

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

ośrodek Automatykacji Procesów Produkcji

OZH

A

Główny wykonawca mgr inż. Zbigniew Stańczak

Wykonawcy : mgr inż. Wanda Ulatowska
mgr inż. Jan Baranowski
mgr inż. Rafał Zbiegieni

Konsultant

Nr zlecenia RP-153.2 - Cel 153 "Opracowanie robota z napędami elektrycznymi i sterowaniem MP i CP o udźwigu 2,5 kg".
Temat 153.2 pt.: "Wykonanie programu sterującego dla robota 2,5 kg".
Zadanie 4.1 pt.: "Wykonanie oprogramowania podstawowego dla celów przeprowadzenia badań modelu robota".
Zleceniodawca CPBR 7.1

Pracę rozpoczęto dnia 01.04.1990r. zakończono dnia 30.09.90r.
Główny Wykonawca Z-ca Dyrektora d/s Kierownik Ośrodka
Automatyki i Pomiarów
Stańczak wfx
mgr inż. Z. Stańczak dr inż. M. Wrzesień
doc dr inż. T. Gałazka

Praca zawiera:

Rozdzielnik - ilość egz:

stron 16

Egz. 1 BOINTE

rysunków

Egz. 2 OAR

fotografii

Egz. 3 OAE

tabel

Egz. 4 OAP

tablic

Egz. 5 OAP-8.

załączników

Egz. 6

Nr rejestr. 6509

Analiza deskrytorowa
ROBOTY PRZEMYSŁOWE; OPROGRAMOWANIE

Analiza dokumentacyjna

Praca zawiera opis programu symulacyjnego, zasady obsługi programów użytkowych oraz omówienie języka programowania. Przedstawiono także działanie programu symulującego panel programowania.

Tytuły poprzednich sprawozdań

1. "Wykonanie programu sterującego dla robota IRp-2,5 kg".
"Opracowanie założeń na oprogramowanie", nr rej. 5957
2. "Wykonanie programu sterującego dla robota IRp-2,5 kg"
"Wykonanie projektu systemu", nr rej. 6427.

338.45 [62/69].002.1/2 Roboty przemysłowe
681.3.06 Oprogramowanie

UKD

MERA-PIAP/TW 331/78 5000

S P I S T R E Ś C I

1. WPROWADZENIE.
2. BUDOWA PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.
3. OPIS PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.
 - 3.1. Tworzenie programu użytkowego.
 - 3.2. Sterowanie wykonaniem programu użytkowego.
4. OPIS DZIAŁANIA PROGRAMU.
 - 4.1. Synchronizacja.
 - 4.2. Praca automatyczna.
 - 4.3. Poruszanie manipulatorem.
 - 4.4. Programowanie instrukcji.
 - 4.5. Edycja programu użytkowego.
 - 4.6. Ręczne operowanie systemem.
5. OGÓLNA CHARAKTERYSTYKA PROGRAMU STERUJĄCEGO.
 - 5.1. Funkcja `init_master()`.
 - 5.2. Funkcja `system()`.
 - 5.2.1. Procedura `progpanl`.
 - 5.2.2. Procedura `oppanel`.
6. PODSTAWOWE CZYNNOSCI PROGRAMU STERUJĄCEGO.
 - 6.1. Wybór narzędzia.
 - 6.2. Ruch quasiliniowy.
 - 6.3. Ręczne operowanie manipulatorem.
 - 6.4. Obsługa panelu operacyjnego.
 - 6.5. Współpraca z panelem programowania.
 - 6.6. Wykonywanie programu użytkowego.
 - 6.7. Synchronizacja robota.
 - 6.8. Współpraca z pamięcią kasetową.
 - 6.9. Operacje na chwytakach.
7. WNIOSKI KOŃCOWE.
8. LITERATURA.

1. WPROWADZENIE.

=====

Niniejsze opracowanie zawiera opis oprogramowania podstawowego dla celów przeprowadzenia badań modelu robota przemysłowego o udźwigu 2,5 kg.

Prace nad opracowaniem programu sterującego robota IRp-2,5 podzielono na dwa etapy [1], z których pierwszy dotyczył tworzenia oprogramowania dla badań modelu robota, w oparciu o zmodernizowany panel programowania.

W trakcie prac programowych, ze względu na bardzo dużą awaryjność panelu programowania, powstała konieczność stworzenia programu na komputer IBM, symulującego pracę panelu programowania. Program ten został opracowany w dwóch wersjach, z których jedna jest przeznaczona do współpracy z programem sterującym robota IRp-2,5 kg. Realizuje on wszystkie funkcje zmodernizowanego panelu programowania, a jego zastosowanie ułatwi w przyszłości zaadaptowanie do układu sterowania robota IRp-2,5 kg, perspektywnego panelu programowania.

W niniejszym sprawozdaniu zamieszczono opis programu symulacyjnego oraz sposób jego wykorzystania podczas badań manipulatora robota IRp-2,5.

Zasady tworzenia, poprawiania i uruchamiania programów użytkowych, ogólne zasady dotyczące działania programu użytkowego, język programowania robota z punktu widzenia użytkownika oraz sposoby zgłaszania przez program sterownika sytuacji błędnych są zawarte w pracy [1].

W niniejszej pracy opisano także zasady ręcznego sterowania manipulatorem robota, w przypadku zastosowania programującego panel programowania robota na komputerze IBM. Program sterujący robota IRp-2,5 dla przeprowadzenia badań manipulatora jest oparty na standardowym programie sterującym robotów IRp, stosowanym dotychczas. Zrealizowano większość jego funkcji, wykorzystując część istniejących procedur programowych. Uwzględniono inną niż w przypadku robotów IRp-6/60 mechanikę manipulatora, dostosowując do niej oprogramowanie układu sterowania poprzez wprowadzenie nowych parametrów matematycznych. Ponadto przystosowano program sterujący do współpracy z manipulatorem o sześciu stopniach swobody, jakim jest manipulator robota o udźwigu 2,5 kg. Struktura programu pozostała bez zmian. Obieg głównej pętli programu jest analogiczny do programu standardowego, również podobne są zasady komunikacji pomiędzy układem sterowania a panelem programowania (bez względu na to, czy stosowany jest panel programowania, czy program symulujący go na komputerze IBM) oraz sygnalizacja błędów. Konfiguracja sprzętowa układu sterowania robota IRp-2,5 kg różni się od standardowej zwiększoną o jeden liczbą pakietów MA-70 (sterowniki osi) oraz mniejszą liczbą pakietów MC-42.

2. BUDOWA PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.

=====

Program źródłowy symulatora składa się z następujących zbiorów:

- ppsym.c - moduł główny, zajmujący się ponadto obsługą błędów,
- paut.c - moduł zajmujący się obsługą wykonywania programu użytkowego,
- ins.c - moduł zajmujący się obsługą tworzenia programu użytkowego,
- edy.c - moduł zajmujący się obsługą edycji programu użytkowego,
- syn.c - moduł zajmujący się synchronizacją robota,
- ope.c - moduł zajmujący się przepisywaniem programu użytkowego do PK oraz z PK,
- przes.c - moduł obsługujący odbiór przesyłek z panelu operacyjnego i wysyłanie przesyłek do panelu operacyjnego,
- prz.c - moduł umożliwiający ręczne sterowanie manipulatorem robota,
- pc.c - moduł współpracujący z modułem 'przes.c'.

Powyższe moduły programowe zostały skompilowane i zlinkowane pod systemem TURBOC, program wykonawczy umieszczony został w zbiorze 'panel.exe'.

3. OPIS PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.

=====

Symulator panelu programowania umożliwia:

- 1). przeprowadzenie synchronizacji manipulatora robota -
- stan pracy 'synchronizacja',
- 2). tworzenie i edycję programu użytkowego - stany pracy
'edycja' oraz 'programowanie instrukcji',
- 3). sterowanie wykonaniem programu użytkowego - stan
pracy 'praca automatyczna',
- 4). współpracę z panelem operacyjnym (a więc programem
sterującym) w zakresie:
- ręcznego sterowania manipulatorem robota we współ-
rzędnych wewnętrznych lub w ruchu pojedynczymi in-
krementami - stan pracy 'poruszanie manipulatorem'
- przepisywania programu użytkowego z PK do RAM oraz
z RAM do PK - stan pracy 'ręczne operowanie syste-
mem'.

3.1. Tworzenie programu użytkowego.

Tworzenie programu użytkowego robota obejmuje programowa-
nie następujących instrukcji:

- 1). instrukcja wyboru narzędzia,
- 2). instrukcja pozycjonowania typu PTP, z programowalną
prędkością ruchu lub czasu jego trwania i sposobem
osiągnięcia położenia zadanego (zgrubne/dokładne),
- 3). instrukcja skoku bezwarunkowego,
- 4). instrukcja czekania bezwarunkowego,
- 5). instrukcja początku pętli programowej,
- 6). instrukcja końca pętli programowej,
- 7). instrukcja ustawienia flagi lub wyjścia,
- 8). instrukcje operacji na chwytakach.

3.2. Sterowanie wykonaniem programu użytkowego.

Sterowanie wykonaniem programu użytkowego obejmuje:

- 1). start od pierwszej instrukcji programu,
- 2). start od bieżącej (aktualnie wyświetlonej) instru-
kcji programu,
- 3). pracę krokową (wykonanie wyświetlonej instrukcji),
- 4). zatrzymanie wykonywania programu,
- 5). wyjście do menu głównego.

4. OPIS DZIAŁANIA PROGRAMU.

=====

Program symulacyjny panelu programowania należy uruchamiać przy włączonym układzie sterowania robota oraz dokonanym połączeniu interfejsu szeregowego pakietu MM-16 w szafie z komputerem. Jest on wywoływany dyrektywą 'panel'.

Wywołanie programu, przy prawidłowej pracy sprzętu robota i prawidłowym połączeniu go z komputerem, powoduje ukazanie się napisu, zawierającego numer wersji programu i typ robota. Po krótkiej chwili ukazuje się główne menu programu. Jeśli wystąpi błąd w pracy sprzętu lub połączenie szafy sterowniczej z komputerem jest nieprawidłowe, program informuje o tym, poprzez wyświetlenie komunikatu "brak komunikacji z układem sterowania". Po zgłoszeniu się programu można załączyć zasilanie napędów robota.

Menu główne programu wygląda następująco :

- 1 Synchronizacja
 - 2 Praca automatyczna
 - 3 Poruszanie manipulatorem
 - 4 Programowanie instrukcji
 - 5 Edycja
 - 6 Ręczne operowanie systemem
- Esc Koniec pracy panelu

Wyboru stanu pracy użytkownik dokonuje poprzez naciśnięcie odpowiedniego klawisza - cyfry bądź klawisza 'Esc'.

Wybranie pozycji '1' powoduje rozpoczęcie procesu synchronizacji manipulatora robota, pozycji od '2' do '6' spowoduje przejście do wybranego stanu pracy, natomiast naciśnięcie klawisza 'Esc' powoduje zakończenie działania programu oraz przejście do systemu operacyjnego.

4.1. Synchronizacja.

Rozpoczęcie procesu synchronizowania robota jest możliwe tylko przy załączonych napędach. Synchronizacja robota w przypadku, gdy robot nie był wcześniej zsynchronizowany, rozpoczyna się po naciśnięciu klawisza z cyfrą '1' w menu głównym. Manipulator robota należy wcześniej doprowadzić do pozycji "przed synchronizacją", tzn. takiej, aby możliwy był ruch w każdej z osi w kierunku zgodnym z kierunkiem przebiegu synchronizacji robota. Proces synchronizacji można zatrzymać poprzez naciśnięcie klawisza z literą 's' lub klawisza 'spacja' (długi klawisz w dolnej części klawiatury). Ponowne naciśnięcie klawisza z cyfrą '1' powoduje kontynuację przerwanej procesu synchronizacji. Naciśnięcie klawisza 'Esc' powoduje powrót do menu głównego. W przypadku wystąpienia błędów w trakcie trwania synchronizacji podany zostanie odpowiedni komunikat.

Jesli natomiast synchronizacja manipulatora zakonczy sie prawidłowo pojawi się napis "Robot zsynchronizowany". Nacisnięcie klawisza 'Esc' spowoduje przejście do menu głównego programu symulacyjnego.

4.2. Praca automatyczna.

Pracę automatyczną można rozpocząć w przypadku, gdy:

- robot jest zsynchronizowany,
- w pamięci RAM systemu znajduje się program użytkowy.

Program użytkowy może zostać wprowadzony do pamięci za pomocą klawiatury lub przy wykorzystaniu pamięci kasetowej PK (w 8 stanie pracy).

Podczas pracy robota, na monitorze w okienku edycyjnym, wyświetlana jest aktualnie wykonywana instrukcja.

Wykonywanie programu użytkowego można w dowolnej chwili przerwać, poprzez nacisnięcie klawisza 'Space'.

Menu stanu 'praca automatyczna' wygląda następująco:

- 1 start od pierwszej instrukcji programu
- 2 start od aktualnie pokazanej instrukcji
- 3 wykonanie aktualnie pokazanej instrukcji
- + zwiększanie prędkości ruchu o 10% aktualnie wykonywanej prędkości
- zmniejszenie prędkości ruchu o 10% aktualnie wykonywanej prędkości
- UP zwiększenie bieżącej prędkości ruchu o 1 %
- DOWN zmniejszenie bieżącej prędkości ruchu o 1 %
- Space zatrzymanie wykonywania programu
- Esc powrót do menu głównego

Wcisnięcie odpowiednio :

- cyfry '1' - powoduje wystartowanie programu użytkowego (jeżeli taki istnieje) od jego pierwszej instrukcji,
- cyfry '2' - powoduje wystartowanie programu od instrukcji, która jest aktualnie wyświetlona w okienku edycyjnym,
- cyfry '3' - wykonanie instrukcji wyświetlonej w okienku edycyjnym (praca krokowa),
- klawisza '+' - powoduje zwiększenie prędkości wykonania programu użytkowego o kolejne 10% w stosunku do wszystkich instrukcji ruchu, proporcjonalnie do prędkości zapisanych w instrukcjach pozycjonowania,
- klawisza '-' - powoduje zmniejszenie prędkości wykonania programu użytkowego jak poprzednio,
- klawisza 'UP' - powoduje zwiększenie bieżącej prędkości ruchu o 1 %,
- klawisza 'DOWN' - powoduje zmniejszenie bieżącej prędkości rucho o 1%,
- klawisza 'Space' - powoduje zatrzymanie wykonywania programu,
- klawisza 'Esc' - powoduje powrót do menu głównego (klawisz jest nieaktywny podczas pracy manipulatora robota)

4.3. Poruszanie manipulatorem.

Menu stanu "poruszanie manipulatorem" wygląda następująco:

- 1 Ruch w pojedynczych inkrementach
 - 2 Układ współrzędnych wewnętrznych
- Esc powrót do menu głównego

Nacisnięcie klawisza '1' lub '2' powoduje wejście do wewnętrznego, identycznego w obu przypadkach menu, opisującego klawisze klawiatury, służące do poruszania manipulatorem. Pojawia się jeden z dwóch możliwych napisów :

RUCH W POJEDYNCZYCH INKREMENTACH

lub

UKŁAD WSPÓLRZĘDNYCH WEWNĘTRZNYCH

a następnie :

Poruszanie manipulatorem - klawiatura funkcyjna.

F1	OŚ	1 -
F2	OŚ	1 +
F3	OŚ	2 -
F4	OŚ	2 +
F5	OŚ	3 -
F6	OŚ	3 +
F7	OŚ	4 -
F8	OŚ	4 +
F9	OŚ	5 -
F10	OŚ	5 +
Ins	OŚ	6 -
Del	OŚ	6 +
Tab	Wł./wył.	ruchu ciągłego
Esc		Powrót do głównego menu

Znak '+' oznacza ruch manipulatora w kierunku zgodnym z kierunkiem ruchu w trakcie trwania procesu synchronizacji, znak '-' natomiast - oznacza ruch w kierunku przeciwnym do kierunku przebiegu synchronizacji.

Nacisnięcie klawisza z cyfrą '1' umożliwia poruszanie manipulatorem robota pojedynczymi inkrementami. W praktyce oznacza to możliwość wolnego, dokładnego dojścia do zadanego punktu przestrzeni roboczej. Wielkość inkrementów ruchu została zadana programowo. Nacisnięcie klawisza z cyfrą '2' umożliwia poruszanie manipulatorem robota w układzie współrzędnych wewnętrznych, oznacza to ruch, w którym jednostka ruchu jest wielokrotnością 1 inkrementu. Wielkość tę można dobrać w zależności od potrzeb.

4.4. Programowanie instrukcji.

Menu programowania instrukcji programu użytkowego jest pokazane poniżej:

- 0 wybór narzędzia
- 1 pozycjonowanie
- 2 skok bezwarunkowy
- 3 oczekiwanie bezwarunkowe
- 4 początek pętli programowej
- 5 koniec pętli programowej
- 6 ustawienie flagi lub wyjścia
- 7 operacje na chwytakach
- Esc powrót do menu głównego

Wcisnięcie odpowiednio :

- cyfry '0' - umożliwia dokonanie wyboru narzędzia pracy,
- cyfry '1' - umożliwia zaprogramowanie instrukcji pozycjonowania. Dokonuje się określenia wartości prędkości (% prędkości podstawowej) i sposobu dojścia do zadanego położenia,
- cyfry '2' - umożliwia określenie numeru instrukcji programu, do której ma nastąpić skok,
- cyfry '3' - umożliwia określenie czasu czekania (w sek.),
- cyfry '4' - umożliwia otwarcie w programie pętli,
- cyfry '5' - umożliwia zamknięcie pętli,
- cyfry '6' - umożliwia ustawienie flagi lub wyjścia na zadaną wartość, wejście lub flagę,
- cyfry '7' - umożliwia operowanie chwytakami,
- klawisza 'Esc' - umożliwia powrót do menu głównego.

4.5. Edycja programu użytkowego.

Wejście w stan edycji programu, powoduje wyprowadzenie na ekran monitora :

- napisu " EDYCJA PROGRAMU ",
- informacji o bieżącej instrukcji programu w postaci :
numer instrukcji, kod instrukcji, [parametry instrukcji],
- menu stanu edycji programu.

Menu stanu edycji programu zawiera:

- 1 instrukcja w przód
- 2 instrukcja w tył
- 3 usuń instrukcję
- 4 wstaw instrukcję
- 5 początek programu
- 6 koniec programu
- 7 zmiana instrukcji
- 8 ustawienie numeru instrukcji
- Esc powrót do menu głównego.

Wcisnięcie odpowiednio :

- cyfry '1' - powoduje pokazanie następnej, względem bieżącej, instrukcji. Jeżeli następna instrukcja nie istnieje pojawi się komunikat :
"nie ma następnej instrukcji - koniec programu".
- cyfry '2' - powoduje pokazanie poprzedniej, względem bieżącej, instrukcji programu. Jeżeli instrukcja ta nie istnieje, pojawi się komunikat :
"nie ma poprzedniej instrukcji - początek programu",
- cyfry '3' - powoduje usunięcie grupy instrukcji programu, w szczególnym przypadku może to być jedna instrukcja,
- cyfry '4' - powoduje wstawienie numeru instrukcji programu, pomiędzy numerem instrukcji bieżącej i następnej. Nowy numer instrukcji jest wyświetlony na ekranie.
- cyfry '5' - powoduje skok do początku programu,
- cyfry '6' - powoduje skok do końca programu,
- cyfry '7' - pozwala na zmianę parametrów numerycznych bieżącej instrukcji lub na zamianę aktualnej instrukcji na inną,
- cyfry '8' - powoduje zmianę aktualnej instrukcji na instrukcję o podanym numerze,
- klawisza 'Esc' - umożliwia powrót do menu głównego.

4.6. Ręczne operowanie systemem.

Menu ręcznego operowania systemem jest następujące:

- 1 przepisanie programu z PK do RAM
- 2 przepisanie programu z RAM do PK
- Esc powrót do menu głównego.

Wcisnięcie odpowiednio :

- cyfry '1' - powoduje rozpoczęcie przepisywania programu użytkowego z pamięci kasetowej do pamięci RAM systemu sterowania,
- cyfry '2' - powoduje rozpoczęcie przepisywania programu użytkowego z pamięci RAM do pamięci kasetowej,
- klawisza 'Esc' - umożliwia powrót do menu głównego.

5. OGÓLNA CHARAKTERYSTYKA PROGRAMU STERUJĄCEGO.

=====

Standardowy program sterujący, jego budowa i działanie zostały dokładnie opisane w literaturze [2]. W poniższym rozdziale zajmiemy się tylko wyjaśnieniem różnic, jakie w stosunku do tego programu, wykazuje program opracowany dla robota 2,5 kg. Schematy blokowe pętli głównej oraz podstawowych funkcji programu zostały przedstawione w opracowaniu [1].

Główną procedurą programu sterującego robota IRp-2,5 jest funkcja 'main'. Funkcja ta wywoływana jako main(), inicjalizuje system (program sterujący i sprzęt). Inicjalizacja wykonywana jest sekwencyjnie, w efekcie wywołania z funkcji 'main' następujących funkcji :

- init_master() - inicjalizacja programu głównego,
- system() - obsługa sterowników osi, panelu programowania i panelu operacyjnego,
- manipulacja() - obsługa ręcznego sterowania manipulatorem.

Po inicjalizacji działania systemu następuje testowanie, czy nie wystąpił błąd związany z pracą sprzętu:

- brak gotowości osi,
- stop awaryjny,
- zbyt duży błąd położenia,
- wyjście ze stanu 'praca',
- za wysoka temperatura w sterowniku,
- utrata synchronizacji,
- przeciążenie,
- inne błędy.

Wystąpienie któregośkolwiek z wymienionych błędów powoduje wywołanie funkcji error(n), (gdzie n - numer błędu), która ustawia system w stan ERROR_STATE, załącza miganie lampki BŁĄD oraz przekazuje kod błędu do panelu programowania.

5.1. Funkcja init_master().

Funkcja ta wywoływana jest jako init_master(). Jej rola polega na zainicjowaniu pracy całego systemu robota.

Wykonywane przez program kolejne kroki są następujące :

- 1). załączany jest sygnał control_ready(),
- 2). system ustawiany jest w stan pracy ręcznej,
- 3). inicjalizowany jest koprocesor arytmetyczny Intel 8087 oraz jego biblioteka,
- 4). czas makrointerpolacji definiowany jest jako 32 ms,
- 5). ograniczenie na maksymalny przyrost wartości zadanej dla sterownika osi A5, ustawiane jest jako połowa różnicy ograniczeń górnego i dolnego,
- 7). ustawiany jest stan początkowy wskaźników :
 - włączony jest stop awaryjny,
 - sygnalizowane jest przekroczenie limitu prądowego,

- brak utraty pozycji,
 - nie ma sygnalizacji błędów,
 - robot nie jest zsynchronizowany,
 - nie jest włączony przycisk praca,
 - zerowany jest wskaźnik stosu pozycji,
 - uwzględniany jest przewijak pamięci kasetowej nr 2.
- 8). wywoływane są funkcje inicjalizujące :
- `init_mmfilgs()` - inicjalizacja funkcji wypełniających obszar użytkowy,
 - `init_intmov()` - inicjalizacja funkcji obsługujących ruch we współrzędnych wewnętrznych,
 - `init_extmov()` - inicjalizacja funkcji obsługujących ruch we współrzędnych zewnętrznych,
 - `init_manipulacja()` - inicjalizacja funkcji obsługujących ręczne operowanie systemem,
 - `init_interpreter()` - inicjalizacja interpretera,
 - `init_hardware()` - inicjalizacja funkcji obsługi sprzętu,
- 9). zapalane są wszystkie lampki panelu operacyjnego na jedną sekundę,
- 10). gaszone są lampki panelu operacyjnego,
- 11). miga lampka synchronizacji.

5.2. Funkcja `system()`.

Funkcja ta wywoływana jest jako `system()`. Inicjuje ona obsługę istotnych elementów systemu sterowania. Są tu kolejno wywoływane następujące funkcje :

- 1). `axes_controller()` - kontrola stanu sterowników osi,
- 2). `progpanl()` - obsługa panelu programowania,
- 3). `oppanel()` - obsługa panelu operacyjnego.

5.2.1. Procedura `progpanl`.

Procedura `progpanl()` jest wywoływana w podprogramie 'system', wykonuje się cyklicznie, co okres obliczeniowy programu sterującego. Kolejno wykonywane są następujące czynności :

- 1). wywołanie podprogramu odczytu adresu przesyłki,
- 2). wywołanie podprogramu obsługi przesyłki,
- 3). powrót z 'progpanl' do procedury wywołującej.

Podprogramy bezpośrednio obsługujące przesyłki wykorzystują dwie procedury pomocnicze :

- `pkfault()` - sygnalizuje wystąpienie błędu podczas transmisji z/do PK,
- `startper()` - sprawdza, czy można rozpocząć wykonywanie programu użytkowego.

Funkcje wykorzystywane przez procedurę `progpanl()` :

- funkcja cont(),
- funkcja cont_step(),
- funkcja cvecend(),
- funkcja cvecstrt(),
- funkcja get_geom(),
- funkcja getinter(),
- funkcja gripoper(),
- funkcja hold_pk(),
- funkcja no_parcel(),
- funkcja pgmstart(),
- funkcja pgmstop(),
- funkcja restrpgm(),
- funkcja savepgm(),
- funkcja sendvelo(),
- funkcja seterror(),
- funkcja setgeom(),
- funkcja storeref(),
- funkcja switchfr(),
- funkcje interfejsowe.

5.2.2. Procedura oppanel.

Funkcja oppanel() obsługuje sygnały podawane na wejścia systemowego bloku wejsc/wyjsc dwustanowych pakietu MC-42 (w układzie sterowania jest również blok wejsc/wyjsc użytkownika). Do tego bloku są dołączone cztery przyciski panelu operacyjnego (START PROGRAMU, SYNCHRONIZACJA, ODCZYT KASETY, STOP PROGRAMU) oraz dwa styczniki (PRACA, STOP AWARYJNY). Procedura ta jest wykonywana cyklicznie, co okres obliczeniowy programu sterującego.

Wykonywane są kolejno następujące kroki :

- 1). odczytanie stanu wejsc bloku MC-42,
- 2). wykonanie procedur obsługujących sygnały :
 - procedura PROGRAM_START_BUTTON,
 - procedura SYNCHRONIZE_BUTTON,
 - procedura READ_CASSETTE_BUTTON,
 - procedura PROGRAM_STOP_BUTTON,
 - procedura SEARCH_STOP_BUTTON,
 - procedura EMERGENCY_STOP_BUTTON,
 - procedura WORK_BUTTON,
 - procedura PROG_TEST_BUTTON,
 - procedura OVERCURRENT_SIGNAL.

5.3. Funkcja manipulacja().

Funkcja ta wywoływana jest jako manipulacja(). Jej zadanie to realizacja sterowania ramieniem robota za pośrednictwem przycisków lub joystick'a.

Funkcja ta wykonuje kolejno następujące czynności :

- 1). sprawdzenie :
 - dla panelu z joystick'iem - stanu aktywności joystick'a,
 - dla panelu z przyciskami - czy jest wciśnięty któ-

rys z przycisków grupy 8 (przyciski do ręcznego sterowania manipulatorem).

Jeśli odpowiedź jest negatywna, to funkcja kończy swoje działanie,

- 2). sprawdzenie, czy jest wciśnięty przycisk stopu awaryjnego oraz czy są włączone napędy.

Jeśli stop awaryjny jest włączony lub napędy nie są włączone, to generowane są odpowiednie komunikaty o błędach i funkcja kończy działanie,

- 3). umożliwienie operatorowi poruszania manipulatorem robota (przy włączonych napędach i braku stopu awaryjnego). Sposób wykonania ruchu zależy od wybranych wcześniej: układu współrzędnych i grupy sterowanych osi,

- 4). zbadanie prawidłowości wykonania ruchu (po powrocie do funkcji wywołującej).

Jeśli wystąpił błąd, to do panelu programowania przekazywany jest odpowiedni komunikat i funkcja kończy działanie,

Jeśli ruch został wykonany prawidłowo, to funkcja umożliwia operatorowi wybór ilości inkrementów ruchu do realizacji lub rezygnację z ruchu inkrementami.

Funkcja kończy działanie.

6. PODSTAWOWE CZYNNOSCI PROGRAMU STERUJĄCEGO. =====

Podstawowe czynności programu sterującego dla badań manipulatora robota IRp-2,5 kg. to :

- wybór narzędzia pracy,
- spowodowanie quasiliniowego ruchu punktu roboczego narzędzia we współrzędnych wewnętrznych i w ruchu pojedynczymi inkrementami,
- operowanie manipulatorem robota za pomocą panelu programowania (programu symulacyjnego) oraz interpretacja wprowadzonego przez użytkownika programu użytkowego.

6.1. Wybór narzędzia. -----

Wybór narzędzia pracy polega na wyznaczeniu współrzędnych punktu TCP (punkt roboczy narzędzia) oraz orientacji wyróżnionej osi narzędzia względem układu kartezjańskiego związanego z podstawą robota.

6.2. Ruch quasiliniowy.

Ruch robota przy jednoczesnej pracy wszystkich sterowników został nazwany pozycjonowaniem quasiliniowym.

Wartość każdej współrzędnej wewnętrznej robota zmienia się liniowo w funkcji czasu. W efekcie uzyskuje się ruch liniowy w przestrzeni współrzędnych wewnętrznych. Nie jest to jednak ruch liniowy w przestrzeni kartezjańskiej, ze względu na nieliniowe przekształcenie współrzędnych wewnętrznych na współrzędne kartezjańskie.

6.3. Ręczne operowanie manipulatorem. -----

Poruszanie manipulatorem realizowane jest w sposób typowy, tj. deszyfrowane są standardowe przesyłki od panelu programowania (w tym przypadku od programu symulacyjnego w którym odpowiednie przyciski klawiatury symulują wychylenie joystick'a). Inicjalizacja ruchu manipulatorem robota jest wykonywana przez funkcję 'init_manipulacja()', która wywołuje funkcja 'init_master()'.

Opis funkcji 'init_manipulacja()' zamieszczono przy okazji opisu funkcji 'master()'. Następnie wywoływana jest funkcja 'joystick()'. Jeśli joystick lub też odpowiednie przyciski są w stanie aktywności, to funkcja 'joystick()' powoduje wykonanie ruchu ramieniem lub kciśnią robota.

6.4. Obsługa panelu operacyjnego.

Podprogram 'oppanel' obsługuje sygnały podawane na wejścia systemowych pakietów we/wy dwustanowych MC-42. Dołączone są tu cztery przyciski panelu operacyjnego (START PROGRAMU, SYNCHRONIZACJA, ODCZYT KASETY oraz STOP PROGRAMU), dwa styczniki (PRACA i STOP AWARYJNY), a także sygnał informujący o przekroczeniu ograniczenia prądowego. Procedura 'oppanel' jest wywoływana w podprogramie 'system', jest więc wykonywana cyklicznie, co okres obliczeniowy programu sterującego.

6.5. Współpraca z panelem programowania.

Komunikacja pomiędzy panelem programowania (czy też programem symulacyjnym) a sterownikiem odbywa się przez złącze szeregowo pakietu MM-16 i analogiczne złącze w panelu programowania. Informacje są przekazywane w postaci przesyłek. Program sterujący nadaje przesyłki procedura 'putparcl', a odbiera procedurą 'getparcl'.

Przed wysłaniem przesyłki program sterujący musi przygotować bufor zawierający przesyłkę. Następnie poprzez wywołanie procedury 'putparcl' inicjuje proces wysyłania przesyłki do panelu programowania. Wszystkie przesyłki są przesyłane w postaci zakodowanej, co ma na celu zwiększenie możliwości wykrywania błędów w odbieranych przesyłkach oraz ułatwia modyfikacje programu i diagnostykę sprzętu (możliwość bezpośredniego podłączenia złącz szeregowych do monitora z klawiaturą i obserwowania oraz symulowania przesyłek).

Procedura 'progpanl' jest wywoływana w podprogramie 'system', a więc wykonuje się cyklicznie, co okres obliczeniowy programu sterującego (podobnie jak procedura obsługi panelu operacyjnego - 'oppanel'). Po wejściu do podprogramu 'progpanl' jest wywoływany podprogram odczytu adresu przesyłki. Wartość przez niego zwracana jest podawiana pod zmienną, która wskazuje typ przesyłki oraz tablice jej parametrów. Jeśli zmienna ta zawiera wartość '0', oznacza to, że w buforze nie ma nieobsłużonej przesyłki. Po odczytaniu adresu przesyłki wywoływany jest podprogram jej obsługi. Zakończenie procesu obsługi aktualnej przesyłki powoduje powrót z 'progpanl' do procedury wywołującej.

Program sterujący zawiera ponadto dwie procedury pomocnicze dla procedur bezpośrednio obsługujących przesyłki. Są to :

- procedura 'pkfault' - sygnalizująca wystąpienie błędu podczas transmisji z/do pamięci kasetowej,
- procedura 'startper' - sprawdzająca, czy można rozpocząć wykonywanie programu użytkowego (sprawdzenie synchronizacji, istnienia programu użytkowego).

6.6. Wykonywanie programu użytkowego.

Program sterujący zawiera procedury, umożliwiające wykonywanie programu wprowadzonego przez użytkownika, bądź za pomocą pamięci kasetowej systemu, bądź też poprzez ręczne wpisanie go z klawiatury panelu programowania (komputera, w przypadku pracy z programem symulacyjnym). Należą tu :

- procedura 'pgmstart' - realizująca rozpoczęcie wykonywania programu użytkowego od pierwszej instrukcji,
- procedura 'cont' - realizująca rozpoczęcie wykonywania programu użytkowego od instrukcji bieżącej,
- procedura 'cont_step' - realizująca wykonanie pojedynczej instrukcji programu (pracę krokową),
- procedura 'pgmstop' - obsługująca przesyłkę przerywającą wykonywanie programu.

6.7. Synchronizacja robota.

Synchronizowanie robota, z punktu widzenia programu sterującego może być dokonywane na dwa sposoby (tylko w przypadku zastosowania programu symulacyjnego).

Pierwszy wykorzystuje przycisk na panelu operacyjnym, drugi zaś to rozpoczęcie synchronizacji odpowiednim przyciskiem z programu symulacyjnego. Zastosowanie do pracy panelu programowania ogranicza dokonanie procesu synchronizacji do przycisku na panelu operacyjnym (brak odpowiedniego przycisku w panelu).

Procedurą umożliwiającą dojście do punktu synchronizacji jest 'getgeom'.

6.8. Współpraca z pamięcią kasetową.

Program sterujący dla badań manipulatora robota IRp-2,5 kg umożliwia także korzystanie z pamięci kasetowej. Są tu wykorzystane następujące procedury :

- procedura 'restrpgm' - obsługująca przesyłkę z panelu programowania, nakazująca wczytanie programu z pamięci kasetowej,
- procedura 'savepgm' - przepisująca program użytkowy z pamięci RAM układu sterowania do pamięci kasetowej,
- procedura 'hold_pk3' - realizująca zatrzymanie pamięci kasetowej w trakcie trwania transmisji,.

6.9. Operacje na chwytakach.

Procedura 'gripoper' jest procedurą realizującą operacje na chwytakach. Na treść procedury składa się wywołanie podprogramu obsługi chwytaków.

6.10. Obsługa błędów.

Do obsługi błędów powstałych w trakcie pracy służy procedura 'seterror'. Obsługuje ona przesyłkę z panelu programowania, informującą o wystąpieniu błędu.

7. WNIOSKI KONCOWE.

=====

Dotychczasowe prace nad 6-cio osiowym robotem o udźwigu 2,5 kg pozwalają na podsumowanie i nakreślenie dalszego kierunku prac w tym temacie.

Kierunki rozwoju robotów przemysłowych w krajach wysoko uprzemysłowionych potwierdzają sensowność wyboru takiego typu robota do dalszych prac badawczych.

W chwili obecnej roboty o 6-ciu stopniach swobody wypierają manipulatory 5-cio osiowe głównie ze względu na większe możliwości aplikacyjne oraz uniwersalność.

Przestrzeń robocza może być wykorzystana w sposób dość swobodny. Wykonany na podstawie modelu użytkowego model kinematyczny manipulatora jest relatywnie prostszy w obliczeniach, dzięki zbieżności punktowej trzech ostatnich osi. Jest to sprawa niebagatelna, biorąc pod uwagę potrzebę prowadzenia obliczeń zmiennoprzecinkowych w czasie rzeczywistym oraz konieczność analizy rozwiązań odwrotnego zadania kinematyki.

Takie rozwiązanie techniczne pozwoliło na korzystanie z "naturalnej" metody rozwiązania zadania odwrotnego kinematyki; nie zaistniała potrzeba stosowania uniwersalnych, ale złożonych w obliczeniach metod macierzowych, jak n.p. metoda łańcucha kinematycznego Craig'a.

Rozwiązania niemożliwe do wykorzystania w praktyce są rugowane na bieżąco w trakcie obliczeń, co zmniejsza liczbę operacji matematycznych.

W trakcie przekształceń istnieje silny związek z fizyczną interpretacją wyników, w przeciwności do metod macierzowych, w których i tak należy wykonać w zasadzie wszystkie obliczenia. Olbrzymia liczba rozwiązań musi podlegać specjalnym analizom, rugującym rozwiązania spoza przestrzeni roboczej manipulatora robota, poprzez liczenie odpowiednich, pierwotnych zadań kinematycznych.

Jeśli, tak cenna cecha manipulatora o 6-ciu stopniach swobody, jak uniwersalność (giętkość), została by wzmocniona dużą precyzyjnością (dzięki zastosowaniu odpowiednich metod programowych oraz technologicznych) oraz ewentualnym wyposażeniem go w korekcję wizyjną, efekt końcowy może znaleźć szerokie zastosowanie w praktyce.

Niestety dotychczasowe elementy aplikacji, tj:

- zawodny i mało wydajny obliczeniowo układ sterowania,
- brak systemu czasu rzeczywistego, jako programowej podstawy programu sterującego,
- przestarzałe narzędzia programowe,
- brak pełnej symulacji komputerowej systemu robota, jako etapu projektowania,

mogą wydłużyć czas opracowania i pogorszyć końcową jakość wyrobu.

Dodatkowo, wydaje się być pożądane, znaczne wzmocnienie pierwszych trzech napędów manipulatora, w celu uzyskania większej dynamiki całego robota oraz powiększenie jego nominalnego udźwigu.

Wyeliminowanie wspomnianych niedogodności może w sposób zasadniczy zadecydować o jego rozpowszechnieniu. Robot, o umownej nazwie IRp-2,5, może być z powodzeniem stosowany w zautomatyzowanych liniach produkcyjnych, po uprzednim wyposażeniu go w odpowiednie oprogramowanie, standaryzujące wymianę informacji między innymi aplikacjami (proponujemy standard MMS), w takim zastosowaniu można właściwie wykorzystać wszystkie jego zalety. Uniwersalność tego typu robota wybitnie predysponuje go do zastosowań w komputerowo zintegrowanym wytwarzaniu (CIM), którego istotą jest akomodacyjność procesów wytwórczych.

8. LITERATURA.

=====

- [1] - Wykonanie programu sterującego dla robota IRp-2,5 kg.
Opracowanie założeń na oprogramowanie.
- Sprawozdanie PIAP - nr rej. 5957.
- [2] - Opis programu sterującego robotów IRp-6/60.
- Dokumentacja eksploatacyjna PIAP.