca, aby jednocześnie zdarzenia były obsługiwane równocześnie przez szereg procesów,
- Przewidywalność - zdarzenia i dane generowane przez otoczenie pojawiają się przypadkowo co nie narusza deterministycznego zachowania systemu,
- Punktualność - odpowiedź systemu na bodźce zewnętrzne powinna być dostarczona w odpowiednich momentach - wymaganych ramach czasowych.

#### **4.1.2. Przygotowanie systemu operacyjnego**

Ze względu na wcześniej wspomnianą specyfikę systemu sterującego oraz zastosowanie platformy sprzętowej typu mini PC wraz z systemem operacyjnym typu UNIX, ważne jest, aby wyeliminować wszelkie możliwe przerwania oraz inne aspekty, które wpływają na płynność oraz czas wykonywania się aplikacji sterującej.

#### **4.1.3. Analiza operacji zmiennoprzecinkowych**

### **4.2. Architektura systemu sterującego**

## **5. Testy systemu sterującego**





## **6. Podsumowanie**



## **Bibliografia**