



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

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

KATEDRA INFORMATYKI AUTOMATYKI I INŻYNIERII BIOMEDYCZNEJ

Praca dyplomowa magisterska

Wykorzystanie systemu wizyjnego w pozycjonowaniu robota IRp-6.

Using vision system with positioning IRp-6 robot.

Autor:

Maciej Podsiadło

Kierunek studiów:

Automatyka i Robotyka

Opiekun pracy:

dr Mieczysław Zaczek

Kraków, 2014

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

1. Struktura i sterowanie robotem.....	7
1.1. Stanowisko.....	7
1.1.1. Robot przemysłowy	7
1.1.2. Sterowanie.....	8
1.1.3. System wizyjny	9
1.1.4. Inne.....	9
1.2. Model sterownika w simulinku	10
1.2.1. Serwomechanizm	10
1.2.2. Stop	11
1.2.3. Sterowanie prędkością.....	11
1.2.4. Koordynacja ruchowa.....	12
1.2.5. Bazowanie	12
1.3. Proste zadanie kinematyki	13
1.4. Odwrotne zadanie kinematyki	16

1. Struktura i sterowanie robotem

1.1. Stanowisko

1.1.1. Robot przemysłowy

Producent: Zakłady Automatyki Przemysłowej Ostrów Wielkopolski

Model: IRp-6

Parametry:

Osie sterowane:		5
Kontrolery:		USR-6
Maksymalne obciążenie:		6 kg
Powtarzalność:		+0,20 mm
Masa jednostki mechanicznej:		125 kg
Zasięg:		
Zakres ruchu:	Oś1:	+ - 160
	Oś2:	+ - 40
	Oś3:	-25..+40
	Oś4:	-25..+120
	Oś5:	-25..+150
Prędkość maksymalna:	Oś1:	60
	Oś2:	60
	Oś3:	60
	Oś4:	75
	Oś5:	125

Podstawa robota została umieszczona na napędzie liniowym, w celu poszerzenia obszaru roboczego, tak aby pokrywał się on z polem widzenia systemu wizyjnego.

Parametry:

Napęd:	Elektryczny napęd liniowy ze śrubą
Wielkość:	
Skok [mm]:	
Maks. siła podawania [N]:	
Moment M_y/M_z [Nm]:	
Moment M_x [Nm]	
Prędkość [m/s]:	

1.1.2. Sterowanie

Sterownie robotem odbywało się przez system modułowy dSpace DS1005. Komputer PC połączony z tym systemem przez magistralę AT-bus, pełnił rolę prototypowania sterownika. W programie Matlab/Simulink utworzony został sterownik robota, który przekonwertowany na kod źródłowy w języku C przez narzędzie RTW (Real-Time Workshop), został skompilowany, zlinkowany oraz wgrany do karty procesowej DS1005 z procesorem sygnałowym za pomocą modułu RTI (Real-Time Interface). Ponadto uruchomiony w komputerze program ControlDesk, był interface'm graficznym między sterownikiem a użytkownikiem, działającym w czasie pracy robota. Za pomocą tego programu wprowadzano potrzebne parametry do odpowiedniej pracy robota.

Komputer klasy PC

System operacyjny: Windows 2000 Serwer

Procesor:

Pamięć RAM:

Oprogramowanie:

- Matlab 2007b z bibliotekami: Simulink (służący do prototypowania sterownika), Image Processing Toolbox (do przetwarzania obrazu) oraz RTW (generacji kodu źródłowego z modelu Simulink) ;
- Program ControlDesk - interface graficzny między operatorem, a sterownikiem;
- Moduł RTI (Real-Time Interface) - służący do komunikacji z systemem dSpace.

System dSpace DS 1005 zawierał następujące karty:

- karta Master DS1005 - główna jednostka obliczeniowa
- karta Multi I/O DS2201 - wielofunkcyjna karta pomiarowa
- karta wyjść analogowych DS2103
- karta DS3001 - zliczanie impulsów z enkoderów

1.1.3. System wizyjny

Producent:

Model:

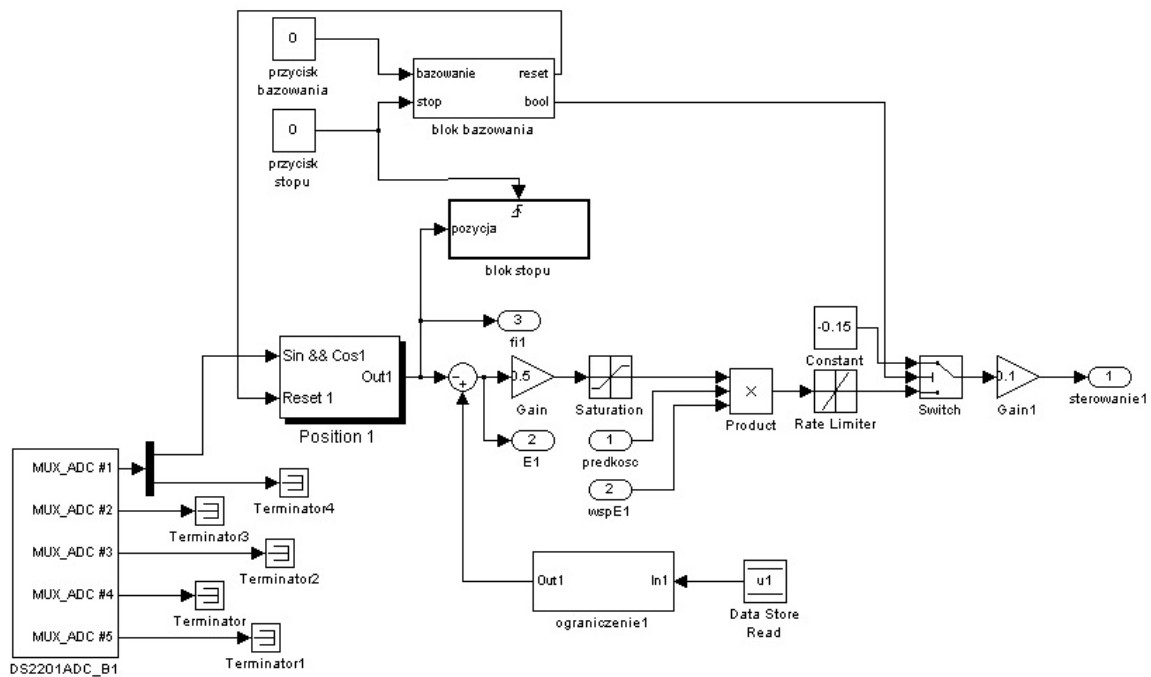
Parametry:

Rozdzielczość:	
Szybkość:	
Ilość klatek na sekundę:	
Pole widzenia:	
Formaty obrazu:	
Oświetlenie:	brak

1.1.4. Inne

Stół o wymiarach:

Wysokość:	0,72 m
Szerokość:	0,8 m
Długość:	1,2 m



Rysunek 1.1: Schemat zamkniętego układu redukcji

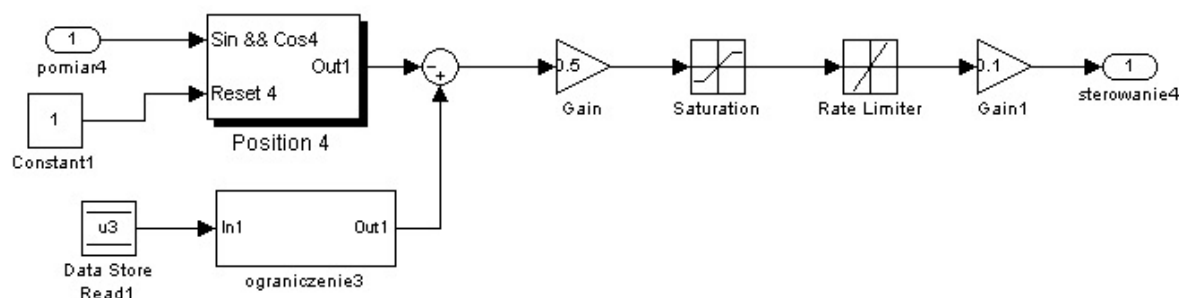
1.2. Model sterownika w simulinku

1.2.1. Serwomechanizm

Sterowanie napędów zostało zrealizowane przez zamknięty układ sterowania, przedstawionym na poniższym schemacie.

W badanym przypadku zamknięcie sprzężenia zwrotnego odbywało się poprzez odczyt położenia kątownego z czujników umieszczonych w złączu. Dane te zostały odczytane przez moduł dSpace, a następnie poprzez blok DS2201ADC_B1 zostały przekazane do modelu. Do sterowania wykorzystano regulator proporcjonalny. Ponieważ dla tego robota były realizowane zadania nadążania i przestawiania, nie zastosowano członu całkującego, który zmniejszyłby zapas stabilności. Wartość współczynnika wzmocnienia została ustalona na podstawie przeprowadzonych eksperymentów opisanych w pozycji [?]. Sygnał sterujący był odpowiednio przetworzony przez następujące operacje:

- ograniczenie sygnału (blok saturation) - silnik jak każdy obiekt rzeczywisty miał ograniczone wartości sterowania.
- zamiana uchybu położenia na prędkość (blok mnożenia) - w zespole napędowym każdego złącza był silnik DC, w którym można sterować jego prędkością.
- ograniczenie narastania sygnału (blok rate-limiter) - nie można było dopuścić do skokowej zmiany prędkości silnika, która mogłaby go uszkodzić.



Rysunek 1.2: Schemat serwomechanizmu bez dodatków

- normalizacja (blok gain2) - karta DS2201 wymagała sygnału z przedziału (-0.3; 0.3)

Dodatkowo wartość zadana była ograniczana (blok ograniczenia..), ze względu na zakresy ruchu każdej osi podane w rozdziale 1.1.1. Blok "Position .." konwertuje sygnał z czujnika na położenie podane w stopniach.

Parametry wejściowe subsystemu:

- "pomiar.." - sygnał z czujnika położenia
- "predkosc" - zadana prędkość podana w %

Parametry wejściowe subsystemu:

- "sterowanie.." - sygnał sterujący silnikiem w danym złączu

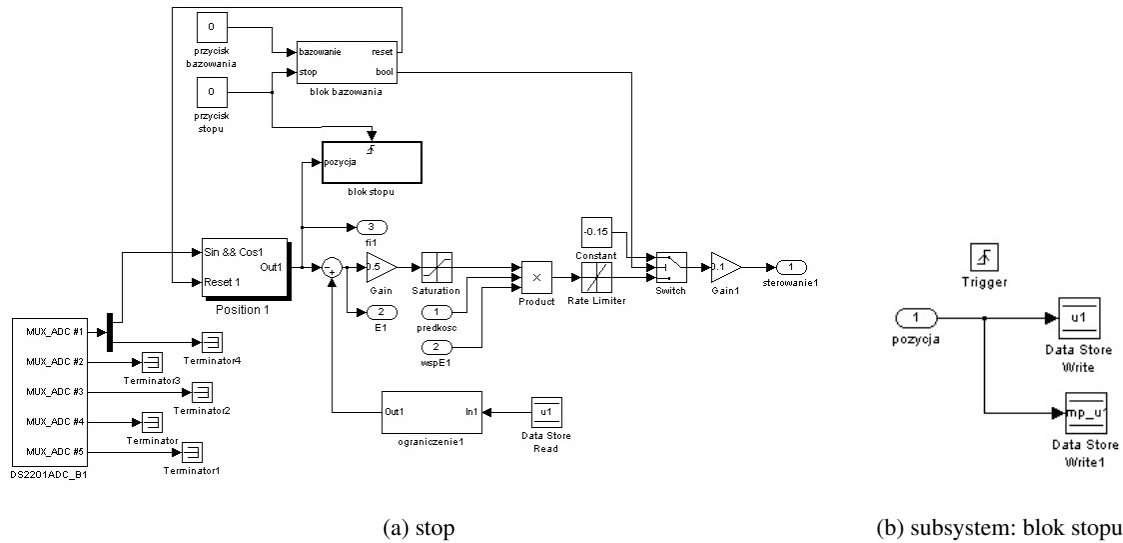
1.2.2. Stop

Zatrzymanie robota w dowolnym momencie było niezbędną częścią sterownika. Ta funkcja umożliwiała uniknięcie uszkodzenia robota lub innych przedmiotów w obszarze roboczym w przypadku awarii lub błędu operatora. To zadanie zrealizowano poprzez jednokrotne przypisanie aktualnego położenia do pozycji zadanej przez co uchyb ustalony zmniejszał się do zera i w efekcie sterowanie także się redukowało. W przypadku zatrzymania nie można było przypisać wprost zera do sterowania, ponieważ przerwałoby się pętlę sprzężenia zwrotnego co skutkowałoby utratą kontroli.

Jednorazowe przypisanie do wartości zadanej zrealizowano poprzez subsystem wyzwalany zboczem rosnącym.

1.2.3. Sterowanie prędkością

Operator mógł sterować prędkością poprzez zadawanie jaka część szybkości maksymalnej miałbyć wykorzystany. Wybrany procent wybranej prędkości był mnożony przez wartość sterowania regulatora. Nie można było podać do toru sterowania konkretnej szybkości, ponieważ skutkowałoby nieprawidłowym działaniem regulatora.



Rysunek 1.3: Mechanizm stopu

1.2.4. Koordynacja ruchowa

W sterowniku zaimplementowano koordynację ruchową (quasiliniowo). Celem jej działania było, żeby wszystkie zespoły napędowe rozpoczynały i kończyły wykonywanie ruchu w tym samym momencie. Zrealizowano to według poniższego algorytmu.

1. Wyznaczenie modułu błędu dla każdego złącza lepsil
2. Znalezienie maksymalnego modułu błędu maxlepsil
3. Wyznaczenie współczynnika prędkości dla każdej osi $wsp = lepsil / maxlepsil$
4. Ograniczenie wyliczonego współczynnika wsp do przedziału $(0;1]$
5. Pomnożenie wartości wzmocnienia przez wsp .

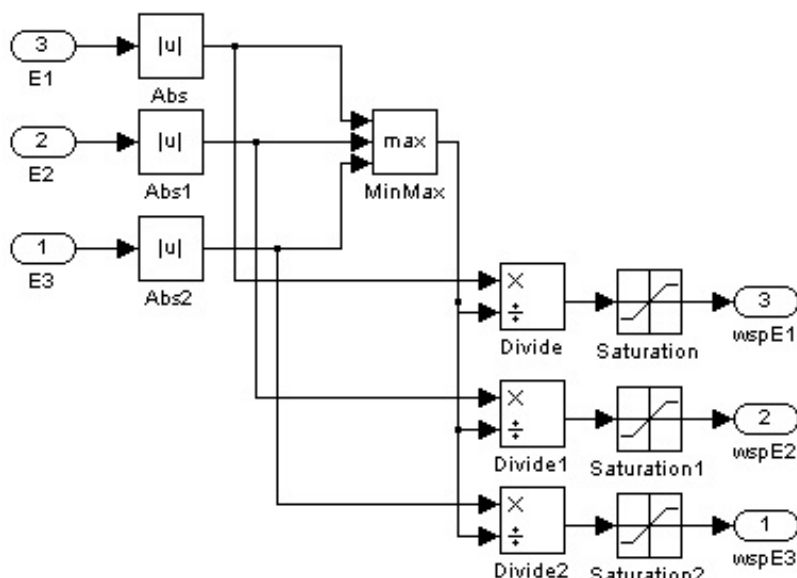
Poniżej przedstawiono model działania tej operacji.

W wejścia E1, E2, E3 zawierały różnicę między wartością zadaną, a wartością aktualną. Natomiast wyjścia $wspE1$, $wspE2$, $wspE3$ są podawane do serwomechanizmu jako mnożnik prędkości.

1.2.5. Bazowanie

Czujniki położenia działały na zasadzie pomiaru różnicy od wartości początkowej. Do ich prawidłowego działania należało zaraz po włączeniu robota ustawić go w określonej pozycji. W tym celu w każdym złączu znajdowały się cyfrowe czujniki położenia bazowego. Do bazowania wykorzystano następujący algorytm:

1. Ręczne wystereowanie robota, tak aby się znalazł w pozycji przedbazowej tzn. każde złącze znajdowało się w obszarze między czujnikiem bazowania, a położeniem skrajnym.



Rysunek 1.4: Schmet quasiliniowości

2. Zadanie małej prędkości na każdy zespół napędowy w kierunku czujnika bazowania.
3. W chwili dotarcia do pozycji bazowej zatrzymanie złącza.
4. Po ustawieniu każdego ramienia w pozycji bazowej, zresetowanie czujników położenia.

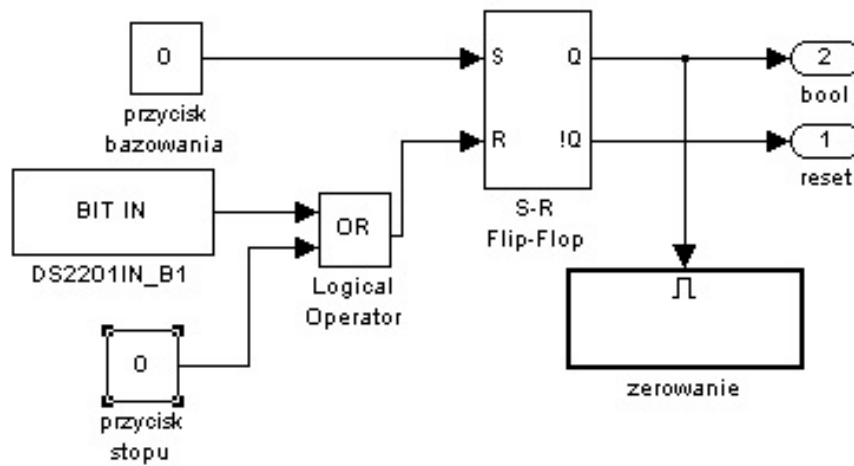
Powyższy algorytm został zrealizowany w Simulinku w następujący sposób.

W bloku zerowanie, wartość 0 została przypisana do odpowiednich zmiennych.

Do wykonania tej operacji wykorzystano przerzutnik SR. Sygnał z przycisku rozpoczynającego bazowanie został podłączony do wejścia SET. Natomiast alternatywa odczytu z cyfrowego czujnika położenia i przycisku Stop do wejścia RESET. Zastosowanie tej operacji dawało kontrolę nad robotem w przypadku awarii. Zgodnie z zasadą działania przerzutnika po chwilowym pojawieniu się sygnału z przycisku rozpoczęcia bazowania wartość Q została ustawiona na 1. Ten sygnał został doprowadzany do przełącznika, który ustawił odpowiednie sterowanie. Kiedy złącze doszło do pozycji bazowej, odczyt z czujnika bazowego zmienił się na 1 i wtedy sygnałem z wyjścia !Q czujnik położenia został zresetowany, a silnik zatrzymany.

1.3. Proste zadanie kinematyki

Niezbędne do sterowania robotem była zamiana współrzędnych złączowych na współrzędne kartezjańskie. W tym celu rozwiązano proste zadanie kinematyki. Poniżej przedstawiono schemat łańcucha kinematycznego, razem z wyznaczonymi układami współrzędnych złączowych według notacji Denavita-Hartenberga. Jako początek głównego układu współrzędnych przyjęto miejsce przecięcia się osi środkowej z podłogą, w pozycji bazowej robota.



Rysunek 1.5: Schemat bazowania

	α	a	d	Θ
1	$\pi/2$	0	d_1	0
2	$\pi/2$	0	p_1	$\Theta_2 + \pi/2$
3	0	p_2	0	$\Theta_3 + \pi/2$
4	0	p_3	0	$\Theta_4 - \pi/2$
5	$-\pi/2$	0	0	Θ_5
6	0	0	p_4	$\Theta_6 - \pi/2$

Tabela 1.1: Parametry złączy

Macierze przekształcenia dla poszczególnych złączy:

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_2 &= \begin{bmatrix} -\sin \Theta_2 & 0 & -\sin \Theta_2 & 0 \\ \cos \Theta_2 & 0 & \cos \Theta_2 & 0 \\ 0 & 1 & 0 & p_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_3 &= \\
 & \begin{bmatrix} -\sin \Theta_3 & -\cos \Theta_3 & 0 & -p_2 \sin \Theta_3 \\ \cos \Theta_3 & -\sin \Theta_3 & 0 & p_2 \cos \Theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
A_4 &= \begin{bmatrix} \sin \Theta_4 & \cos \Theta_4 & 0 & p_3 \sin \Theta_4 \\ -\cos \Theta_4 & \sin \Theta_4 & 0 & -p_3 \cos \Theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_5 &= \begin{bmatrix} \cos \Theta_5 & 0 & -\sin \Theta_5 & 0 \\ \sin \Theta_5 & 0 & \cos \Theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_6 &= \\
&\begin{bmatrix} \sin \Theta_6 & \cos \Theta_6 & 0 & 0 \\ -\cos \Theta_6 & \sin \Theta_6 & 0 & 0 \\ 0 & 0 & 1 & p_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Macierz przekształcenia:

$$T_0^6 = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned}
t_{11} &= \cos \Theta_2 \cos \Theta_6 - \sin \Theta_6 (\cos \Theta_5 (\cos \Theta_3 \cos \Theta_4 \sin \Theta_2 - \sin \Theta_2 \sin \Theta_3 \sin \Theta_4) - \\
&\sin \Theta_5 (\cos \Theta_3 \sin \Theta_2 \sin \Theta_4 + \cos \Theta_4 \sin \Theta_2 \sin \Theta_3))
\end{aligned}$$

Na podstawie wyliczonej macierzy przekształceń można było zapisać funkcje wyznaczające współrzędne końcówki roboczej w układzie kartezjańskim:

$$x = dd$$

$$y = gg$$

$$z = hh$$

1.4. Odwrotne zadanie kinematyki

Bibliografia

- [Dil00] A. Diller. *LaTeX wiersz po wierszu*. Wydawnictwo Helion, Gliwice, 2000.
- [Lam92] L. Lamport. *LaTeX system przygotowywania dokumentów*. Wydawnictwo Ariel, Krakow, 1992.
- [Szp11] M. Szpyrka. *On Line Alvis Manual*. AGH University of Science and Technology, 2011.
<http://fm.ia.agh.edu.pl/alvis:manual>.