

Seminarium dyplomowe

Manipulator przemysłowy ze sprzężeniem wizyjnym

Promotor: Dr inż. Marcin Kiełczewski

Bartosz Babiaczyk
Krystian Borowicz
Wojciech Mila

Opis pracy

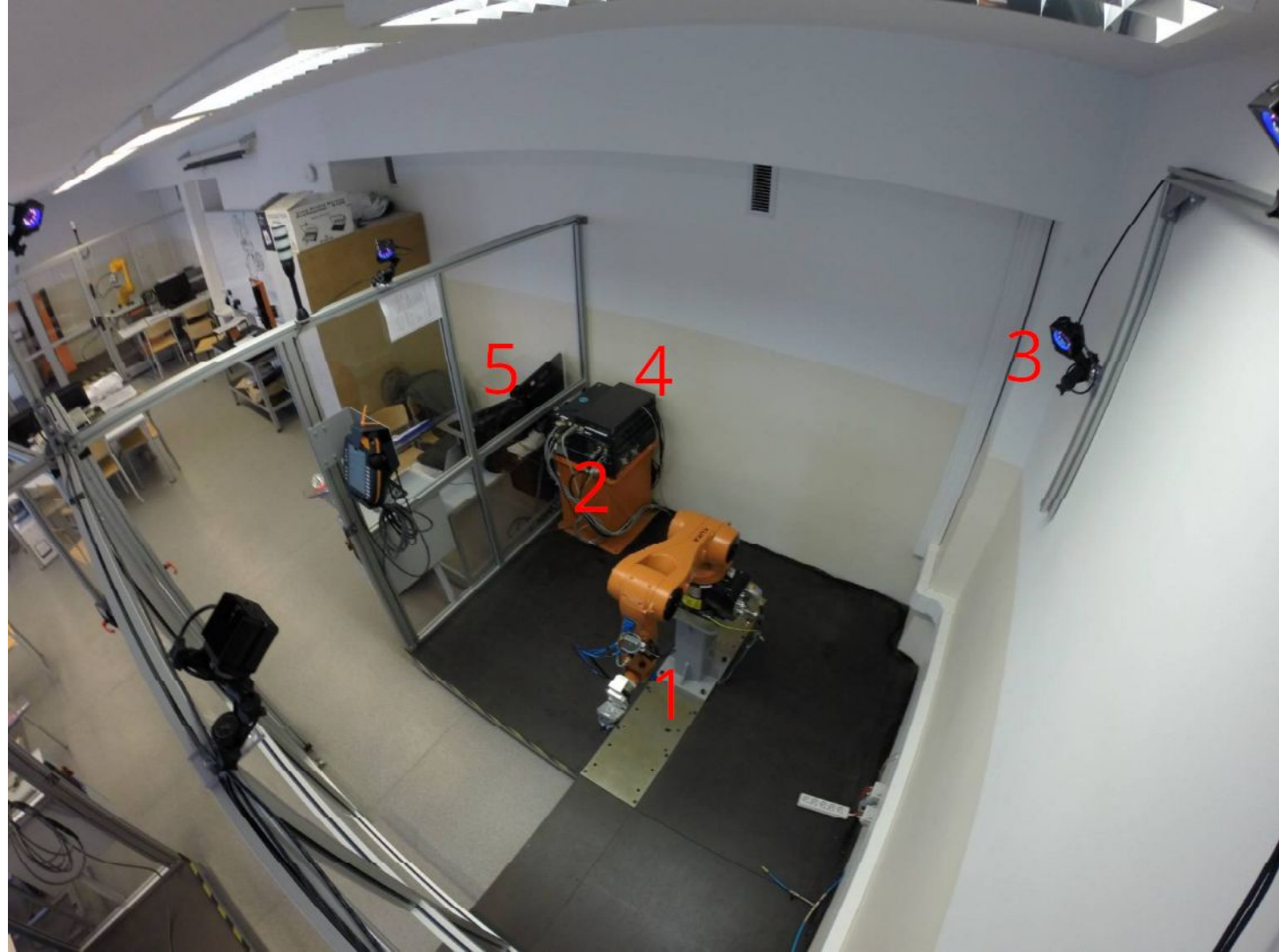
Przygotowanie stanowiska z robotem przemysłowym do realizacji wybranych zadań manipulacyjnych z wykorzystaniem informacji o lokalizacji obiektów z systemu wizyjnego.

Celem pracy jest integracja stanowisk dwóch manipulatorów przemysłowych oraz systemu wizyjnego firmy OptiTrack.

Opis stanowiska

Stanowisko składa się z 2 cel z robotami firmy KUKA Agilus KR6 R900 sixx oraz szafami sterowniczymi KRC4 Compact. Sprzężenie wizyjne zapewnione jest przez system OptiTrack, składający się z 6 kamer, wykorzystujących pasmo światła podczerwonego. Centralnym punktem jest komputer PC (8GB RAM DDR3, intel core i5-4690k 3.5GHz), do którego podłączony jest system wizyjny oraz szafy sterownicze robotów.





5

4

2

3

1

Pierwsze kroki

Pierwsze kroki polegały na wykonaniu kilku ćwiczeń, programując robota z poziomu panelu operatorskiego KRC (Kuka Robot Controller). Umożliwiło to lepsze zrozumienie działania manipulatora oraz języka KRL (Kuka Robot Language).

Następnie przetestowaliśmy kable ethernetowe oraz podłączyliśmy urządzenia tworząc sieć:

- PC - KUKA 1

- PC - KUKA 2

- PC - System wizyjny OptiTrack

Połączenie PC - manipulator



* Dane z systemu wizyjnego



- * Pozycja w układzie kartezjańskim
- * Obrót osi A1 - A6
- * Znacznik czasowy

4 ms

- * Korekta wyznaczona na podstawie danych z systemu wizyjnego
- * Skopiowany znacznik czasowy



Konfiguracja RSI - odbierane / wysłane ramki

Konfiguracja ramki wysyłanej przez robota

```
<SEND>
  <ELEMENTS>
    <ELEMENT TAG="DEF_RIst" TYPE="DOUBLE" INDX="INTERNAL" />
    <ELEMENT TAG="DEF_AIPos" TYPE="DOUBLE" INDX="INTERNAL" />
  </ELEMENTS>
</SEND>
```

Ramka wysłana z robota do komputera

```
<Rob TYPE="KUKA">
  <RIst X="50.342" Y="502.443" Z="121.422" A="0.0" B="0.0" C="0.0" />
  <AIPos A1="0.0" A2="0.0" A3="0.0" A4="0.0" A5="0.0" A6="0.0" />
  <IPOC>123645634563</IPOC>
</Rob>
```

Konfiguracja ramki odbieranej przez robota

```
<RECEIVE>
  <ELEMENTS>
    <ELEMENT TAG="RKorr.X" TYPE="DOUBLE" INDX="1" HOLDON="1" />
    <ELEMENT TAG="RKorr.Y" TYPE="DOUBLE" INDX="2" HOLDON="1" />
    <ELEMENT TAG="RKorr.Z" TYPE="DOUBLE" INDX="3" HOLDON="1" />
    <ELEMENT TAG="RKorr.A" TYPE="DOUBLE" INDX="4" HOLDON="1" />
    <ELEMENT TAG="RKorr.B" TYPE="DOUBLE" INDX="5" HOLDON="1" />
    <ELEMENT TAG="RKorr.C" TYPE="DOUBLE" INDX="6" HOLDON="1" />
  </ELEMENTS>
</RECEIVE>
```

Ramka wysyłana z komputera do robota

```
<Sen Type="ImFree">
  <RKorr X="1.0" Y="0.0" Z="0.0" A="0.0" B="0.0" C="0.0" />
  <IPOC>123645634563</IPOC>
</Sen>
```

Rdzeń systemu - aplikacja C#

Kluczowym elementem pracy jest aplikacja uruchomiona na komputerze PC, zarządzająca wszystkimi podłączonymi urządzeniami. Zastosowanym językiem programowania jest C#, w środowisku programistycznym Visual Studio 2019 Community.

Aplikacja podzielona jest ze względu na funkcjonalność na cztery wątki:

Wątek 1	Odbieranie, przetwarzanie i wysyłanie danych do robota 1
Wątek 2	Odbieranie, przetwarzanie i wysyłanie danych do robota 2
Wątek 3	Odbieranie, przetwarzanie danych z systemu wizyjnego
Wątek 4	Interfejs użytkownika oraz jego aktualizacja

Interfejs graficzny aplikacji

RSI connection test

X+ X-

Y+ Y-

Z+ Z-

A+ A-

B+ B-

C- C+

X

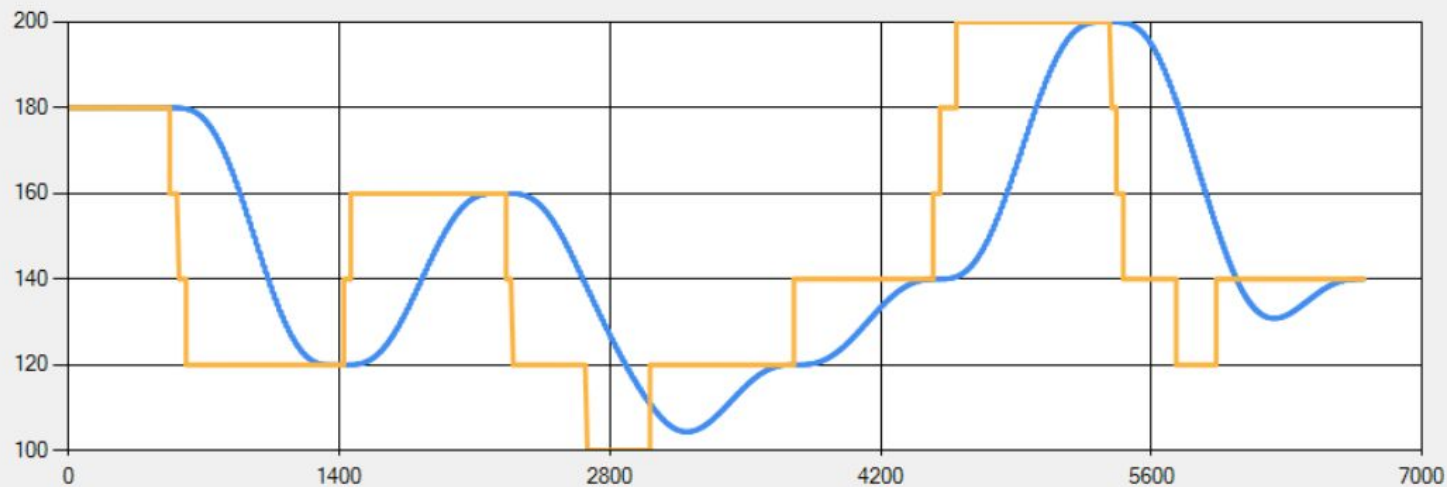
Y

Z

A

B

C



Generator trajektorii ruchu

Pozycja robota wysyłana jest z programu działającego na komputerze, dlatego **to my** odpowiadamy za to, w jaki sposób poruszy się robot (zdefiniowanie prędkości i programowe zabezpieczenie przed jakąkolwiek kolizją również musi być zrealizowane przez nas).

W celu wykonania płynnego ruchu program co 4ms musi wysłać informację o korekcji pozycji manipulatora. Składa się ona z:

X, Y, Z - korekcji **położenia** końcówki manipulatora

C, B, A - korekcji **orientacji** końcówki manipulatora

(w globalnym układzie współrzędnych)

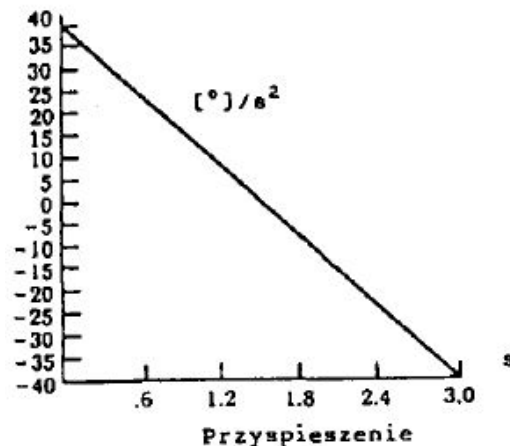
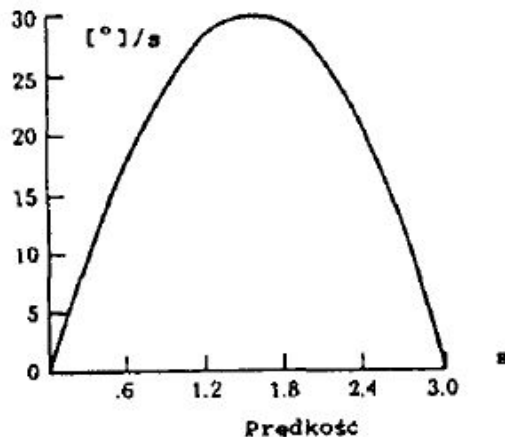
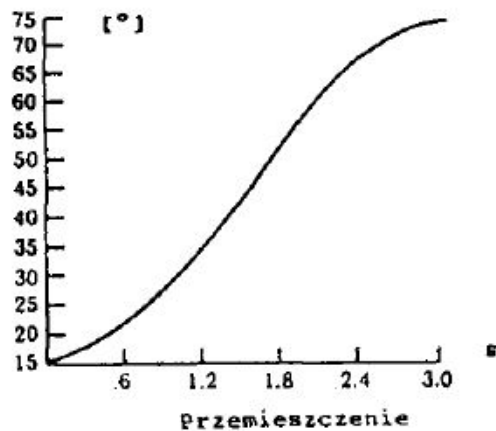
Generator trajektorii ruchu

Aby zachować ciągłość przyspieszenia, przyjęto model położenia robota jako **wielomian III stopnia**:

$$x = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$v = x' = 3a_3 t^2 + 2a_2 t + a_1$$

$$a = x'' = 6a_3 t + 2a_2$$



Generator trajektorii ruchu

W każdej iteracji, czyli co 4ms, program sprawdza, czy roboty znajdują się w zadanej pozycji. Jeżeli tak nie jest, obliczane są współczynniki wielomianu:

$$a_0 = x_0$$

$$a_1 = v$$

$$a_2 = (3(x_k - x_0) - 2vT) / T^2$$

$$a_3 = (vT - 2(x_k - x_0)) / T^3$$

Oraz prędkość:

$$v = 3a_3t^2 + 2a_2t + a_1$$

Gdzie: x_0 - pozycja aktualna, x_k - pozycja docelowa, v - prędkość aktualna,
 T - czas trwania ruchu do punktu docelowego, t - czas iteracji, 4ms

X+	X-
Y+	Y-
Z+	Z-
A+	A-
B+	B-
C-	C+

X

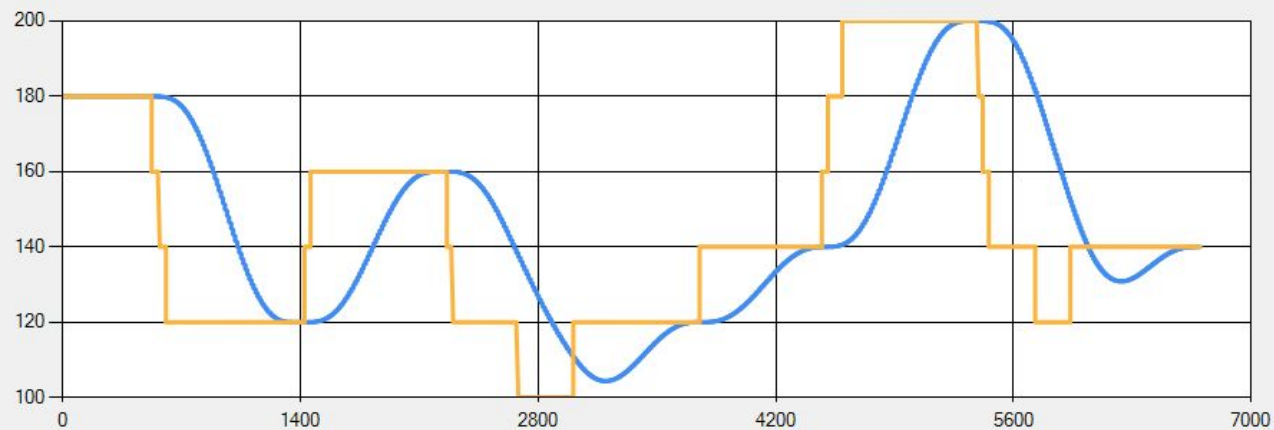
Y

Z

A

B

C



X+	X-
Y+	Y-
Z+	Z-
A+	A-
B+	B-
C+	C-

X

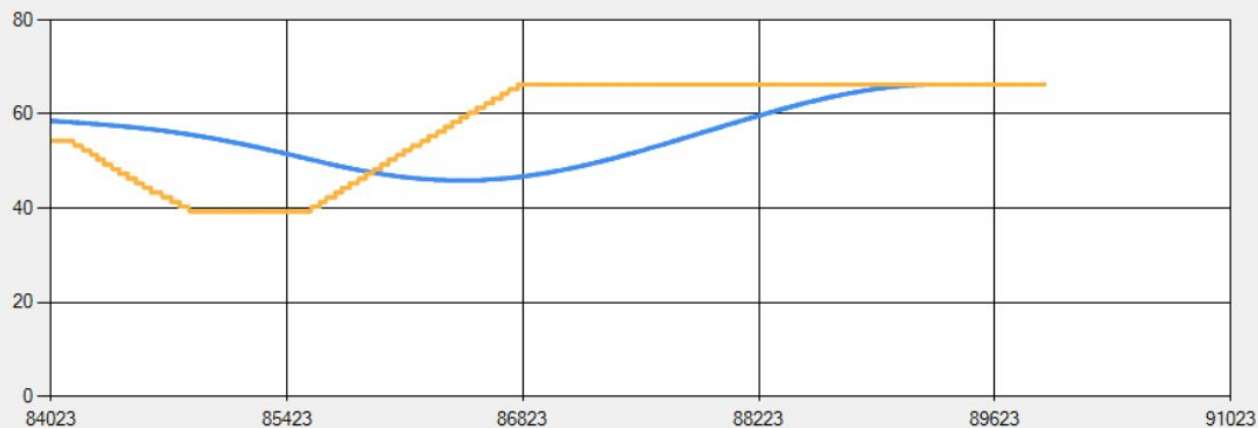
Y

Z

A

B

C



Motive - OptiTrack

Motive jest to program służący do obsługi OptiTracka.

Umożliwia nam zastosowanie kalibracji w celu uzyskania danych dotyczących orientacji oraz położenia kamer.

Udostępnia także możliwość ustawienia podstawowych parametrów kamery:

- THR - jasność odbitych promieni podczerwonych które będą zauważane przez kamerę,
- FPS - częstotliwość pracy kamery,
- Gain - jasność obrazu kamery



Połączenie z OptiTrackiem - NatNet

Połączenie umożliwia nam wcześniej wspomniany program Motive.

Wykonywane są w nim wszelkie potrzebne obliczenia w celu uzyskania pozycji obiektu.

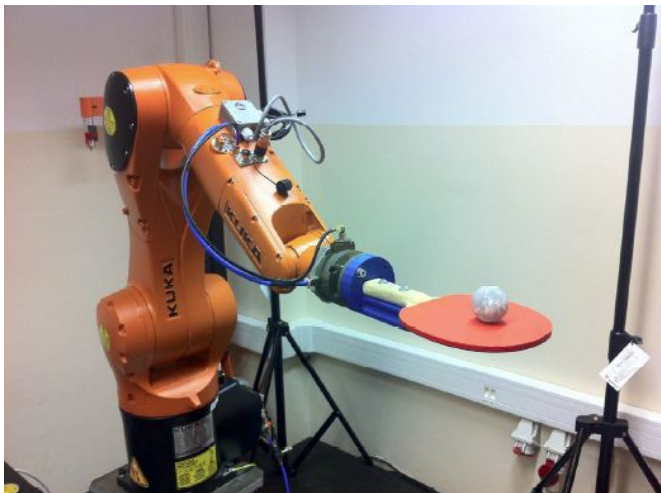
Aby uzyskać ramkę z danymi obiektu wykorzystamy w naszym programie bibliotekę NatNetML.dll.

Kolejne kroki

- Opracowanie macierzy przekształcenia z układu systemu wizyjnego, do globalnego układu robota,
- **Gra w Ping-Ponga:** naprzemienne podbijanie piłeczki przez 2 roboty
 - Przygotowanie paletki
 - Przygotowanie piłeczki
 - Obliczenie trajektorii lotu piłeczki
 - Opracowanie algorytmu odbijania piłeczki, tak by wylądowała w zadanym miejscu

Inspiracja

Nasza praca jest rozwinięciem tematu realizowanego w 2017 roku pt: “Sterowanie robotem manipulacyjnym ze sprzężeniem wizyjnym”, którego efektem końcowym było podbijanie piłeczki przez robota kuka. Ten temat z kolei zainspirowany został reklamą firmy kuka



Bibliografia

John J. Craig *Wprowadzenie do robotyki, mechanika i sterowanie*. Wydawnictwo Naukowo-Techniczne, 1995

Jan Wiktor Meissner, Wojciech Pazda *Sterowanie robotem manipulacyjnym ze sprzężeniem wizyjnym*, praca dyplomowa - inżynierska