

Prototypowanie sterownika dla robota 2DOF

Opis techniczny robota.

Robot 2DOF jest zespołem dwóch ramion o następujących danych:

- Liczba osi – dwie.
- Rodzaj napędu – silniki elektryczne prądu stałego typu PZTK 80-35.
- Zakres ruchów dla obu osi: ± 180 stopni.
- Wysokość – 1220 mm.
- Długość ramienia osi „1” – 630 mm.
- Długość ramienia osi „2” – 510 mm.
- Prędkości maksymalne dla obu osi 31.25 obr/min.

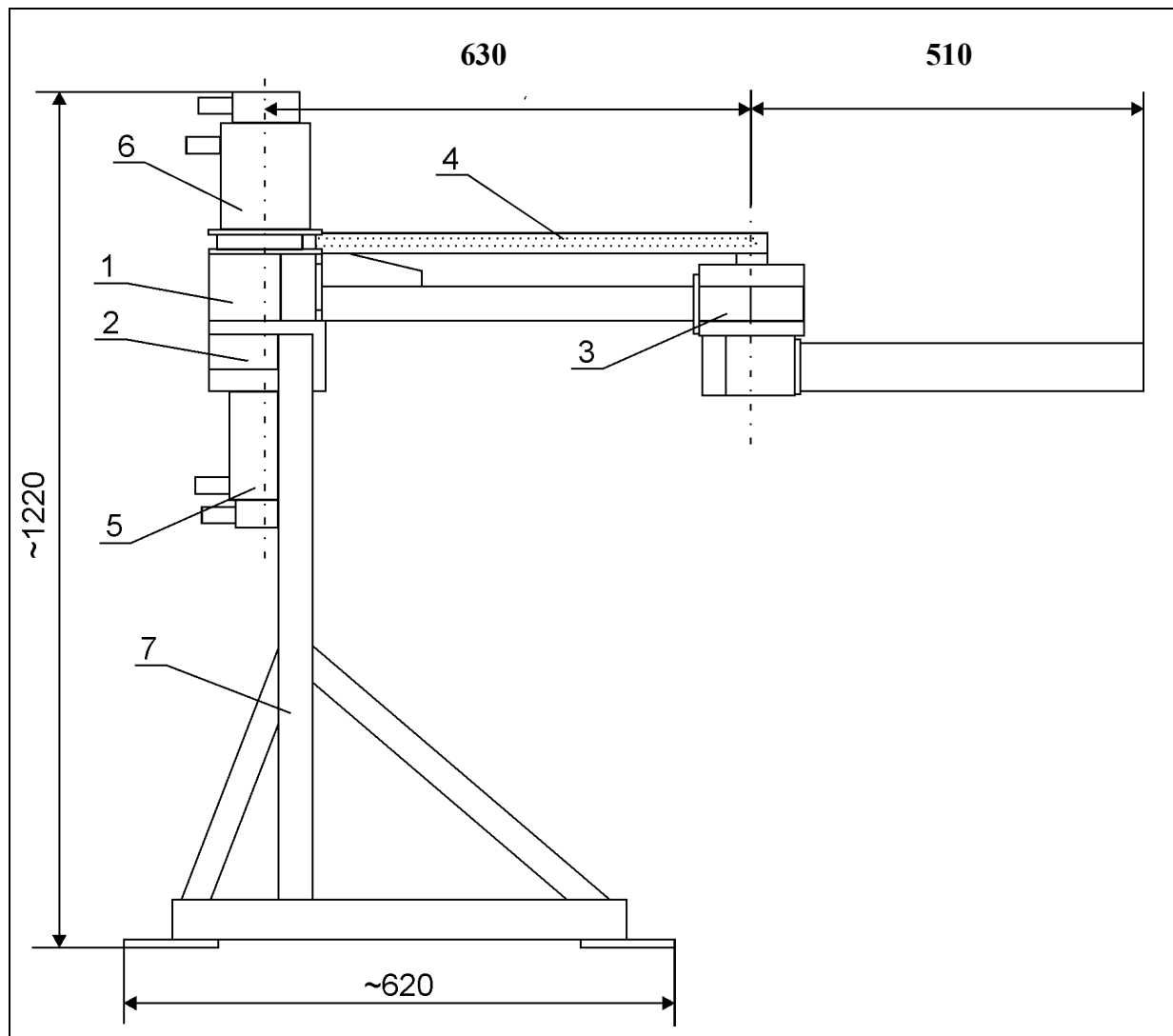
Robot zbudowany jest w oparciu o układ mechaniczny i elektryczny.

Układ mechaniczny składa się z następujących części:

- Stojak manipulatora
- Ramię „1” z przekładnią harmoniczną HDUC-20 o przełożeniu 128:1.
- Ramię „2” z przekładnią harmoniczną HDUC-20 o przełożeniu 128:1.
- Przekładnia pasowa zębata o przełożeniu 1:1 do napędu przekładni ramienia „2”.

Na układ elektryczny składają się:

- Napęd ramienia „1” i „2” – silniki prądu stałego PZTK 80-35.
- Enkodery typu MBL - 500 do pomiaru położenia ramienia „1” i „2” o następujących danych technicznych:
 - Konstrukcja - miniaturowe optoelektroniczne przetworniki obrotowo – impulsowe o średnicy 28 mm, oś ϕ 5 mm .
 - Napięcie zasilania - ± 5 VDC z tolerancją 0.25V.
 - Liczba działek na obrót – 500.
 - Typowy pobór prądu < 100 mA.
 - Sygnały wyjściowe A, B, C – nadajnik linii, 20 mA.
 - Przesunięcia fazowe A i B – 90 stopni
 - Zbocza sygnałów wyjść < 1 μ s.
 - Maksymalna częstotliwość sygnałów wyjść 150 kHz.
 - Typowa żywotność LED – 50000 h.



Rys.1 Układ ramion robota.

Podstawowe zespoły robota z Rys.1:

1. Napęd manipulatora.
2. Przekładnia harmoniczna ramienia 1.
3. Przekładnia harmoniczna ramienia 2.
4. Przekładnia pasowa zębata.
5. Silnik napędu ramienia 1.
6. Silnik napędu ramienia 2.
7. Stojak stanowiska

Zespoły napędowe poszczególnych osi obejmujące silnik prądu stałego, przekładnię oraz enkoder zliczający do pomiaru położenia kąтового (umieszczony przed przekładnią) połączone są z komputerem poprzez kartę DS1104 systemu dSPACE (wejścia enkoderowe karty wykorzystane do pomiaru położenia, wyjścia przetwornika DA do wysyłania sygnału sterującego do zespołów napędowych, wejścia cyfrowe do pomiaru stanu czujników bazowych).

1. W środowisku Simulink zrealizować w postaci schematu blokowego sterownik dla robota (serwomechanizmy z regulatorem P). Należy uruchomić **Matlab R2015a**. Następnie uruchomić środowisko **Simulink**. Przy wykorzystaniu bloków z biblioteki Simulink-a oraz bloków związanych z kartą DS1104 (biblioteka **dSPACE RTI1104/DS1104 MASTER PPC**) zrealizować schemat blokowy sterownika.

Bloki realizujące funkcje pomiarowe i sterujące karty **DS1104** potrzebne do realizacji schematu to:

DS1104ENC_POS_C1	- pomiar sygnału z enkodera
DS1104ENC_SETUP	- blok ustawiający typ enkoderów
DS1104ENC_SET_POS_C1	- reset enkoderów
DS1104DAC_C1	- wysyłanie sygnału sterującego
DS1104BIT_IN_C0	- pomiar stanu czujników bazowych.

Dla bloku **DS1104ENC_POS_C1** z dwóch wyjść które w nim występują należy wykorzystać wyjście *Enc position* , (na drugim umieścić *Terminator*).

Blok **DS1104ENC_SETUP**, umożliwiający ustawienie typu wykorzystywanych enkoderów, musi być umieszczony w schemacie blokowym dla zapewnienia poprawnej pracy układu. Blok ten nie ma połączeń z innymi blokami.

Dla uzyskania pomiaru położenia osi robota należy na wyjściu bloku **DS1104ENC_POS_C1** umieścić blok realizujący następującą operację

$$\frac{-2\pi}{500} \cdot \frac{180}{\pi} \cdot \frac{1}{128} = \frac{-2 \cdot 180}{500 \cdot 128},$$

przeliczając zliczone impulsy z enkodera na stopnie, uwzględniając rozdzielczość enkodera (500 impulsów/obrót) oraz przekładnię osi.

W tworzonym schemacie blokowym, regulator w serwomechanizmach dla poszczególnych osi robota, ustawić na działanie proporcjonalne o wzmacnieniu 0.2.

Sygnał sterujący wyliczony przez regulator powinien być, przed podaniem na wyjście, ograniczony do ± 3 (wykorzystać blok *Saturation*) oraz wprowadzony na blok *Rate Limiter* (parametry: 50, -50).

Przed wejściem na blok *DS1104DAC* należy umieścić blok *Gain* o wzmacnieniu 0.1 (wymaganie karty DS1104 związane z normowaniem sygnału).

2. Wykorzystując narzędzie **Real-Time Interface** wygenerować i załadować do karty DS1104 aplikację czasu rzeczywistego odpowiadającą zrealizowanemu schematowi blokowemu:

- w oknie roboczym wybrać *Code/C/C++ Code/Code Generation Options*
- w zakładce *Solver* sprawdzić ustawienie parametrów
Stop time : inf
Solver options: Type: Fixed-step oraz metodę całkowania (Euler)
- w podzakładce *Additional Options*
Fixed step size: 0.001 (jest to krok próbkowania)
- powrócić do zakładki *Code/C/C++ Code* po czym wybrać *Build Model (Ctrl+B)* ; wybór tej opcji powoduje automatyczną generację kodu C ze schematu

blokowego, kompilację, linkowanie i załadowanie aplikacji do karty DS1104, która przejmuje wykonywanie programu.

3. Przy pomocy pakietu **ControlDesk** zbudować pulpit operatorski umożliwiający komunikację z procesem realizowanym na karcie DS1104, tzn. zmiany i monitorowanie parametrów procesu oraz obserwacje i rejestrowanie przebiegów zmiennych:
 - po otwarciu **ControlDesk** wybrać *File / New* a następnie *Create New Projekt and Experiment*
 - nadać nazwę dla tworzonego projektu oraz ustawić właściwą kartotekę, po czym wybrać *Next>*
 - nadać nazwę dla experimentu, *wybrać Next>*
 - w kolejnym oknie zaakceptować platformę *DS1104 R&D Controller Board*, wcisnąć *Next>*
 - korzystając z opcji *Import from file* wczytać z odpowiedniej kartoteki zbiór *nazwa_aplikacji.sdf* zawierający informacje o zmiennych realizowanej aplikacji, wcisnąć *Finish*
 - w polu roboczym, przy pomocy dostępnych wirtualnych narzędzi (zakładka *Instrument Selector*), utworzyć panel operatorski
 - przypisać zmienne do wirtualnych narzędzi na pulpicie
 - uruchomienie utworzonego panelu (uruchomienie komunikacji z aplikacją czasu rzeczywistego) poprzez wybranie przycisku *Go Online*.

Funkcje jakie ma realizować wykonany prototyp sterownika:

1. Sterowanie napędów: serwomechanizmy dla poszczególnych osi robota, z ograniczeniami dla zakresu wartości zadanej kątów
2. Bazowanie robota
3. Pozycjonowanie w układzie współrzędnych konfiguracyjnych - tryb pracy ręcznej (niezależne ruszanie osiami) i tryb „wyzwalany” (ustawianie wartości zadanych dla osi i uruchomienie ruchu)
4. Zadawanie prędkości roboczej ruchu
5. Koordynacja prędkości ruchu
6. Pozycjonowanie w układzie kartezjańskim:
 - implementacja odwrotnego zadania kinematyki z uwzględnieniem ograniczeń przestrzeni roboczej