Міністерство освіти і науки України
Національний університет «Чернігівська політехніка»

# Промислові роботи

**Частина 2**

**Методичні вказівки**

до виконання лабораторних робіт
з дисципліни “Промислові роботи:
будова, програмування, експлуатація”
для здобувачів другого (магістерського) рівня вищої освіти
за спеціальністю 133 “Галузеве машинобудування”
освітньо-професійної програми “Галузеве машинобудування”

Затверджено
на засіданні кафедри
“Автомобільного транспорту та галузевого машинобудування”
Протокол № 2
від 31.08.2021 р.

Чернігів НУ «Чернігівська політехніка» 2021

Промислові роботи. Частина 2. Методичні вказівки до виконання лабораторних робіт з дисципліни “Промислові роботи: будова, програмування, експлуатація” для здобувачів другого (магістерського) рівня вищої освіти за спеціальністю 133 “Галузеве машинобудування” освітньо-професійної програми “Галузеве машинобудування”/ Укл.: Кальченко В.В., Пасов Г.В. – Чернігів: НУ “Чернігівська політехніка”, 2021. – 100 с.

Укладачі: Кальченко Володимир Віталійович
доктор технічних наук, професор
Пасов Геннадій Володимирович
кандидат технічних наук, доцент

Відповідальний за випуск: Кальченко В.І., завідувач кафедри,
доктор технічних наук, професор

Рецензент: Кологойда А.В., кандидат технічних наук,
доцент кафедри “Автомобільний транспорт
та галузеве машинобудування ”
Національного університету
“Чернігівська політехніка”

# ЛАБОРАТОРНА РОБОТА №4
# 4 ПРОМИСЛОВИЙ РОБОТ М20П.40.01

## 4.1 Мета роботи

Вивчити конструкцію, основні вузли і роботу промислового робота М20П.40.01. Придбати навички по складанню керуючої програми і настроюванню робота.

## 4.2 Призначення промислового робота М20П.40.01

Промисловий робот з ЧПК М20П.40.01 призначений для автоматизації встановлення-зняття заготовок і деталей, зміни інструментів та інших допоміжних операцій при обслуговуванні верстатів із ЧПК. Пристрій даного типу може обслуговувати один чи два верстати й утворювати разом з накопичувальними і транспортними пристроями гнучкий виробничий оброблюючий комплекс, призначений для тривалої роботи без участі оператора.

## 4.3 Технічна характеристика промислового робота М20П.40.01

Номінальна вантажопідйомність, кг ..... 20

Кількість ступенів рухливості ..... 5

Найбільші лінійні переміщення, мм:

по вертикальній осі ..... 500

по горизонтальній осі (L3) ..... 500; 800; 1000

Найбільше кутове переміщення, град:

руки щодо вертикальної осі ..... 300

кисті щодо поздовжньої осі ..... 90; +180

кисті щодо поперечної осі ..... ±3,5

Діапазон швидкостей лінійних переміщень, м/с:

по вертикальній осі ..... 0,005...0,5

по горизонтальній осі ..... 0,008...1,0

Діапазон швидкості кутових переміщень, град/с:

руки щодо вертикальної осі ..... 60

кисті щодо поздовжньої осі ..... 60

кисті щодо поперечної осі ..... 0

Найбільша абсолютна помилка позиціонування, мм ..... ±1

Зусилля затиску схвата, Н ..... 350; 500

Час затискання - розтискання, с ..... 2

Діапазон розмірів, захоплюваних деталей по зовнішньому

діаметру, мм ..... 50...268

Маса (без пристрою ЧПК), кг ..... 570

**4.4 Складові частини та основні вузли промислового робота М20П.40.01**

Загальний вид ПР М20П.40.01 і його технічна характеристика приведені на рисунку 22 [1].

Промисловий робот складається з маніпулятора 1, змінних схватів 2 (виконання С01, ..., С05 і С07) і пристрою ЧПК, виконаного у вигляді автономної стойки 3. Маніпулятор ПР містить у собі наступні складальні одиниці, деякі з яких можуть бути різноманітного виконання: механізм повороту 4; механізм підйому й опускання 5; механізм висування руки 6 (базове, 01 і 02 виконання); балансир 7; блок повороту (кисть руки) 8 (виконання 1 чи 2); блок підготовки повітря (на рисунку 22 не показаний).

Пристрій ЧПК позиційного типу забезпечує керування переміщеннями руки в циліндричній системі координат, циклове керування рухами кисті і затискання-відтискання схвата, а також подачу команд пуску циклів роботи верстатів, іншого технологічного обладнання і прийому відповідних команд після виконання цих циклів. Можливі три режими роботи ПР:

1) “навчання” – повернення в нульову точку, ручне керування і крокове переміщення по кожній осі координат, введення в пам'ять заданих зна-

чень координат, швидкості переміщень, кількості оброблюваних деталей (циклів) і інше;

2) “автомат” – автоматичне керування по заданій програмі;

3) “редагування” – підготовка і коректування даних керування роботом.

Типовий робочий цикл ПР при зміні заготовки на токарному верстаті з ЧПК містить у собі наступні етапи: підведення руки ПР до патрона верстата – захоплення обробленої деталі – відведення руки у вихідну точку – підведення руки до тактового столу – опускання деталі – захоплення чергової заготовки – підведення заготовки до патрона верстата – звільнення заготовки після затиску її в патроні – відведення руки у вихідну точку – початок циклу обробки на верстаті.

Найбільша кількість одночасно керованих координатних переміщень може бути: 1 – у режимі позиціювання (електродвигуна повороту, чи опускання висування руки) чи 2 – у режимі циклового керування (пневмодвигуна блоку повороту кисті руки і схвата).

Принципові (кінематична і пневматична) схеми ПР приведені на рисунку 23 [1]. Діапазони переміщень маніпулятора по координатних осях також показані на рисунку 23 [1]. До складу пневмообладнання ПР включені блоки приводів повороту кисті руки (виконання 1 і 2). Робота блоку повороту (виконання 1) відбувається таким чином. При відключених електромагнітах YA1 і YA2 повітря по магістралях 1, 2 і 3 надходить в обидві порожнини пневмодвигуна Д2. При включеному електромагніті YA1 пневморозподілювача Р4 і відключеному електромагніті YA2 відбувається поворот кисті руки проти годинникової стрілки щодо поздовжньої осі. При цьому повітря з порожнини пневмодвигуна М2 витісняється по магістралі 2 через пневморозподілювач Р4 в атмосферу. При включенні електромагніта YA2 пневморозподілювача Р5 і відключеному електромагніті YA1

аналогічно відбувається поворот кисті руки за годинниковою стрілкою.

Блок повороту (виконання 2) працює таким чином. Поворот кисті руки (щодо її поздовжньої осі) за годинниковою стрілкою здійснюється при включеному електромагніті YA3. При цьому повітря в пневмодвигун М1 надходить через розподільник Р4 по магістралі 5. При включеному електромагніті YA5 відбувається прискорений поворот кисті: повітря з пневмодвигуна М1 витісняється в атмосферу через розподільники Р5, Р3, дросель ДР і глушник Г2 по магістралях 6, 4, 7. При відключеному електромагніті YA5 відбувається уповільнений поворот блоку: повітря з пневмодвигуна Д1 витісняється в атмосферу тільки через розподільник Р5 і дросель ДР по магістралях 6, 4.

Поворот кисті руки проти годинникової стрілки здійснюється при включеному електромагніті YA4. При цьому повітря в пневмодвигун М1 надходить через розподільник Р5 по магістралі 6. При включеному електромагніті YA5 відбувається прискорений поворот блоку: повітря з пневмодвигуна М1 витісняється в атмосферу через розподільники Р4, Р3, дросель ДР і глушник Г2 по магістралях 5, 4, 7. При відключеному електромагніті YA5 відбувається уповільнений поворот блоку: повітря з пневмодвигуна М1 витісняється в атмосферу тільки через розподільник Р4 і дросель ДР по магістралях 5, 4.

Поворот кисті руки щодо її поперечної осі вправо здійснюється при включеному електромагніті YA1. При цьому повітря в пневмоциліндр Ц надходить по магістралі 2 і витісняється з пневмоциліндра Ц по магістралі 3 через розподільник Р2. Поворот кисті руки вліво здійснюється при включеному електромагніті YA2. При цьому повітря в пневмоциліндр Ц надходить по магістралі 3 і витісняється з пневмоциліндра по магістралі 2 через розподільник Р1.

Механізм повороту ПР М20П.40.01 виконаний у вигляді автономного

вузла, показаного на рисунку 24 [1]. На основі 1 робота кріпиться черв'ячний редуктор 2, з'єднаний через зубцювату муфту 3 з електродвигуном 4. На вихідному валу черв'ячного редуктора встановлена зубцювата шестірня 5. Вона входить у зачеплення з циліндричним зубчастим колесом 6, яке з'єднано з валом 7. Таким чином, обертання електродвигуна постійного струму через черв'ячний редуктор і пару циліндричних прямозубих шестірень передається валу 7, який служить опорою для механізму підйому й опускання руки. Контроль кута повороту для керування швидкістю здійснюється за допомогою шляхових перемикачів 8.

Механізм підйому й опускання руки, виконаний у вигляді окремого вузла, показаний на рисунку 25 [1]. Корпус 1, що включає в себе механізм висування руки, переміщається нагору і вниз по двох направляючих 2, які закріплені у верхній і нижній опорних плитах 3 і 4. На верхній опорній плиті 3 встановлена піддвигунова основа 5, в середині якої знаходиться електромагнітне гальмо 6. Електродвигун 7 постійного струму, встановлений на піддвигуновій основі 5, через зубцювату муфту 8 з'єднаний з кульковим гвинтом 9. Гайка 10 кулькової гвинтової пари закріплена в корпусі 1 вузла висування руки. Таким чином, обертання електродвигуна 7 перетворюється в поступальний рух руки нагору або вниз. Для захисту гвинта від пилу і бруду використовується гофрована оболонка 11. Гумові амортизатори 12 дозволяють пом'якшити удар наприкінці ходу руки у верхнім і нижнім положенні. Для керування швидкістю переміщення використовуються шляхові перемикачі, які наїжджають на упори 13 та 14.

Механізм висування руки показаний на рисунку 26 [1]. До задньої стінки корпуса 1 прикріплений кронштейн 2, на якому встановлений електродвигун 3 постійного струму. Обертання електродвигуна через зубцюватий пас 4 передається гвинту 5 кулькової гвинтової пари. Гайка 6 кулькової гвинтової пари з'єднана з кронштейном 7. До верхнього кінця цього крон-

штейна прикріплена качалка 8, яка переміщається вперед чи назад у втулці 9. Нижній кінець кронштейну 7 з роликами рухається по направляючій 10, яка виключає поворот качалки 8. В середині пустотілої качалки проходить трубка 11 для подачі стиснутого повітря у внутрішню порожнину, а відтіля до механізму повороту кисті і схвата. Амортизатори 12 пом'якшують удар наприкінці ходу руки – качалки 8. Контроль положення руки для керування швидкістю переміщення здійснюється за допомогою шляхових перемикачів 13. Довжина ходу качалки 8 залежить від виконання механізму висування руки.

Блок повороту (кисть руки) може бути виконаний у різних виконаннях: 1 чи 2. Конструкція блоку повороту – кисті руки (виконання 1) показана на рисунку 27 [1]. Поворотний блок змонтований у корпусі 1 і складається з механізму обертання кисті щодо поздовжньої осі з двопозиційним керуванням при русі від упора до упора і механізму повороту (колихання) щодо поперечної осі. Обертання кисті руки здійснюється від неповноповоротного пневмодвигуна 2, на передньому торці якого кріпиться стакан 3. На зовнішній поверхні стакана встановлені розподільчі пневмопристрої 4 привода повороту кисті, а усередині на підшипниках – втулка 5 із фланцем для кріплення змінного схвата. Втулка 5 зв'язана з валом пневмодвигуна 2 за допомогою шпонки. На протилежному кінці поворотного блоку знаходиться стопорний механізм 6, який жорстко з'єднаний з валом пневмодвигуна 2. У залежності від установленого змінного двоплечого важеля стопорного механізму можна обмежувати кутове положення кисті в діапазоні від 0 до ±90° (варіант 1) чи від 0 до 180° (варіант 2). Усередині корпуса 1 змонтовано амортизатор 7, що зменшує удар при підході стопора 6 до перестав них упорів 8, а також мікроперемикачі 12 кінцевих положень. У поворотному блоці є також механізм ручної установки кута хитання кисті щодо її поперечної осі 9 (у межах ±3,5°). Кут хитання регулюється гвинтами 10,

які встановлені в нерухливому фланці 11.

Конструкція блоку повороту – кисті руки (виконання 2) також показана на рисунку 27 [1]. Обертання вала пневмодвигуна 1 через приводний плоскозубчатий пас 2 передається на вал 3. На цьому ж валу встановлене електромагнітне гальмо 4, що забезпечує швидку зупинку обертання кисті руки. Обертання вала 3 при включеному електромагніті, який розгальмовує вал, передається генератору 5 зубцюватого хвильового редуктора, що дозволяє збільшити вихідний крутний момент приводу. Для контролю кутового положення блоку повороту використовуються два безконтактних датчики 6, 7 крайніх і проміжних позицій, імпульси від яких надходять у пристрій керування ПР. Дозволяюча здатність датчиків дорівнює 0,1°, що забезпечує стабільність установки кута повороту кисті в межах ±0,3° в усьому діапазоні від –90 до +180°. Поворот (хитання) кисті щодо поперечної осі в межах ±3,5° здійснюється від пневмоциліндра 8, шток якого зв'язаний з кулачком 9, що має похилий паз. За допомогою кулачка шток діє на палець 10 у кронштейні 11 корпуса блоку повороту. Кутове положення контролюється шляховими мікроперемикачами 12 і регулюється упорами 13.

Механізм схвата маніпулятора (виконання С1) уніфікованої конструкції показаний на рисунку 28 [1]. Конструкція схвата з пневмоприводом, використаного у ПР М20П для деталей фланцевого типу, показана на рисунку 28 [1]. Схват оснащений трьома плоскими губками 1, які закріплені на важелях 2 під кутом 120° відносно один одного. Привод губок схвату – пневматичний. Повітря з заводської пневмомережі подається в коробку пневмозолотників 3 і через пневмоклапан 4 швидкі вихлопи в одну з порожнин пневмоциліндра 5: праву – при затиску, а ліву – при розтисканні деталі. Поступальне переміщення штока 6 пневмоциліндра здійснюється за допомогою важеля 7 і шарнірних паралелограмів 8 у радіальний (стосовно

деталі, що затискається) рух важелів 2 лівої і верхньої губок. Привод правої нижньої губки 2 здійснюється за допомогою пари зубчатих коліс 9, 10 і шарнірного паралелограма 11. Введення в кінематичний ланцюг шарнірних паралелограмів забезпечує збереження кутового положення важелів 2 затискних губок, які центрують деталь при її зажимі. Осі поворотних важелів закріплені у верхній 12 і нижній 13 паралельних плитах, які зв'язані між собою планками 14 та утворюють твердий каркас. Каркас з затискними губками кріпиться до основи 15 за допомогою трьох стрижнів 16 і пружин 17, що компенсують погрішності позиціювання схвата з заготовкою при установці її, наприклад, у патрон верстата. Кріплення схвата до механізму кисті руки здійснюється за допомогою кронштейна 18. При неприпустимій деформації кронштейна (наприклад, при упорі схвата в яку-небудь перешкоду) підпружинений упор 19 натискає на кінцевий мікровимикач 20, що подає аварійний сигнал в пристрій керування ПР. Електрокабель підключається через роз'їм 21. Контроль стану затиску або розтискання деталей губками схвата здійснюється відповідно датчиками 22 і 23. Датчики представляють собою кінцеві мікровимикачі, на які впливають плоскі пружини, що упираються у важіль шарнірного паралелограма 11.

### 4.5 Режими роботи промислового робота М20П.40.01

Керуючий пристрій працює в 4-х основних режимах: навчання і редагування, відтворення, діагностика і робота з зовнішнім запам'ятовуючим пристроєм ЗЗП "Ізодиск".

Після первісного включення система встановлюється в режим "Діагностика". Рекомендується робити первинне нулювання робота. За допомогою ручного руху кожний із суглобів потрібно довести до положення, близького до нульового, перш ніж запустити функцію "Встановлення в нульовій позиції" кнопкою "Старт".

Не можна переходити до режимів навчання, редагування і виконання

програм перед тим, як нулювати робот. Ручний рух регулювання робота виконується незважаючи на обмеження припустимої зони руху, тому повинні виконуватись дуже уважно оператором.

Робота в режимі “Робота з ЗЗП” не залежить від стану робота (нулювати чи ні).

Режим ***“Навчання і редагування”***

У цьому режимі можна виконувати наступні функції:

- запис, корекцію, перейменування і стирання програм;
- запис, корекцію і стирання кроків;
- запис, корекцію і стирання точок;
- ручний рух робота;
- виконання однієї команди;
- заповнення, корекцію і стирання таблиць штабелювання.

Режим ***“Відтворення”***

У цьому режимі виконуються наступні функції:

- запуск і зупинка технологічної програми;
- підключення до керування і звільнення від керування машини;
- включення повітряного струменя;
- відкриття і закриття щита машини;
- покрокове виконання програми;
- виконання G - команд програми;
- виконання M - команд програми;
- діагностичні функції;
- вибір програми і номера команди початку інтерпретації.

Режим ***“Діагностика”***

У цьому режимі виконуються наступні функції:

- установлення робота в нульовій позиції;
- читання і корекція параметрів настроювання;

- скидання аварійної помилки;
- ручний рух;
- вивід робота з зони крайнього вимикача;
- перекидання базових величин параметрів з постійної пам'яті в CMOS-пам'ять;
- автоматична корекція сумарної помилки при керуванні осями;
- ініціалізація параметрів, необхідних для сервокерування.

Режим ***"Робота з ЗЗП"***

У цьому режимі виконуються наступні функції:

- форматування гнучкого магнітного міні-диска;
- запис файлу на гнучкому диску;
- читання файлу з гнучкого диска;
- порівняння файлу гнучкого магнітного диска з файлом пам'яті;
- стирання файлу з гнучкого диска;
- читання існуючих на гнучкому диску файлів.

### 4.6 Програмування, види команд

Програмування промислового робота М20П.40.01 аналогічно програмуванню промислового робота М10П.62.01.

Зведена таблиця команд представлена в таблиці 8 [1].

### 4.7 Порядок виконання роботи

1) Вивчити призначення, конструкцію і принцип дії промислового робота М20П.40.01.
2) За заданим завданням скласти програму керування роботом М20П.40.01.
3) Увести розроблену програму в систему керування.
4) Вивчити різні режими роботи робота.
5) Проаналізувати отримані результати і зробити висновки по лабораторній роботі.

### 4.8 Зміст звіту

Звіт по лабораторній роботі повинний містити:

1) Найменування роботи.
2) Мета роботи.
3) Технічна характеристика робота М20П.40.01.
4) Кінематична схема промислового робота М20П.40.01 і короткий її опис.
5) Основні вузли і їхня робота (з рисунками).
6) Основні режими роботи робота.
7) Порядок виконання роботи.
8) Програма керування і схема руху по заданій програмі.
9) Висновки по роботі.

# ЛАБОРАТОРНА РОБОТА №5
# 5 ВИВЧЕННЯ КОНСТРУКЦІЇ ТА ОСНОВ ПРОГРАМУВАННЯ ПРОМИСЛОВОГО РОБОТА РМ-01 З КОНТУРНОЮ СИСТЕМОЮ КЕРУВАННЯ “СФЕРА-36”

## 5.1 Мета роботи

Вивчити кінематичну схему і конструкцію маніпулятора, принцип дії і блок-схему системи керування роботом, а також системи координат. Ознайомитися з режимами навчання робота і ручного керування.

## 5.2 Призначення промислового робота РМ-01

Робот РМ-01 – це універсальний електромеханічний промисловий робот з керуванням від ЕОМ і має антропоморфорну руку із шістьма обертальними парами (суглобами). Робот призначений, у першу чергу, для виконання різноманітних складальних і несилових технологічних операцій (нанесення лаків, мастик, регулювання приладів, виміру вільних розмірів, фінішна обробка складних просторових деталей, зачищення заусенців, свердлення отворів у м'яких матеріалах і т.д.)

## 5.3 Технічна характеристика маніпулятора

Кількість ступенів рухливості ..... 6

Кількість одночасно керованих координат ..... 6

Кількість одночасна програмувальних координат ..... 3

Привод – серводвигун постійного струму з захисними електромагнітними гальмами і датчиком зворотного зв'язку

Найбільша вантажопідйомність (включаючи оснащення) ..... 2,5 кг

Похибка позиціонування – не більш ..... ±0,1мм

Статичне зусилля в робочій точці оснащення – не більш ..... 60 Н

Швидкість переміщення інструмента (оснащення):

по прямолінійній траєкторії – не більш ..... 0,5 м/с

Робочий простір – сферичний з радіусом ..... 920 мм

Привод оснащення (схвата) ..... пневматичний з керуванням по програмі вручну

Маса маніпулятора ..... 53 кг

***Пристрій керування – "Сфера-36"***

Принцип побудови ..... контурний, дворівневий мікропроцесорний пристрій на базі ЕОМ "Електроніка-60"

Структура ..... блочно-модульна на базі БІС

Спосіб розрахунку точок ..... лінійна інтерполяція

Мова програмування ..... ARPS

Периферійний пристрій для навчання і програмування – відеотермінальний пристрій (ВТП) на коді ASCII (стандарт США) і пульт ручного керування (ПРК)

Ємність пам'яті для керування керуючих програм і точок, К байт:

операційної ..... 24

НГМД ..... 64

Входи і виходи (для зовнішнього обладнання й оснащення) ..... 32

Дискретів гальванічно розв'язаних входу: (24 В, 15 ма, пост. струм.) ..... 32

Дискретів гальванічно розв'язаних виходу: ..... (24 В, 2 А, пост. струм.)

Живлення ..... 220 В, 50 Гц, 1 фаза, не більш 2 кв·А

Маса шафи СК ..... 280 кг

Робоча температура навколишнього середовища ..... + 5°... + 35°С

## 5.4 Конструкція маніпулятора

### 5.4.1 Конструкція і принцип дії маніпулятора

Промисловий робот РМ-01 складається з маніпулятора PUMA-560 – 2, системи керування ЕОМ “СФЕРА-36”–1 і 2-х сполучних кабелів – 3 (рисунок 29 [1]).

Ланки маніпулятора обертаються одна відносно одної (рисунок 31 [1]). Обертання колони (1-й суглоб) здійснюється від серводвигуна М1 (рисунок 33[1]) у нижній частині колони через циліндричну зубчасту передачу $z_1$ -$z_2$ на вал-шестірню $z_3$, що входить у циліндричне зубцюване зачеплення з зубцюватим вінцем $z_4$.

Обертання передпліччя (2-й суглоб) здійснюється від серводвигуна М2, розташованого усередині передпліччя через зубцювату конічну передачу $z_5$-$z_6$ з вала двигуна на вал-шестірню $z_7$ і через циліндричну зубчасту передачу $z_7$-$z_8$ на зубцюватий вінець $z_8$, укріплений на плечі рухливої колони.

Обертання ліктя (3-й суглоб) здійснюється від серводвигуна М3 розташованого поруч із двигуном М2, через сильфонні (гнучкі) муфти 4 і твердий вал на вал-шестірню $z_9$, далі – через конічне зубцюване зачеплення $z_9$ -$z_{10}$ на вал-шестірню $z_{11}$, що входить у циліндричне зубцюване зачеплення $z_{11}$ -$z_{12}$ із зубцюватим вінцем $z_{12}$, укріпленим на лікті. Муфти 4 за рахунок пружних деформацій забезпечують передачу обертання під кутом, тобто утворюють карданний механізм.

Обертання кисті (4-й суглоб) здійснюється від серводвигуна М4, встановленого у верхній частині ліктя, через сильфонну муфту, вал, сильфонну муфту, далі – через двоступінчастий зубцюватий редуктор, що складається з 2-х циліндричних зубцюватих зачеплень $z_{13}$-$z_{14}$ і $z_{15}$-$z_{16}$. Вал-шестірня $z_{13}$ входить у зачеплення зі східчастою шестірнею $z_{14}$-$z_{15}$ , яка зачіпається зуб-

чатим вінцем $z_{16}$, який має отвір для проходу проміжних валів 5-го і 6-го суглобів.

Хитання кисті (5-й суглоб), розташованої на консолі щодо ліктя, здійснюється рухливим сегментом кисті від серводвигуна М5, що знаходиться в лікті, через сильфонні муфти і двоступінчастий редуктор, що складається з циліндричного зубцюватого зачеплення $z_{17}$-$z_{18}$ і конічного зубцюватого зачеплення $z_{19}$-$z_{20}$. Конічна шестірня $z_{20}$ розташована співвісно з віссю хитання сегмента кисті і нерухомо до нього прикріплена (рисунок 34 [1]).

Обертання фланця кисті, розташованої на сегменті кисті (6-й суглоб), здійснюється від серводвигуна М6, що знаходиться в лікті, і, подібно 5-му суглобу, через сильфонні муфти і двоступінчастий редуктор передається на фланець. Редуктор складається з двох пар конічних зубцюватих зачеплень $z_{21}$-$z_{22}$ і $z_{23}$-$z_{24}$. Конічна шестірня $z_{22}$ розташована на осі хитання сегмента кисті (тому при повороті сегмента завжди залишається в зачепленні) і передає усередину сегмента кисті через вал-шестірню $z_{24}$ обертання на вал фланця 1, що має конічну шестірню $z_{23}$. Фланець 1 розташований на консолі і призначений для кріплення оснащення.

Ступіні рухливості маніпулятора позначені на рисунку 32 [1] і має наступні обмеження по куту повороту:

| | |
|---|---|
| 1-й суглоб (колона) | 320° |
| 2-й склад (передпліччя) | 266° |
| 3-й суглоб (лікоть) | 284° |
| 4-й суглоб (поворот кисті) | 280° |
| 5-й суглоб (хитання сегмента кисті) | 200° |
| 6-й суглоб (обертання фланця кисті) | 520° |

Маніпулятор може мати дві конфігурації: права рука (2-й суглоб нахилений у позитивному напрямку) і ліва рука 2-й суглоб нахилений у негативному напрямку). Ці конфігурації легко установити, представивши свою

руку на місці руки робота.

Маніпулятор може використовуватися як у напольному варіанті, так і в підвісному (колона розташована вертикально, рука вниз).

Характерною рисою кінематики маніпулятора є наявність у кожнім приводі суглобів понижуючого двоступінчастого редуктора, конструктивно виконаного в корпусах суглобів. Повороти 3-го, 4-го, 5-го і 6-го суглобів передаються через сильфонні муфти під кутом через усю довжину 2-го і 3-го суглобів (передпліччя і ліктя). При повороті 4-го суглоба (кисті) вали сильфонних передач 5-го і 6-го суглобів між собою можуть перекручуватися за рахунок пружних деформацій. Тому поворот 4-го суглоба обмежений 200°. З погляду кінематики оригінальна конструкція кисті. Обертання на 6-ий суглоб передасться через два обертових суглоби (4-й і 5-й) завдяки трьом співвісним конічним зачепленням і сильфонним (пружним) муфтам.

Принцип дії маніпулятора заснований на одночасному, погодженому керуванні серводвигунами шести суглобів по замкнутому ланцюжку (ЕОМ – підсилювач потужності – електродвигун – датчик зворотного зв'язку – ЕОМ) з однієї сторони і з іншого боку – від серводвигуна через трансмісію (сильфонні муфти і зубцюваті редуктори) на виконавчий орган по розімкнутому ланцюзі.

Положення ланок маніпулятора під час руху контролюється фотоімпульсними електричними датчиками, які установлені на валах серводвигуна. Поточне положення маніпулятора визначається щодо відомого вихідного (абсолютного) положення ланок. Визначення абсолютного положення ланок (калібрування маніпулятора) здійснюється за допомогою потенціометричних датчиків, що мають аналогові виходи (напруга – від 0 до 5В).

Центральний процесор, одержуючи інформацію від потенціометрів, визначає приблизне положення ланок. Інформація, що надходить від фотоімпульсних датчиків, дозволяє системі керування уточнити положення ко-

жної ланки маніпулятора. Це відбувається в такий спосіб. По команді CAL кожна ланка повертається на необхідний кут до одержання індексного імпульсу з фотоімпульсного датчика, таким чином, потенціометричний датчик визначає сектор, у якому знаходиться ланка, а при повороті ланки шукається реперна точка в цьому секторі, що визначається одержанням індексного імпульсу.

Крім індексних імпульсів (імпульс на один оберт вала датчика) з фотоімпульсного датчика надходять ще дві послідовності імпульсів, що мають однакову частоту (значно велику частоту індексних імпульсів), але зрушених відносно один одного на 1/4 періоду. Логічна обробка цих двох послідовностей імпульсів дозволяє визначити фактичне збільшення положення з урахуванням напрямку руху, а також відносну швидкість ланки маніпулятора.

Серводвигуни оснащені гальмами з електромагнітним керуванням. При включенні живлення приводів маніпулятор блокується в тім положенні, у якому він знаходився в момент включення живлення.

### 5.4.2 Система координат маніпулятора

Для забезпечення зручності програмування (розрахунку точок, навчання, ручного керування) маніпулятор крім природної системи координат (рисунок 32 [1]) по осях (кути повороту суглобів) має ще дві: основна система координат WORLD) (рисунок 36 [1]) і система координат інструмента, “плаваючу” (рисунок 37 [1]).

***Основна система координат***

Складається з трьох перпендикулярних друг до друга осей (XYZ), що перетинаються в центрі плеча маніпулятора. При цьому вісь Z збігається з вертикальною віссю колони.

Основна система координат установлюється при ручному керуванні за допомогою ПРК одноразовим натисканням кнопки “WORLD” і індикацією світлодіода поруч.

Основна система координат не рухається при переміщенні ланок.

Системою WORLD зручно користатися при навчанні робота, переміщаючи інструмент (схват) паралельно його осям, натискаючи кнопки на пульті ручного керування: – X + , – Y + , – Z + (рисунок 42 [1]).

***Система координат інструмента***

Система TOOL складається з трьох осей X', Y', Z', що перетинаються в центрі фланця кисті маніпулятора. Система координат TOOL – “плаваюча”, тобто рухається при переміщенні суглобів руки й орієнтується за допомогою кутів повороту 0, a, t щодо осей системи WORLD (рисунки 37, 38 [1]). Для установки системи TOOL необхідно натиснути кнопку “TOOL” (індикація світлодіода поруч). Причому вісь Z' системи TOOL проходить перпендикулярно фланцю, по осі інструмента (схвата). Системою TOOL також ефективно користатися при навчанні робота.

***Система координат по осях***

Це природна система координат маніпулятора (повороти суглобів 1, 2, 3, 4, 5 і 6). Установлюється кнопкою “JOINT” на пульті ручного керування і приводиться в дію відповідно кнопками: – 1 + , – 2 + , – 3 + , – 4 + , – 5 + , – 6 +. При цьому по черзі обертається який-небудь суглоб (при одночасному натисканні кнопок - кілька суглобів). Зручна при навчанні робота, особливо коли необхідно переміщати інструмент на великі відстані в робочій зоні.

### 5.4.3 Система керування “Сфера-36”

Система керування розміщена у виді окремих блоків і модулів у шафі з двома начіпними дверима (рисунок 39 [1]). Вона складається з наступних основних частин: ЦЕОМ, периферійних пристроїв, підсилювачів потужності, блоків живлення, ПРК, пульта (панелі) оператора, доступ до яких можливий з передніх дверей, і блоків входів-виходів (доступ – через задні двері).

Дисплей із клавіатурою і відблиски пам'яті утворить відеотермінальний пристрій (ВТП). Блок-схема системи керування показана на рисунку 40 [1]. Вона має блочно-модульну структуру і конструктивно складається з наступних блоків:

МЦП – модуль центрального процесора;

ОЗП – оперативне запам'ятовуючий пристрій;

ПЗП – постійне запам'ятовуючий пристрій;

МАВ – модуль аналогових входів;

МЗ – модуль зв'язку;

МПІ – модуль послідовного інтерфейсу;

МВВ – модуль входів-виходів;

НГМД – накопичувач на гнучких магнітних дисках;

ВТП – відеотермінальний пристрій;

ПРК – пульт ручного керування;

МПП – модуль процесора привода;

МКП – модуль керування приводом;

ШІП – широтноімпульсний перетворювач (підсилювач потужності);

ПК – пульт керування;

БЖ – блок живлення;

БВ – блок вентиляторів;

БРФ – блок розв'язки і фільт- рації;

М – серводвигун;

П – потенціометр;

Т – теплообмінник;

БПП – блок живлення привода.

Центральна ЕОМ (ЦЕОМ) є головним модулем системи (на блок-схемі це БМ ЦЕОМ – блок модулів ЦЕОМ) і складається з двох рівнів: верхнього (власне ЦЕОМ) і нижнього (модулі керуванні приводами). Верхній рівень ЦЕОМ обчислює траєкторію руху фланця маніпулятора, логічно обробляє інформацію про стан пристроїв і синхронізує їхню роботу, забезпечує діалог оператора з пристроєм керування. Обмінюється інформацією з периферійними пристроями ВТП, НГМД, ПРК, МВВ), калібрує маніпулятор, обчислює діагностику СК і керує сервоприводами через ПРК. ЦЕОМ складається з наступних блоків:

МЦП – модуль прийому, обробки і видачі результатів обчислень єдиної на інші модулі;

МПІ – модуль для зв'язку з периферійними пристроями;

МС – модуль рівнобіжного інтерфейсу для зв'язку з МПП;

ПЗП – модуль, що зберігає в постійній пам'яті транслятор мови;

ARPS – (зміст ПЗП не стирається при вимиканні);

ОЗП – пам'ять для збереження програм користувача і координат точок мовою ARPS (ОЗП має резервне живлення від акумуляторів, завдяки чому його зміст зберігається при вимиканні живлення);

МАВ – модуль для перетворення аналогових сигналів потенціометричних датчиків у цифровий код.

Нижній рівень керування призначений для рішення задачі регулювання параметрів руху (положення, швидкість) ланок маніпулятора відповідно до програм руху, які формовані верхнім рівнем керування і є керуючою частиною електроприводів, що стежать.

Модуль керування приводом (МКП) служить для зворотного зв'язку із серводвигуном через ФІД і для формування імпульсів у ШІП. МПП служить для розрахунку координат проміжних точок по алгоритму, сформованому МЦП, МПП, МКП, ШІП і утворює нижній рівень. ШІП формує могутні сигнали безпосереднього керування електродвигунами постійного току і мають електронну схему захисту сервоприводов від перевантаження.

Панель керування служить для включення живлення ЕОМ і приводів, вибору режиму роботи й аварійної зупинки робота в позаштатній ситуації.

ВТП служить для діалогу оператора з ЦЕОМ, наборі і редагування програм користувача і навчання робота.

БП здійснює електроживлення стійки керування, містить запобіжники головний контактор, фільтр шуму і рознімання для нестатків техобслуговування.

БПП генерує необхідний для роботи серводвигунів постійного струму (40 В).

БВ і Т служать для підтримування усередині стійки керування необхідної температури.

ПРК служить для ручного керування маніпулятором і навчання робота.

Блоки входів-виходів необхідні для зв'язки з зовнішніми пристроями (одержання від них сигналів і керування ними шляхом подачі сигналів). За допомогою вхідних і вихідних каналів забезпечується гальванічно розв'язаний зв'язок робота з кінцевими і ручними вимикачами, сигнальними лампами, вентилями, програмувальними логічними контролерами, ЕОМ, верстатами, пристосуваннями, іншими роботами. Вхідні і вихідні сигнали можуть бути тільки on/off (“0” та “1”). Через вхідні канали роботу можна дати, наприклад, команду в окрему адресу програми і т.д. Також можна повідомити роботу про надходження в робочу зону чергового виробу, щоб

робот міг почати роботу в потрібний момент часу. Через вихідні канали робот може керувати роботою пристосувань верстатів, конвеєром і т.д.

НГМД служить для запису і зчитування програм користувача на гнучкі магнітні диски (магнітодиски по 64 КБайт). Використання дисків дає можливість створити бібліотеку програм робота.

### 5.5 Програмування і настроювання робота

#### 5.5.1 Запуск робота

Запуск робота здійснюється в наступній послідовності:

1. Включити живлення “СУ”.

2. Натиснути кнопку “СУ 1” на пульті керування і через 3 секунди система керування включається і на екрані дисплея висвітиться:

“NOKIA ARPSIN BOS.RM-01

Zero memory (Y, N) or AUTOSTART ?”

3. Відповісти на запитання: Y чи N (можна просто натиснути RETURN) чи натиснути кнопку “АВТОСТАРТ”.

Y – стерти зміст ОЗП (що зберігаються в пам'яті програми користувача за рахунок резервного живлення ОЗП від акумуляторів).

N – зберегти програми в ОЗП.

AUTOSTART – автоматичний запуск робота при наявності готових програм користувача на диску (дискету необхідно вставити в НГМД) з використанням пускової програми START.

4. Якщо не обраний режим AUTOSTART, то включити живлення приводів кнопкою “Привод-1”.

5. Виконати калібрування робота введенням через ВТП команди CAL.

**Увага**: Робот повинний знаходитися на відстані від перешкод і мати положення суглобів, які не граничать із крайніми положеннями суглобів, у противному випадку відбудеться фатальна помилка в МЦП, і запуск при-

йдеться повторити, виконавши вище зазначені вимоги.

6. Далі можна робот запускати по програмі командою RUN, навчати і вводити нову програму. Вимикання робота здійснюється в зворотній послідовності: “Привода-0”, “СУ-0”, “Сеть”.

### 5.5.2 Ручне керування роботом

Пульт ручного керування (ПРК) використовується для позиціонування маніпулятора і навчання (рисунок 42 [1]). Він має 5 режимів роботи:

- COMP – передача керування ЕОМ (робота з ВТП) – позиція 3 (рисунок 42 [1]).
- WORLD – керування прямолінійними переміщеннями кисті за допомогою клавіші X, Y, Z (позиція 14), щодо осей основної системи координат і обертанням фланця кисті, щодо осей системи координат X, Y, Z за допомогою клавіші: RX, RY, RZ (позиція 14).
- TOOL – те ж саме, але щодо системи координат інструмента (позиція 6).
- JOINT – керування поворотами суглобів за допомогою клавіш: 1, 2, 3, 4, 5 і 6 (позиція 5).
- FREE – відключення приводів (позиція 7) з відключенням гальм. Обраний режим індицирується сигнальною лампою. Для зупинки виконання програми служить кнопка OFF (позиція 2).

Разжим і затиск схвата маніпулятора (подача стиснутого повітря) здійснюється відповідно кнопками OPEN і CLOSE (позиції 11 і 12).

Установка швидкості здійснюється кнопками “+ – SPEED” (позиції 9 і 10) і контролюється по загорянню світлодиодів індикатора (позиція 8). Перший сегмент індикатора відповідає швидкості 1,9 мм/с, а далі швидкість подвоюється.

При навчанні робота в режимі TGO (TGOS) використовується кнопка

STEP (позиція 13) для запису в пам'ять (ОЗП) координат тих точок, яких навчають, і автоматичного запису рядків програми.

Світлодиод CALIB (позиція 15) індицирує відсутність калібрування робота (якщо він включений, то калібрування не виконане).

Індикатор стану системи (позиція 15) висвітлює узагальнення центрального процесора.

Кнопка на торці ПРК (позиція 1) служить для екстреного переривання виконуваної програми з виключенням живлення приводів.

### 5.5.3 Схема керування роботом

При запуску керуючої програми оператором по команді RUN центральний процесор починає обчислювати, передбачену програмою, траєкторію руху фланця маніпулятора. Значення, обчислені на основі поточного положення, передаються в модулі керування привода. Нові значення обчислюються завчасно десятки разів у секунду. Таким чином, принцип керування рухом маніпулятора полягає в тому, що при русі маніпулятора від однієї запрограмованої точки в іншу йому дається трохи “проміжних точок” (лінійна інтерполяція траєкторії руху). У залежності від команди маніпулятор може переміщатися по прямолінійній траєкторії (команда GOS) чи по вільно інтерпольованій (команда GO), тобто по криволінійній, яка сформувала МЦП.

Для кожного ступеня рухливості передбачений індивідуальний МКП і ШІП, що забезпечують керування відповідною ланкою маніпулятора виходячи з отриманої від центрального процесора інформацією. Необхідний для керування зворотній зв’язок забезпечується встановленими на серводвигунах фотоелектричними імпульсними датчиками. Таким чином, створюються між МКП та двигунами замкнуті ланцюги керування. Розрахунок нових проміжних значень (координат точок) МПП здійснюється набагато

швидше, ніж у МЦП, приблизно 1000 разів у секунду. Застосований спосіб руху суглобів (лінійна інтерполяція) забезпечує максимальну плавність руху.

### 5.5.4 Навчання робота

Навчання робота здійснюється двома різними способами і заключається в записі в пам'ять координат положення маніпулятора, досягнутого за допомогою ручного керування.

При першому способі на рівні редактора системи (вхід у редактор командою: EDIT < ім'я програми >) встановлюється режим TGO, чи TGOS набором команди : TGO < ім'я точки з індексом > (наприклад “TGOAL”), і маніпулятор за допомогою ПРК переміщається з точки у точку. Для запам'ятовування координат точки (положення фланця руки) необхідно натиснути кнопку “STEP” на ПРК в цьому положенні маніпулятора. При другому способі навчання також за допомогою ПРК позиціонуємо маніпулятор у необхідну точку, у цьому положенні набирає команду: HERE < ім'я точки > і за допомогою її вводимо в ОЗП координати точки (використовується ВТП).

При першому способі навчання робот повторює в програмі точки в строгій послідовності, з якою його навчили. При другому способі послідовність навчання будь-яка. При першому способі відбувається автоматичний запис команд переміщення в навчальну точку програм.

Найбільш ефективний спосіб навчання – комбінований (комбінація описаних вище).

Завдання координат точок може бути й аналітичним за допомогою команд MOVE (MOVES). При цьому, точка не має ім’я, а утворюється як прирощення попереднього положення маніпулятора. Даний спосіб вимагає математичних методів обчислення координат необхідних точок у тривимірному просторі оператором.

### 5.5.5 Програмування промислового робота РМ-01

***Вступ***

*Система керування ARPS*

ARPS (Advanced Robot Programming System, Удосконалена система програмування робота) – система на базі обчислювальної машини, призначена для керування роботом таким чином, що завдання виду роботи, виконуваної роботом, здійснюється за допомогою введення програм в ЕОМ. Можливість програмування, забезпечувана системою ARPS, дозволяє навчити робот швидко й акуратно виконувати прості та складні операції.

Система керування ARPS включає центральну ЕОМ, відеотермінал, накопичувач на гнучких магнітних дисках (НГМД), переносний пульт ручного керування і лінії входу/виходу. Програмування здійснюється шляхом запису з клавіатури відеотермінала інструкцій, призначених для керування роботом. При необхідності наступного використання програми можна записати на гнучкий магнітний диск. Пульт ручного керування використовується для навчання робота програмувальним точкам позиціонування. За допомогою ліній входу/виходу робот може керувати різним устаткуванням, наприклад, конвеєрами, верстатами, зварювальними установками і т.д.

Операційна система ARPS постійно зберігається в програмувальному постійно запам'ятовуючому пристрої (ППЗП) пристрою керування. Вона містить інструкції керування роботом, а також ряд допоміжних функцій, за допомогою яких виконується програмування робота методом навчання, запис даних на гнучкий магнітний диск і т.д. Забезпечена можливість складання нових програм під час роботи робота. Доцільно скласти ряд програм загального використання (бібліотеку підпрограм), наприклад, для операцій завантаження-розвантаження, кругової інтерполяції і т.д. Комбінування цих бібліотечних програм дозволяє швидко програмувати ро-

бот для виконання більш складних задач і заощаджувати час, необхідний для налагодження програм.

*Термінологія*

АБСОЛЮТНА ТОЧКА – положення руки робота, представлене в абсолютних кутових значеннях зчленувань.

АДРЕСА РЯДКА (МІТКА) – записане перед інструкцією програми ціле число, на яке може бути зроблене посилання в інструкції розгалуження програми.

АРГУМЕНТ – дані, що уточнюють значення чи команди інструкції.

ВХІД/ВИХІД – вхідні і вихідні лінії для зв'язку з зовнішніми пристроями.

ДИРЕКТИВА – інструкція рівня монітора системи ARPS.

ІМПУЛЬСНИЙ ДАТЧИК – фотоелектричний імпульсний датчик для визначення величини кутів руху зчленувань.

ІНТЕРПОЛЬОВАНИЙ РУХ ЗЧЛЕНУВАНЬ – нелінійна траєкторія руху, при якій усі зчленування робота роблять мінімальний рух, необхідний для досягнення заданої точки.

КОМАНДА – інструкція рівня редактора системи ARPS.

КОМБІНОВАНА ТОЧКА – точка, обумовлена щодо іншої точки (чи іншим точкам).

КОНСТАНТА – число, величина яка незмінна.

КОНФІГУРАЦІЯ – визначає кути шарнірів 2, 3, 5, при яких робот переміщається в задану точку.

КООРДИНАТНА ТОЧКА – точка, значення якої визначається місцем розташування інструмента (x, y, z) і його орієнтацією (o, a, t).

МОНІТОР – верхній рівень керування системи ARPS.

ПОЧАТОК ОСЕЙ СИСТЕМИ КООРДИНАТ – точка перетинання

осей X, У, Z.

НОМЕР РЯДКА – номера на початку рядків робочої програми, що збільшуються від рядка до рядка.

ОРІЄНТАЦІЯ – визначає орієнтацію інструмента в заданій точці.

ОСНОВНА СИСТЕМА КООРДИНАТ – прямокутна система координат, початок відліку координат якої розташовано в центрі основи робота.

ПЕРЕМІННА – число, величину якого можна змінити програмно.

ПРОГРАМНИЙ ПЕРЕМИКАЧ – внутрішня перемінна операційної системи, що може приймати дві величини: активне (enabled) і пасивне (disabled).

ПРИКЛАДНА ПРОГРАМА – записана користувачем програма, що задає рухи й операції, виконувані роботом.

РЕДАКТОР – програмний рівень системи ARPS для запису програм.

СИНТАКСИЧНЕ ПРАВИЛО – правило запису команд.

СИСТЕМА КООРДИНАТ ІНСТРУМЕНТА – прямокутна система координат, початок відліку координат якої розташовано у фланці кисті для установки інструмента.

ТОЧКА – позиція в межах робочої зони робота.

ФАЙЛ – записана на гнучкому магнітному диску сукупність точок і/чи програм, до якої можна звертатися, указавши найменування даного файлу.

ФРЕЙМ – задана користувачем робоча площина робота, що проходить через три точки.

ФОРМАТ КОМАНД І ДИРЕКТИВ – звичайний формат команд і директив має вид:

КОМАНДА аргумент 1, аргумент 2, ...

де КОМАНДА – послідовність символів, що ілюструє операцію, яку варто виконати. Інструкції можуть складатися як з однієї частини (напри-

клад, BASE), так і з двох частин (наприклад, ZERO MEMORY), і дуже часто можуть бути скорочені до одного, двох чи трьох символів. (Наприклад, BASE може бути скорочене до B, а ZERO MEMORY – до ZM.) Однак, інструкцію не можна скорочувати до такого ступеня, щоб її однозначність утратилася. В методичних вказівках всі інструкції написані в нескороченій формі заголовними буквами. При програмуванні, однак, рекомендується використовувати скорочену форму;

аргумент 1, аргумент 2, ... – змінні, точки, величини кутів і т.д., зв'язані з командою. Вид і кількість аргументів коливаються в залежності від конкретної команди. Всі аргументи написані прописними буквами. Наприклад, у команді:

DELAY час

де час – єдиний аргумент команди DELAY, якому користувач повинний додати необхідне значення. Наприклад:

DELAY 5

що приведе до затримки в 5 секунд при виконанні програми.

Якщо тип аргументу укладений у кутові дужки (наприклад, < час >), то аргумент є необов'язковим і користувач не повинний обов'язково додавати йому значення. У цьому випадку система керування використовує задане внутрішнє значення за замовчуванням, що майже завжди дорівнює нулю. Наприклад:

BASE < dx >, < dy >, < dz >, < z-rotation >

У команді всі аргументи є необов'язковими і тому команда, що визначає перехід основної системи координат на 100 мм у напрямку по осі Z, може бути записана в таким чином:

BASE , , 100 або BASE 0, 0, 100, 0.

*Визначення точок*

Наступна операція – це навчити робот необхідним у програмі точкам. Визначення точок може бути виконано різними способами. Заради простоти на цьому етапі використовується тільки команда HERE.

Спочатку навчить робот точці, у якій виконується схват деталі. За допомогою пульта ручного керування переміщають маніпулятор у положення, у якому можна виконувати схват деталі. Після цього в ЗП робота записуються координати даного положення таким чином:

> HERE PART (сг)

Відповідним способом навчають точці BOX:

> HERE BOX (сг)

Тепер програма може бути ініційована і перевірена її функціонування.

*Виконання програми мовою ARPS*

Перша задача перед запуском робота – це установити придатну для даної програми величину швидкості переміщення. Установка швидкості здійснюється командою SPEED. Наприклад, швидкість 100 мм/с встановлюється таким чином:

> SPEED 100 (сг)

Примітка. На швидкість переміщення маніпулятора впливають також установка максимальної швидкості (максимальна припустима швидкість (MAXSPEED) і коефіцієнт швидкості (SPEED %). Максимальна швидкість, наприклад 500 мм/с, установлюється командою:

> MAXSPEED 500 (сг)

а коефіцієнт швидкості, наприклад 100 відсотків, командою:

> SPEED % 100 (сг)

Установлені значення швидкості можна вивести директивою LIST STATUS.

Коли встановлено необхідне значення швидкості, програма може бути запущена. Перед цією операцією, однак, варто розтиснути затискачі натисканням кнопки OPEN на пульті ручного керування, оскільки при складанні програми передбачалося, що спочатку затискачі розціплені.

Якщо маніпулятор не переміщається так, як треба, то натиснути кнопку останова на торці пульта ручного керування або кнопку OFF. Живлення привода можна також виключити натискаючи кнопку “ПИТАНИЕ ПРИВОДА 0” або кнопку аварійного останова.

Програма знову запускається командою RUN

> RUN DEM0.1 (cr)

RUN >

Робот виконає задану програму до кінця, переміщаючись через записані в ЗП точки, і потім зупиниться. Після цього на екрані дисплея виводиться повідомлення:

RUN > Exit

Stopped at STEP 9

Та ж програма може бути виконана 5 разів послідовно командою:

> RUN DEM0.1, 5 (cr)

Програма може бути зупинена трьома різними способами:

а) Директивою ABORT. Дана директива дозволяє виконати поточну команду до кінця, після чого робот зупиняється. Потім на екрані виводиться номер наступного кроку програми.

RUN > ABORT (cr)

Aborted

Stopped at STEP 6

б) Натискання кнопки на торці пульта ручного керування або кнопку OFF негайно зупинить виконання програми.

в) Шляхом натискання кнопки “Питание привода 0” виконання про-

грами негайно припиняється.

Студент повинен чітко розуміти, як записати програми робота за допомогою редактора, як навчити робот точкам, а також як можна запустити і зупинити складені програми.

Якщо оператор хоче більш докладно ознайомитися з виконанням основних операцій з роботом, таких, як наприклад програмування, використання редактора і т.д., це доцільно робити за допомогою команди “ARPS PRIMER”.

***Моніторні директиви***

*Загальні відомості*

Коли система керування ARPS ініціалізірована, відбувається перехід у режим керування МОНІТОР. Задача монітора – одержавши директиви, уведені з клавіатури оператором, виконати операції, зазначені в цих директивах. За допомогою моніторних директив можна запускати і зупиняти виконання прикладних програм, записувати і зберігати програми на гнучкому диску, навчати робот визначеним точкам і т.д.

Коли викликано режим монітора, то на екрані дисплея виводиться:

>

чи RUN >

Символ “ > ” указує, що жодна з програм робота не ініціалізірована (маніпулятор не рухається). Текст “ RUN >” означає, що програма, що знаходиться в ЗП блоку керування, запущена.

Звичайна форма моніторних директив наступна:

INSTRUCTION argument 1,  argument 2...

де INSTRUCTION – ім'я операції;

argument 1, argument 2... – перемінні, точки, і т.д., зв'язані з директивою. Якщо аргумент умовний, то він зображується у форматі команди в

кутових дужках (<argument>).

*Групи моніторних команд*

У цьому пункті коротенько описані окремі групи моніторних директив. Більш докладний опис операцій, виконуваних за допомогою моніторних директив.

*Директиви для визначення точок*

Точки в системі керування ARPS можна визначити за допомогою наступних директив:

CHANGE (точка)

Директива виводить значення точки робота з ЗП блоку керування на дисплей, після чого оператор із клавіатури може ввести нове значення точки.

HERE (точка)

За допомогою цієї директиви запишеться в ЗП поточна інформація про позицію маніпулятора.

WHERE <#>

Директива безупинно виводить на дисплей інформацію про позицію маніпулятора.

LTEACH (точка)

Ця директива встановлює режим “навчання”, коли шляхом натискання кнопки STEP (крок) на пульті ручного керування запишеться поточна позиція маніпулятора.

*Директиви для роботи з магнітними дисками*

STORE файл < = програма >, < програма > ...

Організує запис програм користувача і навчених точок на гнучкий диск.

LOAD (файл)

Директива забезпечує завантаження в ЗП блоку керування програм і точок, які записані на гнучкому магнітному диску.

FPACK

Директива стискає файли на гнучкому диску так, що між ними не буде невикористаних зон пам'яті.

*Директиви одержання листингов*

PLIST < програма >. . . .

Здійснює листинг програм на дисплей.

LLIST < точка >

Виводить перелік бажаних точок і їхніх величин на дисплей.

LIST COMMANDS

Директива виводить на дисплей список усіх моніторних директив, команд редактора і програм.

LIST STATUS

Директива виводить на дисплей дані про стан блоку керування.

*Директиви виконання програми*

RUN програма, < число прогонів >

Ця директива ініціалізірує бажану програму з ЗП блоку керування.

CONTINUE < постійна >

Ця директива продовжує виконання програми з перерваного кроку або з початку наступного кроку.

ABORT

Ця директива перериває виконання програми.

EXIT

Директива зупинить виконання програми наприкінці поточного циклу програми.

DIA (файл)

Ця директива виконує завантаження і запуск сервісної програми з диска.

COM (програма)

Ініціалізірує програму директив, що містить тільки моніторні директиви (порівнянні з командою COM)

*Директиви видалення*

LDEL < точка >

Вилучає бажані точки з ЗП блоку керування.

FDEL (файл)

Вилучає файл із гнучкого диска.

DLOAD (файл)

Вилучає з ЗП блоку керування ті програми або точки, які можна знайти у файлі.

ZERO DISK

Стирає інформацію, записану на гнучкому диску.

ZERO MEMORY

Стирає в ЗП блоку керування всі програми і точки маніпулятора, а також встановлює у вихідний стан усі внутрішні перемінні блоку керування.

*Директиви роздруківки довідника*

FDIR

Директива виводить на дисплей найменування файлів, що містяться на гнучкому диску.

PDIR

Виводить на дисплей найменування програм, що містяться в ЗП блоку керування.

*Директиви завдання швидкості руху "руки" робота*

MAX SPEED (швидкість)

Команда задає максимальну швидкість руху робота.

SPEED (швидкість)

Директива задає швидкість руху маніпулятора у виді абсолютного значення (мм/сек).

SPEED % (коефіцієнт швидкості)

Визначає масштабний коефіцієнт, на який треба помножити абсолютну швидкість, щоб одержати дійсну швидкість руху маніпулятора.

*Спеціальні директиви*

CAL < + / – > (індекс зчленування)

Калибрує маніпулятор.

DISABLE (перемикач програми)

Установлює перемикач програми в пасивний стан (вимкнено).

ENABLE (перемикач програми)

Установлює перемикач програми в активний стан (ввімкнено).

CLOAD (файл)

Завантажує з диска нові коефіцієнти калібрування.

SYSMON

Викликає перевід системи в режим монітора блоку керування (восьмеричне налагодження).

ZERO вихід 1 TO вихід 2

Установлює лінії виходу 1...2 у нульовий стан (обнуляє).

C (текст)

Виводить рядок для коментарів.

EDIT < програма >

Ініціалізирує редагування програми.

WEAVE амплітуда, < час циклу >, < час затримки >

Задає параметри (дивись команду GOS&WEAVE)

*Виконання команд на рівні монітора*

У режимі монітора можна виконувати також окремі команди, якщо поставити символ “.” перед командою. Наприклад,

>. OPEN (команда розтискання схвата маніпулятора).

Перевага використання команд із точкою полягає в тому, що якщо треба виконати окремі команди, то немає необхідності редагувати для даної команди власні програми користувача.

Примітка. При використанні команд із точками передбачається, що в даний момент програма користувача не запущена (тобто монітор виводить на екран дисплея символ “ > ”, а не RUN >). Пульт ручного керування повинний бути в режимі COMP.

*Докладний опис моніторних директив*

У цьому пункті приводиться опис окремих моніторних директив із прикладами. Крім того, указується рівень, на якому можна виконувати кожну директиву. Символ “ > ” означає, що дану директиву можна виконати, якщо до цього не була ініціалізірована жодна з програм користувача. Символ “RUN >” означає, що дану директиву можна виконати, якщо в цей момент виконується одна з програм користувача. Більшість моніторних дире-

ктив працюють на обох рівнях.

ABORT (RUN >)

Ця директива перериває виконання програми наприкінці поточного кроку. Формат директиви:

ABORT

C (>. RUN >)

Директива C (COMMENT) виводить рядок для коментарів у режимі монітора. Формат директиви:

C (текст)

де ТЕКСТ – довільна послідовність символів. Коментарі на рівні монітора можуть допомогти при створенні програми для режиму автоматичного запуску AUTOSTART (порівняно з командою COM). Наприклад:

```
C
C++++++++++++++++++++++++++++++
C PROGRAM NAME: PRO1
C++++++++++++++++++++++++++++++
C
```

CAL ( > )

За допомогою команди CAL (CALIBRATE) можна прокалібрувати пристрій кодування двигуна, що ведуть маніпулятор. Інакше кажучи, пристрій керування обчислює точні кути зчленувань маніпулятора. Формат директиви:

CAL < (номер зчленування)

де номер зчленування = 1...6 і визначає зчленування, у якому компенсація можлива.

Якщо номер зчленування зі знаком плюс, то положення його буде компенсовано в позитивному напрямку. Якщо номер зчленування зі знаком мінус, то компенсація положення зчленування йде в протилежному напрямку. Величина компенсації відповідає одному оберту двигуна даного зчленування, причому число їх не обмежене.

Примітка Успішність калібрування можна перевірити за допомогою команди “GO READY”.

Приклади:

а) CAL – виконує калібрування.

б) CAL 1, -5 – виконує калібрування і додає до положення зчленування 1 один оберт, що компенсує, двигуна зчленування, а для зчленування 5 виконує один оберт, що компенсує, у протилежному напрямку.

CHANGE (>. RUN >)

За допомогою цієї директиви можна змінити значення точок і/або записати нові точки. Формат директиви:

CHANGE (точка)

де точка – координатна точка, комбінована точка або абсолютна точка.

Після директиви CHANGE на дисплей виводиться значення позиції і з'являється знак питання:

CHANGE location? (змінити точку?)

Після чого користувач може задати з клавіатури нові значення елементам точок, розділені комами. Якщо значення яких-небудь елементів не задано, а на клавіатурі набираються лише розділові символи (коми), то ці значення залишаються незмінними. Операція CHANGE (заміна) закінчується тим, що на дисплей виводиться нове значення (чи старе, якщо елементи не були змінені) і натискається клавіша “CR” (“повернення каретки”). Якщо була задана нова точка, то вона буде записана в ЗП з її нульовими

елементами значення (за замовчуванням). Якщо точка – комбінована точка, то лише внутрішня точка може бути нової (наприклад X(A), а точка X повинна бути задана раніш. Приклади:

а) Точка A1 піддається зміні таким чином:

> CHANGE A1 (cr)

| | x | y | z | o | a | t |
|---|---|---|---|---|---|---|
| Al | 10.00 | 5.00 | 0.00 | 5.000 | 0.000 | 0.000 |

CHANGE location?: 5, , 5, 0 (cr)

| | x | y | z | o | a | t |
|---|---|---|---|---|---|---|
| A1 | 5.00 | 5.00 | 5.00 | 0.000 | 0.000 | 0.000 |

б) Абсолютна точка B1 піддається зміні таким чином:

CHANGE #B1 (cr)

| сочл. 1 | сочл. 2 | сочл. 3 | сочл. 4 | сочл. 5 | сочл. 6 |
|---|---|---|---|---|---|
| 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Change location?:, , , 10 (cr)

| | сочл. 1 | сочл. 2 | сочл. 3 | сочл. 4 | сочл. 5 | сочл. 6 |
|---|---|---|---|---|---|---|
| #B1 | 0.000 | 0.000 | 0.000 | 10.000 | 0.000 | 0.000 |

CLOAD ( > )

За допомогою директиви CLOAD (CALIBRATION LOAD) можна завантажити нові калібровані дані з диска для енкодера двигунів маніпулятора. Калібровані коефіцієнти застосовуються для визначення позиції зчленувань під час калібрування (порівняйте з моніторною директивою CAL). Формат директиви:

CLOAD (файл)

де файл – ім'я файлу на диску, що містить нові калібровані дані.

Файл зберігається на диску при виконанні сервісної програми POTCAL.

Примітка. Перед початком завантаження живлення приводів робота автоматично відключається.

Вихідні калібровані дані постійно зберігаються в ППЗП блоку керування, звідки вони автоматично передаються в ОЗП. Так як різні калібровані коефіцієнти енкодеров трохи відрізняються друг від друга, то кожний з них треба визначати окремо і зберігати в ППЗП. Якщо довелося замінити енкодер визначеного зчленування, то калібровані коефіцієнти треба знову визначити за допомогою сервісної програми POTCAL.

COM ( > )

Директива COM (COMMANDS) ініцізірує програму, якщо містить моніторні директиви (при редагуванні програми COM пишуть перед кожною командою, наприклад, COM CALL. Формат директиви:

COM програма

де програма – програма, що містить моніторні директиви.

При роботі з режимом COM моніторні директиви читаються з даної програми і виконуються оператором набрав їх із клавіатури. Переривати роботу COM можна натисканням будь-якої клавіші.

Приклад:

COM PROG – ініціалізірує програму команд PROG.

CONTINUE ( > )

За допомогою цієї директиви можна знову запустити зупинену програму. Формат директиви:

CONTINUE < ціле число >

де ціле число – 32768 ... 32767. Якщо число не задане або воно позитивне, то робота програми продовжується з наступного кроку. Якщо ж число негативне (рекомендується – 1), то програма продовжується з перерва-

ного кроку.

Якщо програма була перервана під час руху маніпулятора, то директива CONTINUE запускає програму руху робота знову з перерваної команди руху.

Іноді виникають ситуації, коли продовжити прогін програми за допомогою директиви CONTINUE неможливо. Тоді на дисплеї після директиви CONTINUE виводиться:

Try RUN command (спробуйте команду “Прогін”),

тоб то оператор повинен за допомогою команди RUN запускати програму із самого початку.

Приклади:

а) > CONTINUE – продовжує виконання перерваної програми.

б) > CONTINUE – 1 – продовжує виконання програми знову з перерваного кроку.

DLOAD ( >. RUN > )

Директива DLOAD (DELETE-LOAD) видаляє програми з ЗП блоку керування і/або точки на підставі назви файлу. Формат директиви:

DLOAD файл

де файл – найменування файлу, у масиві якого здійснюється видалення.

За допомогою даної директиви відбувається зчитування файлу програм з диска (файл .P) чи файлу точок (файл .L) чи обох файлів (при відсутності специфікатора .L або .P) і видаляють ті програми і/або точки з ЗП блоку керування, які можна знайти у файлах.

Примітка. Операцію можна переривати натисканням клавіші RETURN чи CONTROL-S (клавіші CONTROL і S треба натиснути одночасно).

Приклади:

а) DLOAD PROG.P – видаляє програми, перераховані у файлі PROG.P

б) DLOAD PROG – стирає програми, перераховані у файлі PROG.P і точки, перераховані у файлі PROG.L.

DIA

За допомогою директива DIA (DIAGNOSTIC) відбувається завантаження потрібної сервісної програми з гнучкого диска і її ініціалізація. Формат директиви:

DIA файл

де файл – найменування сервісної програми, що знаходиться на диску.

На початку виконання директиви перевіряти, чи ввімкнено живлення маніпулятора, і якщо так, то на дисплеї з'являється наступне повідомлення:

Warning: ARM POWER ON (Увага! Живлення маніпулятора ввімкнено)

Примітка. Після виконання будь-якої сервісної програми, коли робот знову запускається, оператор повинен обов'язково стирати ОЗП (режим ZERO MEMORY).

Приклади:

а) > DIA POT10SE – ініціалізірується сервісна програма POT10SE.

DISABLE ( >. RUN > )

За допомогою даної директиви відбувається обнуління перемикачів програми. Формат директиви:

DISABLE program switch  (перемикач програми вимкнено)

де program switch – дивись “Перемикачі програми”

Приклад:

а) DISABLE INCALLS – заборона на всі переривання входів/виходів.

ENABLE ( >. RUN > )

Цією директивою перемикачі програми переводяться в активний стан

(вимикаються). Формат директиви:

ENABLE program switch

де program switch – дивись “Перемикачі програми”

Наприклад:

а) > ENABLE INCALLS – дозволяє зовнішні переривання входів/виходів.

EDIT ( >. RUN > )

Ця директива ініціалізірує редактор. Формат директиви:

EDIT < програма >

де програма – найменування програми, яку треба редагувати. Якщо найменування програми не зазначено, то за замовчуванням відбувається редагування останньої програми, з якою працювали в режимі редактора.

Приклад:

а) EDIT PR61 – ініціалізірує редагування програми PR61.

EXIT (RUN >)

Ця директива зупиняє виконання програми при закінченні робочого циклу (число прогонів програми, задане командою RUN, обнуляется). Формат директиви:

EXIT

FDEL ( >. RUN > )

За допомогою директиви FDEL (FILE-DELETE) відбувається стирання файлів із гнучкого диска. Формат директиви:

FDEL файл

де файл – ім'я файлу, який треба стирати.

Приклади:

а) FDEL PROG.P – стирає файл PROG.P

б) FDEL PROG.L – стирає файл PROG.L

в) FDEL POT10 – стирає сервісну програму POT10

FDIR ( >. RUN > )

За допомогою директиви FDIR (FILE-DIRECTORY) виводиться список найменувань файлів, що знаходяться на гнучкому диску, їхня довжина в блоках (блок = 256 символів) і обсяг вільних блоків на гнучкому диску. Формат директиви:

FDIR

FPACK ( >. RUN > )

Завдяки директиві FPACK (FILE-PACK) вільні зони, що є на гнучкому диску, підготовляються для використання за рахунок стиску файлів. Вільні місця на гнучкому диску виникають при стиранні файлів з диска за допомогою директиви FDEL. Формат директиви:

FPACK

HERE ( >. RUN > )

За допомогою HERE встановлюється в ЗП величина точки, яка дорівнює поточній позиції маніпулятора. Формат директиви:

HERE точка

де точка – координатна точка, комбінована точка або абсолютна точка.

Примітка. Якщо використовується комбінована точка, то невизначеної може бути тільки внутрішня точка (наприклад X(A), а точка X повинна бути попередньо задана).

Наприклад:

а) > HERE A1 – значення точки A1 дорівнює значенню поточної пози-

ції маніпулятора (значення x, y, z, o, a, t).

б) > HERE #B1 – значення абсолютної точки #B1 дорівнює кутовому значенню шарнірів маніпулятора.

в) > HEPE PALLET (A) – значення точки A1 дорівнює різниці значень між точкою PALLET і поточною позицією маніпулятора.

LDEL ( >. RUN > )

Директива LDEL (LOCATION-DELETE) стирає точки з ЗП робота. Формат директиви:

LDEL точка, < точка > ...

де точка – координатна точка або абсолютна точка.

Приклади:

а) > LDEL A1, A2 – стирає в ЗП точки A1 і A2.

б) > LDEL # B1 – стирає в ЗП абсолютну точку # B1.

LIST COMMANDS ( >. RUN > )

Ця директива виводить на дисплей у скороченій формі моніторні директиви, команди, команди редактора, а також типи аргументів команд. Формат директиви:

LIST COMMANDS

Символи типів аргументів команд, що виводяться на екран дисплея, мають наступне значення:

ang – кутова величина;

аорг – арифметичний оператор;

стр – оператор порівняння;

dis – відстань;

file – найменування файлу;

jnt – індекс зчленування (шарніра);

1b1 – мітка, адреса рядка;

1oc – точка;

nn.n – десяткове число, –3276,8...+3276,7;

n.nn – десяткове число, –327,68...+327,67;

nnn – ціле число, –32768...+32767;

prg – найменування програми;

str – черга символів;

swit – перемикач програми;

var – перемінна;

... – повторення аргументу без обмежень.

Якщо аргумент викладений у кутових дужках (< >), то це означає, що він є умовною залежністю.

Наприклад:

Якщо директива BASE виводиться на екран дисплея у формі

BASE < dis >, < dis >, < dis >, < ang >

то три перших аргументи є відстанню (значення x, y, z лежать у межах – 1024 мм ... +1023,99 мм (маніпулятор PUMA), а четвертий аргумент – кутової (обертання осі Z між – 180,000 і + 179,995).

LIST STATUS ( >. RUN > )

Ця директива виводить на дисплей дані про внутрішній стан, такі як:

– зрушення основної системи координат (зрушення по осях x, y, z, поворот осі Z, дивись директиву BASE;

– обсяг вільної області пам'яті (ОЗП);

– стан перемикачів програми;

– дзеркальна система координат, якщо вона використовується (дивись команду MIRROR);

– масштабна система координат, якщо вона застосовується (дивись

команду SCALE);

– зрушення системи координат інструмента (дивись команди TOOL і LTOOL);

– завдання швидкості (максимальна швидкість переміщення маніпулятора, абсолютна швидкість, масштабний коефіцієнт швидкості);

– виконувані програма і кроки програми;

– виконано кількість прогонів програми і загальна кількість прогонів.

Формат директиви:

LIST STATUS

Приклад:

а) LIST STATUS

BASE 0.00 0.00 0.00 0.000 (основна система координат)

Unused mem. ( %) : 78,55 (невикористана пам'ять)

BREAK DISABLED (перемикач BREAK вимкнено)

DIST10 DISABLED (перемикач DIST10 вимкнено)

INCALIS DISABLED (перемикач INCALLS вимкнено)

PRINT ENABLED (перемикач PRINT ввімкнуто)

SERROR ENABLED (перемикач SERROR ввімкнуто)

TOOL: TL1 0.00 0.00 100.00 0.000 0.000 0.000

(зрушення системи координат інструмента)

Max speed (mm/s): 500.0 speed (mm/s): 100.0 speed % 100

(максимальна швидкість (мм/сек.): 500.0, швидкість (мм/сек..): 100.0, швидкість % 100)

Loops done: 1. of 1. (виконано циклів: 1. з 1.)

Next step: 5., program START (слід, крок: 5., програма START)

Next step: 9., program DEMO (наступний крок: 9, програма DEMO)

Примітка. У цьому прикладі програма START викликала підпрограму DEMO. Якщо на дисплеї з'явилося повідомлення про помилку типу

“Unused mem (%): RAM error” (Обсяг незайнятої пам'яті %: помилка ОЗП), то відбулося обнуління ЗП, тому що ОЗП втратило частину свого змісту або весь зміст цілком.

LLIST ( >. RUN > )

По директиві LLIST (LOCATION-LIST) на дисплей виводяться найменування точок і їхне значення. Формат директиви:

LLIST < точка >, ...

де точка – координатна чи абсолютна точка.

Якщо в директиві точки не задані, то усі значення точок, які є в ЗП, виводяться на дисплей. Якщо ж у директиві задана одна чи кілька точок, то на дисплей будуть виведені тільки вони.

Приклади:

а) >LLIST – на дисплей виводяться всі точки.

б) >LLIST A1, B1 – на дисплей виводяться значення точок A1 і B1.

LOAD ( >. RUN > )

За допомогою директиви LOAD у ЗП блоку керування з диска завантажуються команди, програми і/або точки. Формат директиви:

LOAD файл

де файл – найменування файлу, з якого завантажуються програми і/або точки. Якщо в найменуванні файлу є специфікатор “.P”, то завантажується тільки файл програм. Якщо є специфікатор “.L”, то завантажується тільки файл точок. Якщо ж у найменуванні файлу специфікатора нема, то будуть завантажені і файл програм і файл точок.

Примітка. Операція завантаження може бути перервана натисканням клавіші RETURN чи CONTROL і S (треба натискати одночасно).

Приклади:

а) LOAD FX – завантажить з диска файли FX.L і FX.P

б) LOAD FX.P – завантажить з диска файл FX.P

LTEACH ( >. RUN > )

За допомогою директиви LTEACH (LOCATION-TEACH) навчають робот точкам у режимі ручного керування. При кожному натисканні кнопки STEP (крок) на пульті ручного керування визначається одна точка і їй присвоюється значення, яке відповідає поточної позиції маніпулятора. Формат директиви:

LTEACH точка

де точка – координатна, комбінована чи абсолютна точка.

При кожному натисканні кнопки STEP на дисплей виводяться найменування точки і її значення. Крім найменування точки їй присвоюється й індекс, що для кожної наступної точки автоматично зростає на 1. Цей режим навчання закінчується натисканням кнопки CR.

Якщо в директиві як точка задається комбінована точка, то невизначеної може бути тільки внутрішня точка (наприклад X (A1), а значення точки X повинне бути визначене раніше.

Приклади:

а) > LTEACH A1 – визначає в ЗП робота послідовні точки A1, A2 і т.д.

б) >LTEACH #B1 – визначає послідовно точки # B1, # B2 и т.д.

в) >LTEACH A(X1) – визначає послідовні точки X1, X2 і т.д.

MAXSPEED ( >. RUN > )

Директива задає бажану межу швидкостей руху маніпулятора. Формат директиви:

MAXSPEED швидкість

де швидкість – швидкість мм/сек у межах 3000.0 мм/сек...2 мм/сек (прямолінійний рух). За замовчуванням максимальна швидкість 500 мм/сек.

Приклади:

а) > MAXSPEED 300 – задає максимальну швидкість, межа швидкості 300 мм/сек.

PDIR ( >. RUN > )

Директива PDIR (PROGRAM-DIRECTORY) виводить на дисплей найменування програм, що знаходяться в ЗП. Формат директиви:

PDIR < програма>, …

PLIST ( >. RUN > )

За допомогою директиви PLIST (PROGRAM-LIST) виводяться програми на дисплей. Формат директиви:

PLIST < програма >

де програма – найменування виведеної на дисплей програми. Якщо в директиві найменування програми не задано, то на дисплей виводяться всі програми, зберігаються в ЗП. Якщо ж у директиві найменування програм задані, то на дисплей виводяться тільки ці програми.

Приклади:

а) > PLIST – висвітиться на дисплей усіх програм із ЗП.

б) > PLIST PRG1, PRG2 – на дисплей виводяться програми PRG1, PRG2.

RUN

Ця директива запускає програми. Формат директиви:

RUN програма, < кількість прогонів >

де програма – найменування програми, яку треба виконати;

кількість прогонів – кількість прогонів програми (– 32768 ... + 32767). Від'ємне значення кількості циклів виконання програми приводить до беззупинного циклу, який можна перервати командами ABORT чи EXIT. Робочий цикл переривається натисканням будь-якої кнопки або якщо відбувається якийсь збій. Значення за замовчуванням циклів – 1.

При запуску програми директивою RUN ручне керування автоматично переходить у режим COMP.

Приклади:

а) > RUN PRG – виконає програму один раз.

б) RUN PRG, 10 – виконає програму PRG 10 разів.

в) RUN PRG, –1 – виконує програму PRG необмежену кількість.

SPEED ( >. RUN > )

Цією директивою задається базова швидкість рухів інструмента маніпулятора. Формат директиви:

SPEED швидкість

де швидкість – швидкість у мм/сек (рух по прямій), що вибирається в межах 3000 ... 2 мм/сек.

Примітка. Щоб одержати дійсну швидкість, треба помножити базову швидкість на масштабний коефіцієнт швидкості.

Приклад:

а) SPEED 300 – задає базову швидкість 300 мм/сек.

SPEED % ( >. RUN > )

Ця директива використовується для завдання масштабного коефіцієнта швидкості, за допомогою якого можна розрахувати абсолютну швидкість руху маніпулятора. Формат директиви:

SPEED % (масштабний коефіцієнт швидкості)

де масштабний коефіцієнт швидкості – цілочисельна перемінна чи константа, за якою масштабується абсолютна швидкість. Значення 100 ( = 100%) відповідає коефіцієнту 1.

Приклади:

а) SPEED 50 % – зменшує швидкість руху робота в 2 рази.

б) SPEED % X – зменшує швидкість руху робота на величину X

STORE ( >. RUN > )

За допомогою цієї директиви програми і/або точки записуються з ОЗП на магнітний диск. Формат директиви:

STORE файл < = програма >, ...

де файл – файл, у якому будуть зберігатися програми і/або точки. Якщо найменування файлу закінчується специфікатором .P, то зберігатися в ньому будуть тільки програми. Якщо найменування кінчається специфікатором .L, то зберігатися будуть тільки точки. Якщо ж немає специфікатора, то зберігатися будуть і програми і точки (автоматично формуються 2 файли, один із яких закінчується .P – файл програм, а інший .L – файл точок);

програма – найменування програми, яку треба записати. Якщо в директиві не вказується найменування програми, то записуються всі програми і/або точки, якщо ж у директиві зазначені найменування, то будуть записані тільки пойменовані програми і/або точки. Операцію можна перервати за допомогою клавіш RETURN чи CONTROL-S, причому клавіші CONTROL і S повинні бути натиснуті одночасно.

Приклади:

а) > STORE ALL – усі програми будуть записані у файл ALL.P, а всі точки у файл ALL.L.

б) > STORE ALL.L – усі точки будуть записані у файл ALL.L.

в) > STORE PRG. = PRG – усі програми будуть записані у файл PRG.P, а всі точки, використовувані в PRG, будуть записані у файл PRG.L.

г) > STORE LOC.L = PRG1, PRG2 – усі точки, записані в програмах PRG 1 і PRG 2, будуть записані у файл LOC.L.

SYSMON ( > )

Директивою SYSMON (SYSTEM-MONITOR) можна звертатися у власний монітор центрального процесора. Формат директиви:

SYSMON

Перед початком виконання директиви блок керування запитує:

Are you sure (Y. n)?

На це питання треба відповісти введенням із клавіатури символу “У” (так), інакше звертання до монітора системи не буде.

Приклад:

а) SYSMON

Are you sure? Y (Ви упевнені? Так)

WHERE ( >. RUN > )

Директива виводить на дисплей поточну позицію інструмента маніпулятора. Формат директиви:

WHERE <#>

Якщо директива задається із символом #, то позиція маніпулятора виводиться на дисплей у виді кутових значень шарнірів (режим абсолютної точки).

Виконання цього режиму можна перервати натисканням символу CR чи CONTROL-S. Операцію виводу можна тимчасово призупинити і продовжити знову натисканням будь-якої кнопки.

Приклади:

а) > WHERE

| x | y | z | o | a | t |
|---|---|---|---|---|---|
| 500.00 | 543.12 | 101.12 | 50.000 | –45.000 | 0.000 |

б) > WHERE #

| шарнір 1 | шарнір 2 | шарнір 3 | шарнір 4 | шарнір 5 | шарнір 6 |
|---|---|---|---|---|---|
| 0.000 | 90.000 | –90.000 | 0.000 | 0.000 | 0.000 |

WEAVE ( >. RUN > )

Директива задає параметри для коливання руху маніпулятора (порівняємо з командою COS & WEAVE). Формат директиви:

WEAVE амплітуда, < час коливання >, < затримка >

де амплітуда – подвоєна амплітуда (коливання від піка до піка, максимальне значення її 256 мм). Якщо амплітуда дорівнює 0, то коливання не буде;

час коливання – час одного циклу коливання (цей рух подвоєних амплітуд), він може бути в діапазоні 0.00 – 327.67 сек;

затримка – час останова (затримки руху робота в крайніх точках коливання), його величина лежить у межах 0.00 – 327.67 сек.

Приклади:

а) > WEAVE 5, 0.5 – встановлює амплітуду коливання маніпулятора, рівну 5 мм, а час коливання, рівне 0.5 сек.

б) WEAVE 0 – скасовує коливання маніпулятора.

ZERO DISK ( >. RUN > )

Ця директива видаляє зміст гнучкого диска. Формат директиви:

ZERO DISK

Перед початком виконання цієї команди на екрані дисплея з'являється питання:

Are you sure (y,n)? (Ви упевнені (так, ні)?,

на який треба відповісти натисканням клавіші “У” (“так”), щоб видалити зміст диска.

Примітка. Нагадаємо, що новий диск перед його використанням треба обов'язково стерти.

Приклад:

a) > ZERO DISK

Are you sure ( y, n )?:  Y

ZERO MEMORY ( > )

Ця директива очистить пам'ять користувача. Формат директиви:

ZERO MEMORY

Перед початком виконання директиви на екрані з'являється питання:

Are you sure ( y, n )?  (Ви упевнені? Так, ні?),

на який треба відповісти натисканням клавіші “У” (“так”), щоб очистити пам'ять.

Приклад:

a) >ZERO MEMORY

Are you sure (y,n)? : Y

***Програмні команди***

*Загальні відомості*

Програма робота – група команд, що знаходиться в ЗП блоку керування, що керує рухами робота і контролює їх. Програмі, з будь-якої кількості символів, можна дати найменування. В якості цих символів можуть бути літери A...Z, числа 0...9 і точка (.).  Програми записуються в ЗП блоку

керування за допомогою команд редактора, після чого можна ініціалізірувати їх виконання за допомогою моніторної директиви RUN.

*Групи програмних команд*

У цьому пункті розглядаються окремі групи програмних команд, а більш докладно їхнє значення і виконання ними роботи розглядаються нижче.

*Команди для визначення точок*

HERE точка – записує в значення поточну позицію.

FRAME точка 1 = точка 2, точка 3, точка 4, <точка 5> – формує площину, яка проходить через точки 2, 3 і 4.

LOCATE точка 1 = < INVERSE > точка 2 – обчислює значення точки 2 і встановлює результат у точку 1.

SHIFT точка = < dx >, < dy >, < dz > – зрушує місце розташування даних точок.

DISTANCE перемінна = точка 1, точка 2 – обчислює відстань між точками 1 і точками 2 у міліметрах.

*Команди розгалуження*

JUMP мітка – виконання розгалуження в рядку, що задається міткою.

IF < INGROUP > перемінна cmp < INGROUP > перемінна 2 THEN JUMP labe1 – програма розгалужується в рядок, заданий міткою, якщо справедлива умова між перемінної 1 і перемінної 2.

IF IN INPUT < вхід >, < вхід >, < вхід >, < вхід > then JUMP мітка – програма розгалужується в рядок, заданий міткою, якщо коректний стан вхідних проводів/шин.

CALL програма – програма переходить на підпрограму.

RETURN < перемінна > – програма повертається після виконання підпрограми.

*Команди керування маніпулятором*

GO точка – переміщає маніпулятор у задану точку по інтерпольованій траєкторії шарнірів.

GOS точка – переміщає маніпулятор прямолінійно в задану точку.

GO&OPEN точка – розтискає схват маніпулятора і переміщає маніпулятор у задану точку уздовж по інтерпольованій траєкторії шарнірів.

GO&CLOSE точка – стискає схват і переміщає його уздовж інтерпольованої траєкторії шарнірів у задану точку.

GOS&OPEN точка – розтискає схват і переміщає його в задану точку по прямолінійній траєкторії.

COS&CLOSE точка – стискає схват і переміщає його в задану точку по прямолінійній траєкторії.

GONEAR < точка >, відстань – переміщає маніпулятор на задану відстань від точки уздовж інтерпольованої траєкторії шарнірів.

GOSNEAR < точка >, відстань – переміщає маніпулятор на задану відстань від точки уздовж прямолінійної траєкторії.

GOS&WEAVE – точка переміщає маніпулятор у задану точку уздовж прямолінійної траєкторії, додатково додаючи маніпулятору коливання, що вводиться командою WEAVE.

MOVE < dx >, < dy >, < dz > – переміщає маніпулятор щодо основної системи координат уздовж інтерпольованої траєкторії руху шарнірів.

MOVES < dx >, < dy >, < dz > – переміщає маніпулятор щодо основної системи координат уздовж прямолінійної траєкторії.

TMOVE < dx >, < dy >, < dz > – переміщає маніпулятор щодо системи координат інструмента по інтерпольованій траєкторії шарнірів.

TMOVES < dx >, < dy >, < dz > – переміщає маніпулятор щодо системи координат інструмента по прямолінійній траєкторії.

MOVE JOINT шарнір, кут – переміщає окремий шарнір маніпулятора.

ALIGN – вирівнює вісь Z системи координат інструмента робота в напрямку найближчих осей X, *У, Z* основної системи координат.

GO READY – переміщає маніпулятор у спеціальне положення. Більш докладно щодо цього середнього положення – дивись вище.

*Команди установки швидкості руху маніпулятора*

SPEED швидкість – задає базову швидкість руху маніпулятора.

SPEED % масштабний коефіцієнт швидкості – установлює масштабний коефіцієнт швидкості руху маніпулятора.

SPEED NEXT швидкість – задає швидкість наступної команди переміщення маніпулятора.

*Команди керування схватом маніпулятора*

OPEN – розтискає схват маніпулятора

CLOSE – стискає схват маніпулятора

ODELAY час – установлює час перебування схвата маніпулятора відкритим.

CDELAY час – задає час перебування схвата маніпулятора закритим.

*Команди керування входами /виходами*

OUTGROUP група = < INGROUP > число 1 < арифметичний оператор > < INGROUP > число 2 – установлює 16-канальний порт виходів.

OUT вихід, < вихід >, < вихід >, < вихід > – керує окремими вихідними лініями.

RUNOUT вихід, < вихід >, < вихід >, < вихід >

IF IN вхід, < вхід >, < вхід >, < вхід > THEN JUMP мітка – тестує окремі вхідні лінії.

WAIT IN вхід, < вхід >, < вхід >, < вхід > – очікує потрібної комбінації вхідних ліній.

INCALL вхід, програма < NOBREAK > – дозволяє зовнішні переривання.

NO INCALL вхід – забороняє зовнішні переривання.

*Команди для зрушення системи координат інструмента*

TOOL < x >, < y >, < z >, < o >, < a >, < t > – задає зрушення системи координат інструмента.

LTOOL точка – указує точку, значення якої відповідає зрушенню системи координат інструмента.

*Команди керування конфігурацією маніпулятора*

J2 RIGHT – визначає праву конфігурацію руху маніпулятора.

J2 LEFT – визначає ліву конфігурацію руху маніпулятора.

J3 UP – визначає конфігурацію “лікоть нагору”.

J3 DOWN – визначає конфігурацію “лікоть униз”.

J5 PLUS – робить позитивним кутове значення шарніра 5.

J5 MINUS – робить від’ємним кутове значення шарніра 5.

*Команди останова*

STOP < текст >, < число > – перериває виконання програми.

HALT < текст >, < число > – перериває виконання програми, неможливо продовжити виконання програми директивою CONTINUE.

EXIT – зупиняє виконання програми наприкінці поточного числа.

*Команди керування гнучким диском*

LOAD файл < NOBREAK > – завантажує програми і/або точки робота з гнучкого диска в ОЗП.

DLOAD файл < NOBREAK > – видаляє програми і/або точки на основі назви файлу.

WAIT LOAD – очікує завершення операцій із гнучким диском.

*Спеціальні команди*

BASE < x >, < y >, < z >, < обертання навколо осі z > – зрушить розташування основної системи координат.

C текст – видає рядок для коментарів.

COM моніторна директива – додає до програми моніторну директиву.

DELAY час – установлює затримки в програмі.

DISABLE перемикач програми – установлює перемикачі програми в пасивний стан.

ENABLE перемикач програми – установлює перемикачі програми в активний стан.

PRINT < текст >, < число > – виводить текст і значення перемінної на дисплей.

SET перемінна = < INGROUP > число1 < арифметичний оператор > < INGROUP > < число2 > – обчислює значення арифметичної операції, у якій є дві перемінні.

TOL NARROW – задає максимальну точність руху робота.

TOL WIDE – зменшує точність руху маніпулятора.

MIRROR точка – активує дзеркальне відображення роботи робота щодо якої-небудь точки або площини.

NO MIRROR – забороняє роботу робота в “дзеркальному” режимі.

WEAVE амплітуда, < час коливання >, < затримка > – установлює па-

раметри коливання.

SCALE точка = < x >, < y >, < z > – активує режим масштабувань щодо якої-небудь точки.

NO SCALE – забороняє роботу в режимі масштабувань.

*Докладний опис програмних команд*

У цьому пункті даний більш докладний опис перерахованих вище програмних команд. Якщо маніпулятор зупиняється перед виконанням даної команди, то ця команда супроводжується позначкою BREAK (переривання).

ALIGN програмна команда

За допомогою цієї команди відбувається вирівнювання інструмента маніпулятора по осях основної системи координат. Блок керування робота обчислює, які з осей основної системи координат є найближчої до осі Z інструмента робота і переміщає маніпулятор у напрямку цієї осі. Після цього осі стають паралельними. Формат команди:

ALIGN

BASE програмна команда (BREAK)

Ця команда дозволяє зрушити основну систему координат робота. Формат команди:

BASE < x >, < y >, < z >, < ротація осі z >

де X – зрушення основної системи координат щодо вихідної основної системи координат у напрямку осі X – 1024 ... 1023.99 мм;

У – те ж, що і X, але в напрямку осі Y;

Z – те ж, що і X, але в напрямку осі Z;

ротація осі Z – ротація основної системи координат навколо осі Z, повинна бути в межах – 180.00 ... + 179.00. При позитивних значеннях рота-

ція відбувається за годинниковою стрілкою.

Кілька послідовних команд BASE не підсумуються, а зрушення завжди розраховуються щодо положення вихідної основної системи координат.

Приклади:

а) BASE 10, 20, ,45 – зрушить основну систему координат: на 10 мм по осі X, на 20 мм по осі B, на 0 мм по осі Z, на 45 градусів повертає навколо осі Z.

б) BASE – обнуляє режим “BASE”

C програмна команда

За допомогою цієї команди (COMMENT) до програми можна додати рядки коментарів, що не впливають на роботу програми. Формат команди:

C текст

де текст – довільна послідовність символів.

Приклад:

а) C STOP IF ERRORS

CALL програмна команда

Цією командою здійснюється перехід на підпрограму. Формат команди:

CALL програма

де програма – назва програми, який передається керування.

У програмі, перехід до якої здійснюється за допомогою команди CALL, останньою командою повинна бути команда RETURN.

Приклад:

а) CALL PRG1 – викликає перехід до підпрограми PRG1.

CDELAY програмна команда

Ця команда (CLOSE-DELAY) використовується для того, щоб задати затримку, що вимагається для стиску схвата маніпулятора. (Формат команди:

CDELAY час

де час – затримка, відпущена на стискання схвата маніпулятора (0.00...328.67 сек.). По замовченню ця затримка складає 0.2 сек. Команда CDELAY не впливає на роботу команд GO&CLOSE і GOS&CLOSE.

Приклад:

а) CDELAY 0,5 – задається затримка 0,5 сек. на стискання схвата маніпулятора.

COM програмна команда

За допомогою команди COM (MONITOR COMMAND) можна писати програми, що включають моніторні директиви (так звані програми команд). Формат команди:

COM моніторна директива

Програмні команди і моніторні директиви, записані за допомогою команди COM, не сполучити в складанні однієї і тієї ж програми.

При виведенні на дисплей програм команд за допомогою директиви PLIST виводяться команди COM із трьома послідовними точками, щоб відокремити їх від програмних команд. Наприклад, якщо в програму введена команда COM CAL, вона буде виведена у вигляді COM...CAL.

Приклади:

а) COM MAXSPEED 600 – установлює максимальну швидкість 600.

б) COM COM HELP – ініціалізірує програму команд HELP.

CLOSE програмна команда (BREAK)

Цією командою здійснюється стискання схвату маніпулятора. Формат команди:

CLOSE

За замовчуванням час затримки на закриття схвата складає 0,2 сек, але його можна змінити за допомогою команди CDELAY.

DELAY програмна команда (BREAK)

За допомогою цієї команди можна формувати затримки в програмі. Формат команди:

DELAY час

де час – час затримки (0,00...327,67 сек.)

Приклад:

а) DELAY 0,5 – у програмі створюється затримка 0,5 сек.

DISTANCE програмна команда

По цій команді здійснюється обчислення відстані між двома точками в системі координат. Формат команди:

DISTANCE перемінна = точка 1, точка 2

де перемінна – целочисельна перемінна, значення якої задає відстань між точкою 1 і точкою 2;

точка 1 – координатна точка;

точка 2 – координатна точка.

Точність обчислення складає або 1 мм (коли перемикач програми DIST10 виключений) або 0,1 мм (коли перемикач програми DIST10 включений).

Приклад:

а) DISTANCE X = LOC1, LOC2 – обчислюється відстань між точками

LOC1 і LOC2, величина встановлюється у виді перемінної X.

DISABLE програмна команда (BREAK)

Ця команда працює також, як і аналогічна моніторна директива (переводить внутрішні перемикачі програми в пасивна/виключений стан). Формат команди:

DISABLE перемикач програми

DLOAD програмна команда.

За допомогою цієї команди (DELETE-LOAD) відбувається видалення програм і / або точок з файлів, розміщених у ЗП. Робота команди аналогічна відповідній моніторній директиві. Формат команди:

DLOAD файл < NOBREAK >

де файл – назва файлу, у якому знаходиться програма або точка, яку треба видалити. Якщо в назві є специфікатор .P, то віддаляються всі програми. Якщо в назві є специфікатор .L, то стираються тільки точки. Якщо ж у назві немає специфікатора, то стираються і програми і точки.

NOBREAK – якщо заданий аргумент NOBREAK, то програми і точки стираються без зупинки руху робота. Якщо ж аргумент NOBREAK не заданий, то виконання програми роботи робота припиняється, поки не буде закінчене видалення.

Операцію видалення можна переривати натисканням клавіші “CONTROL-S” (клавіші CONTROL і S натискати одночасно).

УВАГА!

Варто бути дуже обережним, щоб не видалити з ЗП ті програми, що робот можливо виконує саме в цей час. Наслідком неправильних операцій видалення може бути останов програмного забезпечення робота. Так як під час видалення виконується зчитування з гнучкого диска, то чекання закін-

чення операції видалення можна здійснювати за допомогою програмної команди WAIT LOAD.

EXIT програмна команда

Ця команда працює аналогічно відповідній моніторній директиві, тобто припиняє виконання програми наприкінці поточного циклу. Формат команди:

EXIT

ENABLE програмна команда (BREAK)

Ця команда працює аналогічно відповідній моніторній директиві, тобто включає перемикачі програми. Її формат:

ENABLE перемикач програми

FRAME програмна команда

Формує робочі площини для робота. Формат команди:

FRAME точка 1 = точка 2, точка 3, точка 4, < точка 5 >

де точка 1 – назва формованої площини FRAME;

точка 2 – початок системи координат Y FRAME;

точка 3 – напрямок осі X FRAME щодо точки 2;

точка 4 – точка, через яку проходить площина, паралельна осі X FRAME;

точка 5 – визначення остаточного початку системи координат FRAME (значення x, y, z). Якщо позиція 5 не задана, то початок системи координат FRAME розташований в точці 2.

Приклади:

а) FRAME F = OR, XPOINT, YPOINT – утворить систему координат F, у якій площина XY проходить через точки OR, XPOINT, YPOINT. Початок

системи розташований у точці OR.

б) FRAME F = OR, XPOINT, YPOINT, CORNER – все аналогічно прикладу а), але початок системи переміщається в точку CORNER.

GO і GOS програмні команди

За допомогою цих команд можна перемістити маніпулятор у бажану точку. Траєкторія руху в точку – це інтерпольована траєкторія руху шарніра по команді GO і прямолінійна траєкторія руху по команді GOS. Формат команд:

GO точка

GOS точка

де точка – координатна, комбінована чи абсолютна точка.

Приклади:

а) GO LOC1 – маніпулятор переміщається в точку LOC1.

б) COS LOC1 – маніпулятор переміщається в точку LOC1 по прямолінійній траєкторії.

в) COS #B1 – маніпулятор переміщається в абсолютну точку #B1 по прямолінійній траєкторії.

г) GO LOC(X) – маніпулятор переміщається в комбіновану точку LOC(X)

GONEAR і GOSNEAR програмні команди

За допомогою цих команд можна пересунути маніпулятор на потрібну відстань до заданої точки (у напрямку осі Z системи координат інструмента робота). Рух по команді GONEAR відбувається по інтерпольованій траєкторії руху шарніра, а команді GOSNEAR він відбувається по прямолінійній траєкторії. Формат команд:

GONEAR < точка >, відстань

GOSNEAR < точка >, відстань

де точка – координатна, комбінована чи абсолютна точка. Якщо точка не задана, то по замовчуванням використовується поточна позиція інструмента робота;

відстань – відстань, на яку пересунеться маніпулятор від бажаної точки. Відстань виміряється в мм у напрямку осі Z системи координат інструмента робота.

Приклади:

а) GONEAR #B1, 100 – маніпулятор пересунеться до абсолютної точки #B1 і буде віддалений від неї на відстань 100 мм.

б) GOSNEAR LOC(X), 5 – маніпулятор пересунеться в напрямку до комбінованої точки LOG X по прямолінійній траєкторії і буде віддалений від неї на 5 мм.

в) GOSNEAR 100 – маніпулятор пересунеться на 100 мм від її поточного положення в напрямок осі Z системи координат інструмента робота.

GO&OPEN, GOS&OPEN і GO&CLOSE, GOS&CLOSE програмні команди

Ці команди забезпечують операції, аналогічні командам GO і GOS, але на початку руху маніпулятора схват чи розціплений: GO&OPEN, GOS&OPEN, чи стиснутий: GO&CLOSE, GOS&CLOSE. Формат команд:

GO&OPEN точка

GOS&OPEN точка

GO&CLOSE точка

GOS&CLOSE точка

де точка – координатна, комбінована чи абсолютна точка.

Ця команда не перериває контурний рух, і тому використання команд CDELAY і ODELAY не впливає на її виконання.

GO READY програмна команда

По цій команді маніпулятор переміщається у вертикальне положення (кута для шарнірів 1 = 0°, 2 = 90°, 3 = –90°, 4 = 0°, 5 = 0°, 6 = 0°). Формат команди:

GO READY

GOS&WEAVE програмна команда

По цій команді маніпулятор переміщається в потрібне положення по прямолінійній траєкторії. Якщо користувач за допомогою команди WEAVE задає параметри коливання, то режим коливання накладається на прямолінійний рух. Формат команди:

GOS&WEAVE точка

де точка – координатна, комбінована або абсолютна точка.

Приклад:

а) GOS&WEAVE A – маніпулятор переміщається в положення A с коливанням по прямолінійній траєкторії.

HERE програмна команда (BREAK)

Ця команда працює також, як і аналогічна моніторна директива (записує в СК координати поточної позиції маніпулятора). Формат команди:

HERE точка

де точка – координатна, комбінована або абсолютна точка.

HALT програмна команда (BREAK)

Ця команда зупиняє виконання програми і неможливо продовжити виконання програми за допомогою директиви CONTINUE. Формат команди:

HALT < текст >, < число >

де текст – довільна послідовність символів, що виводиться на дисплей при виконанні команди HALT

число – цілочисельна перемінна або константа, значення якої виводиться на дисплей після тексту при виконанні команди HALT.

Приклади:

а) HALT ERROR, ERRORCODE – перериває виконання програми і виводить на дисплей текст ERROR (помилка), а також значення цілочисленої перемінної ERRORCODE.

б) HALT ERROR – працює аналогічно пункту а), але на дисплей виводиться тільки слово “ERROR”.

в) HALT 5 – перериває виконання програми і виводить на дисплей номер 5.

г) HALT – перериває виконання програми.

IF програмна команда

По цій команді виконується порівняння двох чисел між собою і якщо порівняння істина, то відбувається розгалуження програми. Формат команди:

IF <INGROUP> число 1 оператор порівняння <INGROUP> число 2 THEN JUMP мітка

де число 1 (перемінна 1) – цілочисельна перемінна чи константа, значення якої є першою величиною порівняння. Якщо оператор INGROUP стоїть перед перемінної 1, то виконується зчитування відповідної групи з 16-канального порту входів, що використовується як перша величина порівняння;

2 (перемінна 2) – друга величина порівняння;

CMP – оператор порівняння;

< > – не дорівнюють;

= – дорівнює;

> – більше чим;

< – менше ніж;

>_ – більше або дорівнює;

<_ – менше або дорівнює;

мітка 1 – мітка адреси рядка програми, на яку передається керування, якщо порівняння перемінної 1 і перемінної 2 виявилося істиною.

Приклади:

а) IF A = B THEN JUMP 10 – перевіряються числа A и B, і якщо вони рівні, то керування передається на рядок з адресою 10.

б) IF INGROUP 1 > 0 THEN JUMP 10 – перевіряється, чи є значення групи входу 1 більше 0.

IF IN програмна команда

За допомогою цієї команди тестуються окремі вхідні лінії. Формат команди:

IF IN вхід, < вхід >, < вхід >, < вхід > THEN JUMP мітка

де вхід – цілочисельна константа чи перемінна, значення якої указує вхідну лінію, яку треба тестувати. Якщо величина позитивна, то тестується стан “1” (тобто струм йде у вхідній лінії). Якщо ж величина негативна, то тестується стан “0” (наприклад, струму в лінії немає). За допомогою цієї команди можна перевірити 1, 2, 3 чи 4 вихідні лінії;

мітка – адреса рядка, на яку передається керування, якщо всі задані в команді вхідні лінії знаходяться в стані, передбаченому умовою.

Приклади:

а) IF IN 1 THEN JUMP 10 – перевіряється, чи знаходиться вхідна лінія 1 у стані “1” і якщо так, то керування передається на рядок з адресою 10.

б) IF IN 3, –5 THEN JUMP 100 – здійснюється перевірка, чи знахо-

диться вхідна лінія 3 у стані “1” і вхідна лінія 5 у стані “0”, то програма вертається на рядок з адресою 100.

в) IF IN CHX THEN JUMP 10 – перевіряється вхідна лінія, стан якої визначається перемінної CHX. Якщо CHX позитивна, то перевіряється активний стан, якщо ж CHX негативна, то перевіряється пасивний стан.

INCALL програмна команда (BREAK)

Ця команда активує зовнішнє переривання. Формат команди:

INCALL вхід, програма < NOBREAK >

де вхід – цілочисельна перемінна чи константа, величина якої позначає вхідну лінію, по якій здійснюється переривання. Якщо ця величина позитивна, то переривання відбувається при стані “1” даної вхідної лінії (передній фронт). Якщо ж ця величина негативна, то переривання відбувається при стані “0” вхідну лінію (задній фронт). Переривання дозволяються вхідним лініям 1...16;

програма – назва підпрограми, на яку передається керування, коли відбувається переривання вхідної лінії. Ця підпрограма повинна закінчуватись командою RETURN;

NOBREAK – якщо в команду включений аргумент NOBREAK, то під час руху робота переривання не допускається, воно дозволяється тільки наприкінці кожного руху робота (тобто рух не переривається). Якщо ж оператор NOBREAK не заданий, то маніпулятор зупиняється відразу, коли відбувається переривання.

Якщо перемикач програми INCALL обнулений (виключений), то всі переривання вхідної лінії скасовуються. Іншими словами, прапор INCALLS треба задавати за допомогою команди ENABLE INCALLS до того, як можна застосувати переривання входів. Коли відбувається переривання якої-небудь лінії і розгалуження програми на підпрограму, переми-

кач програми INCALLS автоматично скидається, а нові переривання будуть заборонені.

Примітка. В одній і тій же вхідній лінії неможливо одночасно здійснювати переривання в стані лінії “1” (передній фронт) і в стані “0” (задній фронт). Команда INCALL автоматично забороняє всі попередні переривання в цій же вхідній лінії.

Приклади:

а) INCALL 4, INT 4 – здійснює переривання у вхідній лінії 4. Стан “1” у вхідній лінії 4 призводить до передачі керування на виконання програми INT 4.

б) INCALL – 9, INT9 NOBREAK – здійснює переривання у вхідній лінії 9. Якщо стан цієї лінії буде “0”, то після виконання поточної команди керування передається на виконання програми INT9.

в) INCALL INTWX, INT – реалізує переривання вхідної лінії, заданої перемінної INTWX. Якщо INTWX позитивна, то переривання відбувається при стані “1” вхідної лінії. Якщо ця перемінна негативна, то переривання відбувається при стані “0” вхідної лінії. Після переривання керування передається на виконання підпрограми INT.

JUMP програмна команда

При завданні цієї команди організується розгалуження програми на бажаний рядок. Формат команди:

JUMP мітка

де мітка – адреса рядка, на яку передається керування.

Приклад:

а) 100 DELAY 1

.....

JUMP 100

J2 RIGHT і J2 LEFT програмні команди

Ці команди використовуються для вибору конфігурації маніпулятора (“права рука”, “ліва рука”). Формат команди:

J2 RIGHT

J2 LEFT

Командою J2 RIGHT вибирається конфігурація “права рука” (тобто в наступну точку маніпулятор переміщається в такому положенні, при якому шарнір 2 знаходиться на правій стороні дивлячись позаду кожуха даного шарніра). Команду J2 LEFT використовують для вибирання конфігурації “ліва рука”.

J3 UP and J3 DOWN програмні команди

За допомогою цих команд створюється конфігурація шарніра 3 (ліктя) маніпулятора. Формат команд:

J3 UP

J3 DOWN

За допомогою команди J3 UP вибирається конфігурація “лікоть у верхньому положенні” (тобто в наступну точку робот пересувається так, що шарнір 3 буде над прямою лінією між шарнірами 2 ... 4). За допомогою команди J3 DOWN вибирається конфігурація “лікоть у нижньому положенні”.

JS PLUS and J5 MINUS програмні команди

За допомогою цих команд задається конфігурація шарніра. Формат команд:

J5 PLUS

J5 MINUS

Командою J5 PLUS створюється така конфігурація шарніра 5, при

якій кутова величина шарніра 5 буде позитивною. Командою J5 MINUS задається негативна кутова величина шарніра 5.

LOAD програмна команда

По цій команді здійснюється завантаження в ОЗП програм і/чи точок із гнучкого диска. Робота команди аналогічна такій же моніторній директиві. Формат команди:

LOAD файл < NOBREAK >

де файл – назва файлу, що містить програми і/чи точки, який треба завантажити в ЗП. Якщо в назву входить специфікатор .P, то завантажуються тільки програми. Якщо в назву входить специфікатор .L, то завантажуються тільки точки. Якщо ж специфікатор не заданий, то завантажуються і програми, і точки.

NOBREAK – при наявності аргументу завантаження програм і точок відбувається без зупинки руху маніпулятора. Якщо ж цей аргумент не заданий, то робот зупиняє виконання програми, поки не буде завершено завантаження.

Примітка. Завантаження можна здійснити за допомогою натискання клавіші CONTROL-C, причому клавіші CONTROL і C треба натискати одночасно.

Оператору треба бути дуже уважним, щоб не завантажувати в ЗП ті програми, що робот виконує саме в цей час. Набір операцій, що послідовно задаються, видалення може “зіпсувати” програмне забезпечення робота. Так як під час завантаження відбувається зчитування з гнучкого диска, то за допомогою команди WAIT LOAD можна організувати режим чекання закінчення завантаження.

Приклад:

а) LOAD NEWPRG NOBREAK – завантажує програму з файлу

NEWPRG.P і точки з файлу NEWPRG.L у ЗП робота. Під час завантаження робот продовжує виконувати програму.

LOCATE програмна команда

По цій команді значення однієї точки задається як значення другої точки. Формат команди:

LOCATE точка 1 = < INVERSE > точка 2

де точка 1 – координатна чи абсолютна точка, чиє значення задається як значення точки 2;

точка 2 – координатна, комбінована чи абсолютна точка;

INVERSE – якщо заданий оператор INVERSE (= інверсія), то обчислюється зворотна величина точки.

Примітка. Якщо точка 2 – абсолютна точка, то і точка 1 повинна бути абсолютною точкою.

Приклади:

а) LOCATE A1 = A2 – як величину точки A2 задається точка A1.

б) LOCATE #B1 = #B2 – як значення абсолютної точки #B2 береться значення абсолютної точки #B1.

в) LOCATE A1 = INVERSE X (Y) – як значення точки A1 використовується зворотне значення комбінованої точки X (У).

LTOOL програмні інструкції (BREAK)

По цій команді для зрушення системи координат інструмента маніпулятора задається величина точки. Формат команди:

LTOOL точка

де точка – координатна точка, значення якої задається як зрушення системи координат інструмента маніпулятора.

Приклад:

а) LTOOL TOOL1 – значення точки TOOL1 вибирається як величина зрушення системи координат інструмента маніпулятора.

MIRROR програмна команда

Команда реалізує обчислення дзеркального відображення руху робота відносно якої-небудь точки. Формат команди:

MIRROR точка

де точка – координатна точка, щодо якої створюється дзеркальне відображення.

Приклад:

а) MIRROR CORNER GO CORNER (XI) – переміщає маніпулятор у комбіновану точку, що утворена з точки CORNER (кут) і дзеркального відображення точки XI.

Примітки. У даний момент можна реалізувати тільки одне дзеркальне відображення. Якщо задається кілька послідовних команд MIRROR, то реалізується тільки остання. Команда RUN автоматично обнуляє режим дзеркального відображення. Рекомендується використовувати як точку для дзеркального відображення точки, сформовані командою FRAME.

MOVE і MOVES програмні команди

По цих командах маніпулятор переміщається на задану відстань щодо основної системи координат. Формат команд:

MOVE < dx >, < dy >, < dz >

MOVES < dx >, < dy >, < dz >

де dx – величина переміщення маніпулятора по осі X основної системи координат;

dy – аналогічно зазначеному вище, але по осі У;

dz – аналогічно зазначеному вище, але по осі Z.

По команді MOVE здійснюється інтерпольована траєкторія руху шарніра, по команді MOVES прямолінійна траєкторія.

Примітка. Рух у напрямку осей X, У, Z відбувається одночасно, а не по черзі, (спочатку по осі X, потім по осі У і потім по осі Z).

Приклади:

а) MOVE 10, 20 – маніпулятор пересувається на 10 мм по осі X і на 20 мм по осі Z.

б) MOVES – 100, – 100, – 100 – маніпулятор пересувається по прямій на – 100 мм по всіх осях основної системи координат.

MOVE JOINT програмна команда

За допомогою цієї команди виконуються рухи окремих шарнірів. Формат команди:

MOVE JOINT шарнір, кут

де шарнір – цілочисельна константа, значення якої вказує, який шарнір треба переміщати, ця константа повинна бути в межах 1 ... 6;

кут – величина пересування шарніра в градусах.

Приклад:

а) MOVE JOINT 2, 30 – шарнір 2 переміщає на 30 градусів у позитивному напрямку.

NO INCALL програмна команда (BREAK)

Команда заборони всяких переривань у вхідному проводі. Формат команди:

NO INCALL вхід

де хід – цілочисельна перемінна чи константа, значення якої вказує, переривання яких вхідних ліній забороняються (при цьому забороняються

переривання й у стані “1” і в стані “0”. Значення цієї перемінної чи константи повинне бути в межах 1 ... 16.

Приклади:

а) NO INCALL 1 – забороняє всі переривання вхідної лінії 1.

б) NO INCALL INX – забороняються переривання вхідних ліній, що задаються перемінної INX.

NO MIRROR програмна команда

Команда скасовує роботу в дзеркальному відображенні. Формат команди:

NO MIRROR

NO SCALE програмна команда

Команда скасовує роботу маніпулятора в заданому масштабі. Формат команди:

NO SCALE

ODELAY програмна команда

За допомогою цієї команди встановлюється час затримки схвата маніпулятора. Формат команди:

ODELAY час

де час – час затримки на розтискання схвата маніпулятора (0,00...327,67 сек.).

Примітка. Команда ODELAY не впливає на команди GO&OPEN і GOS&OPEN.

Приклад:

а) ODELAY 0,5 – затримка при розтискання схвата маніпулятора задається 0,5 сек.

OPEN програмна команда (BREAK)

По цій команді відбувається розтискання схвата маніпулятора. Формат команди:

OPEN

Значення часу за замовчуванням на розтискання схвата маніпулятора задається 0,2 сек. Час затримки можна змінювати за допомогою команди ODELAY.

OUT програмна команда

Ця команда керує станом вихідних ліній робота. Формат команди: OUT вихід, < вихід >, < вихід >, < вихід >

де вихід – цілочисельна чи константа перемінна, значення якої вказує, якими вихідними лініями керують. Якщо це значення позитивне, то вихідна лінія переводиться в стан “1”, якщо негативне – у стан “0”.

Приклади:

а) OUT 1, 100, – 3 – задає стан “1” на лініях 1 і 100, а на вихідній лінії 3 стан “0”.

б) OUT X – керує станом вихідних ліній, що задаються перемінної X. Якщо X позитивна, то вихідна лінія встановлюється в стан “1”, якщо X негативна, то в стан “0”.

OUTGROUP програмна команда

Установлює 16-канальний порт виходів. Формат команди:

OUTGROUP група 1 = < INGROUP > число 1 <арифметичний оператор > < INGROUP > < число 2 >

де число 1 – цілочисельна перемінна чи константа, значення якої указує установлюваний вихідний порт;

число 1, INGROUP, <ариф. опер.>, число 2, порівняйте з командою

SET.

Приклади:

а) OUTGROUP 1 = 0 – обнуляє вихідний порт 1.

б) OUTGROUP 2 = INGROUP X – переставляє значення вихідного порту, що задається перемінної X, у вихідний порт 2.

в) OUTGROUP X = AD + OFFSET – переміщає суму перемінних AD і OFFSET у вихідний порт, заданий перемінною X.

PRINT програмна команда

По цій команді здійснюється вивід на дисплей і роздруківка текстів і/чи значень перемінних. Формат команди:

PRINT < текст >, < число >

де текст – довільна послідовність символів, виведена на дисплей. Послідовність починається і закінчується символом “,”;

число – цілочисельна перемінна чи константа, значення якої виводиться на термінал після тексту.

Приклади:

а) PRINT 'CHANGE TOOL' – виводить на дисплей CHANGE TOOL.

б) PRINT 'WARNING', 5 – виводить на дисплей текст WARNING 5.

RETURN програмна команда

По цій команді здійснюється повернення від підпрограми до програми, яка раніше викликала підпрограму. Формат команди:

RETURN < число >

де число – цілочисельна перемінна чи константа, значення якої вказує, скільки команд треба пройти у викликаній програмі перед поверненням. Якщо значення дорівнює 0 чи воно не задано, то відбувається повернення в головну програму використовуючи команду CALL.

Усі підпрограми і підпрограми переривань повинні закінчуватися командою RETURN. Якщо команди RETURN нема, то керування передається в початок головної програми (тобто в програму, що запускається командою RUN).

Примітка. Коли обчислюється місце повернення в програму, рядки коментарів програми в розрахунок не приймаються.

Приклади:

а) RETURN – повернення з підпрограми.

б) RETURN 1 – перед поверненням пропускають одну команду після команди CALL (виклик).

в) RETURN X – перед поверненням у головну програму після команди CALL буде пропущене число команд програми, задане перемінною X.

RUNOUT програмна команда

Цією командою задається, які вихідні лінії обнулити, коли виконання програми переривається через помилку, чи натисканням кнопки WAIT (чекання), і при нормальному виконанні всіх циклів програми. Формат команди:

RUNOUT вихід, < вихід >, < вихід >, < вихід >

де вихід – цілочисельна перемінна чи константа, величина якої позначає вихідну лінію, яку треба обнулити. Якщо ця величина від'ємна, то обнуління вихідної лінії скасовується. При запуску програми по команді RUN обнуління усіх вихідних ліній забороняється. При виконанні команди RUNOUT відповідний порт виходу не змінює свого стану, а його станом керують, як правило, за допомогою команд OUT і OTGROUP. Вплив команди RUNOUT виявляється тільки тоді, коли програма зупиниться. Тоді всі порти виходів, що були зазначені в команді RUNOUT, будуть обнулені, приведені в стан "0". Якщо за допомогою команди CONTINUE виконання

програми продовжене, обнуленні вихідні лінії повернуться в той же стан, що у них був до зупинки програми.

Приклади:

а) RUNOUT 3, 5 – здійснюється обнуління ліній 3 і 5.

б) RUNOUT X – здійснюється обнуління вихідної лінії, заданої перемінною X, якщо значення X позитивне. Якщо ж значення X від'ємне, то обнуління вихідної лінії стирається.

в) RUNOUT –2 – стирає режим обнуління вихідної лінії 2.

SCALE програмна команда

Ця команда здійснює масштабування щодо якої-небудь точки. Формат команди:

SCALE точка = < коефіцієнт X >, < коеф. У >, < коеф. Z >

де точка – координатна точка, щодо якої треба виконати операцію масштабування;

масштаб X – цілочисельна чи константа перемінна, котра позначає масштабування щодо осі X. Коефіцієнт 1000 відповідає масштабному коефіцієнту 1;

масштаб У – аналогічно зазначеному вище, але по осі У;

масштаб Z – аналогічно зазначеному вище, але по осі Z.

Приклади:

а) SCALE CORNER = 2000, 2000, 2000 GO CORNER (X1) – інструмент маніпулятора переміщається в комбіновану точку, утворену точкою CORNER (кут) і точкою X1. Значення по осях X, У, Z збільшуються на 2 (тобто коефіцієнт масштабування 2000).

б) SCALE CORNER = XSCALE, YSCALE, ZSCALE GO CORNER (X1) – інструмент маніпулятора переміщається в комбіновану точку, утворену точкою CORNER і точкою X1. Масштабні переміщення по осях X,У,Z ви-

значаються перемінними XSCALE, YSCALE, ZSCALE.

Примітка. Треба пам'ятати, що в даний момент може бути задана тільки одна команда масштабування. Якщо ж задається кілька таких команд, то виконується тільки остання з них. Режим RUN автоматично обнуляє масштабування до запуску програми. Рекомендується, щоб використовувана в команді SCALE точка була сформована командою FRAME.

SET програмна команда

За допомогою цієї команди обчислюється значення для цілочисельної перемінної. Формат команди:

SET перемінна = < INGROUP > число 1 < арифметичний оператор > < INGROUP > < число 2 >

де число 1 – цілочисельна перемінна, величина якої задається по результату арифметичної операції;

число 2 – цілочисельна перемінна чи константа. Якщо перед числом 1 стоїть оператор INGROUP, то використовується значення порту входів, задане перемінною 1.

арифметичний оператор – арифметична операція перемінними числом 1 і числом 2. Оператори такі:

| | |
|---|---|
| + | – додавання |
| - | – вирахування |
| * | – множення |
| / | – розподіл |
| MOD | – залишок, різниця, модуль |
| AND | – подвійне “І” |
| OR | – подвійне “АБО” |

число 2 = число 1

Приклади:

а) SET X = O – обнуляє перемінну X.

б) SET X = X + 1 – до перемінної X додається 1.

В) SET X = INGROUP 1 + X – до перемінної X додається величина 1 групи портів входів.

г) SET X = 0 – 1 – змінює знак перемінної X.

SHIFT програмна команда

Ця команда використовується для переміщення точки в основній системі координат. Формат команди:

SHIFT точка = < dx >, < dy >, < dz >

де dx – відстань (мм), що додається до значення X точки;

dy – відстань(мм), що додається до значення Y точки;

dz – відстань (мм), що додається до значення Z точки.

Приклад:

а) SHIFT LOCI = 100, , – 200 – до значення X точки додається 100 мм, а до значення Z точки додається 200 мм.

SPEED програмна команда

Робота цієї команди аналогічно моніторній директиві SPEED, тобто команда задає базову швидкість руху маніпулятора. Формат команди:

SPEED швидкість

де швидкість – швидкість руху інструмента маніпулятора (руху по прямій, мм/сек).

SPEED NEXT програмна команда

Ця команда встановлює базову швидкість руху, що дійсна тільки на час наступної команди руху. Формат команди:

SPEED NEXT швидкість

де швидкість – швидкість наступного руху (мм/сек).

Приклад:

а) SPEED NEXT 30

GOS A – переміщення в точку А відбувається зі швидкістю 30 мм/сек. (Передбачається, що масштабний коефіцієнт швидкості SPEED % = 100).

SPEED % програмна команда

Робота цієї команди аналогічно відповідній моніторній директиві, тобто вона задає масштабний коефіцієнт швидкості. Формат команди:

SPEED % масштабний коефіцієнт швидкості

де масштабний коефіцієнт швидкості – цілочисельна перемінна чи константа, що позначає коефіцієнт масштабування швидкості у відсотках, на який треба помножити базову швидкість руху. У результаті виходить швидкість руху робота.

STOP програмна команда (BREAK)

Ця команда переривае виконання програми. Формат команди:

STOP < текст >, < число >

де текст – послідовність символів, виведена на дисплей. Послідовність символів починається і закінчується символом “,”.

число – цілочисельна перемінна чи константа, значення якої виводиться на дисплей після тексту.

Примітка. Після команди STOP продовжити виконання програми можна за допомогою директиви CONTINUE.

Приклад:

а) STOP 'CHANGE TOOL' – виводить на дисплей текст CHANGE

TOOL, після чого виконання програми зупиняється.

TMOVE і TMOVES програмні команди

За допомогою цих команд реалізується запуск переміщення маніпулятора в системі координат робочого інструмента робота. Формат команди:

TMOVE < dx >, < dy >, < dz >

TMOVES < dx >, < dy >, < dz >

де dx – довжина руху маніпулятора, обмірювана по осі X системи координат інструмента;

dy – те ж, що вище, але по осі У;

dz – те ж, що вище, але по осі Z.

При команді TMOVE траєкторія руху – це інтерпольований рух шарніра, при команді TMOVES – це лінійний рух. Рух по всім трьох осях відбувається одночасно (тобто не так, щоб спершу по осі X, потім по осі У, а потім по осі Z).

Приклади:

а) TMOVE 10, , 20 – маніпулятор переміщається на 10 мм по осі X і на 20 мм по осі Z у системі координат інструмента.

б) TMOVES –100, –100, –100 – маніпулятор переміщається по прямій на –100 мм по всіх осях системи координат інструмента.

TOL NARROW і TOL WIDE програмні команди

Ці команди визначають точність, з якою маніпулятор досягає заданої точки. По команді TOL NARROW маніпулятор буде рухатися до мети з точністю біля ±40 імпульсів енкодера на кожен шарнір. По команді TOL WIDE маніпулятор не перевіряє точність наближення до мети. Вона лежить у деякому припустимому діапазоні, так що час виконання програми зменшується. Формат команди:

TOL NARROW

TOL WIDE

Коли маніпулятор пересувається з однієї точки в іншу по безупинній траєкторії, ці команди не впливають на рух. Якщо ж траєкторія переривається (наприклад, у програмі є команда затримки DELAY), то при необхідності можна контролювати точність руху.

TOOL програмна команда (BREAK)

Цією командою здійснюється зрушення системи координат інструмента. Формат команди:

TOOL < dx >, < dy >, < dz >, < do >, < da >, < dt >

де dx – величина зрушення системи координат (мм) інструмента робота по осі X;

dy – величина зрушення (мм) по осі У;

dz – величина зрушення (мм) по осі Z;

do – величина зрушення (мм) кута “o” (градус);

da – величина зрушення (мм) кута “a” (градус);

dt – величина зрушення (мм) кута “t” (градус).

Приклад:

а) TOOL , , 100 – задаються такі величини зрушення системи координат інструмента робота, при яких наконечник інструмента розміщається на відмітці 100 мм осі Z системи координат інструмента.

WAIT IN програмна команда (BREAK)

По цій команді перевіряють вхідні лінії й очікують доти, поки вони не прийдуть у потрібний стан. Формат команди:

WAIT IN вхід, < вхід >, < вхід >, < вхід >

де вхід – цілочисельна перемінна чи константа, величина якої позна-

чає лінію, що перевіряється. Якщо ця величина позитивна, то перевіряється стан “1”, якщо негативна, то перевіряється стан “0” порт. Кількість вхідних ліній, які можна перевірити, складає 1, 2, 3 чи 4. За умовами перевірки потрібно, щоб усі вхідні лінії, зазначені в команді, одночасно були б у визначеному стані.

Приклади:

а) WAIT IN –1 – очікує до того, поки вихідна лінія 1 не перейде в стан “0”.

б) WAIT IN 3, X – очікує до того, поки вихідна лінія 3 і X не перейде в стан “1” (якщо X позитивна величина) чи в стан “0” (якщо X від’ємна величина).

WAIT LOAD програмна команда (BREAK)

За допомогою цієї команди можна програмним способом перевірити, чи відбуваються в даний момент які-небудь операції завантаження з гнучкого диска або видалення на ньому (по командах LOAD і DLOAD з аргументом NOBREAK). Якщо такі операції виконуються, то виконання програми переривається командою WAIT LOAD. По закінченні роботи з дисками виконання програми продовжиться. Формат команди:

WAIT LOAD

WEAVE програмна команда (BREAK)

Робота цієї команди аналогічно роботі однойменної моніторної директиви. Цією командою встановлюються, задаються параметри коливального руху. Формат команди:

WEAVE амплітуда, <час коливання >, < затримка >

де амплітуда – подвійна амплітуда коливання, її максимальне значення 256 мм. Якщо амплітуда дорівнює 0, то коливання не буде;

час коливання – час одного циклу коливання, дорівнює часу руху подвійної амплітуди, він лежить у межах 0,00...327,67 сек;

затримка – задає час зупинки, затримки руху робота в крайніх позиціях коливання. Його значення лежить у межах 0,00...327,67 сек.

***Редагування***

*Команди редагування, загальне*

Задача і призначення редактора – скласти прикладні програми робота на основі програмних команд, набраних оператором із клавіатури. За допомогою редактора можна створювати нові програми і редагувати вже існуючі. Перехід у режим редагування починається з моніторної директиви

EDIT < програма >

де програма – назва програми, яку треба відредагувати. Якщо назва програми не зазначена, то мається на увазі назва програми попереднього редагування.

Якщо редактор готовий почати роботу, тобто готовий приймати команди, що вводяться оператором із клавіатури, то на екран дисплея виводиться номер рядка програми. Після цього номера оператор може записати команди двох видів – команди редагування і програмні команди. Останні записуються в ЗП, якщо вони були правильно набрані. Перед кожним рядком програми можна поставити адреси рядка (цілочисельна константа), якому можна використовувати при розгалуженні програми (дивись, наприклад, команду JUMP). Та ж сама адреса не може бути в двох різних рядків програми. Нижче приведений простий приклад написання нової програми:

> EDIT PRG1 (cr)

Program PRG1

1. SET I = 0 (cr)

2. 10 SHIFT LOCI = 10 (cr)

3. GO LOCI (сг)

4. SET I = I + 1 (сг)

5. IF I < 15 THEN JUMP 10 (сг)

До числа команд редагування відносяться наступні:

збільшення кількості рядків програми.

P < рядок >, < кількість рядків > – вивід заданої кількості рядків програми.

D < кількість рядків > – стирання рядків програми.

R текст1 ^ текст2 – заміна тексту 1 на текст 2.

RA текст1 ^ текст2 – заміна тексту 1 текстом 2 у всій програмі.

TGO точка – ініціалізація режиму навчання робота (інтерполяційний рух робота).

TGOS точка – ініціалізація режиму навчання робота (руху робота по прямій).

E – повернення з режиму редагування в режим монітора.

*Команди редагування*

D

Командою D (DELETE) стираються рядки програми. Формат команди:

D < кількість рядків >

де кількість рядків – кількість рядків, яке треба стерти. Значення за замовчуванням кількості рядків – 1. Стирання здійснюється з поточного рядка, тобто з того, на якому була дана команда D.

Наприклад:

а) D 5 – стирається 5 наступних рядків.

б) D – стирається 1 рядок програми.

E

Команда закінчення редагування, виходу з цього режиму (EXIT). Фо-

рмат команди:

E

I

Команда I (INSERT) служить для додавання рядків програми в середину програми. Формат команди:

I

При завданні команди I на дисплеї повинний з'являтися символ I>, після чого можна записати нові рядки програми. Операція додавання закінчується введенням порожнього рядка – натиснути символ (cr).

P

Команда P (PRINT) виводить рядки програми на дисплей. Формат команди:

P < рядок >, <кількість рядків >

де рядок – номер того рядка програми, з якого починається вивід на дисплей. Передбачається, що це поточний рядок. Якщо зазначений номер зі знаком мінус, то треба відрахувати назад відповідну кількість рядків.

кількість рядків – кількість рядків, яке треба вивести на дисплей. За замовчуванням воно дорівнює “1”.

Примітка. Якщо з командою P не зазначені ні номер, ні кількість рядків, тобто була виведена тільки команда P, то виводиться перший рядок програми.

Наприклад:

а) P, 10 – виводяться наступні 10 рядків;

б) P 5 – виводиться п'ятий рядок;

в) P – виводиться перший рядок.

R і RA

Ці команди заміщають послідовність символів у командному рядку на іншу (REPLACE, REPLACE ALL). При цьому команда R стосується тільки одного рядка, що редагується, а команда RA усіх рядків програми (покрокове заміщення). Формат команди:

R текст1 ^ текст2

RA текст1 ^ текст2

де текст1 – послідовність символів, яку треба замінити на текст2;

^ – символ поділу між текстом 1 і 2;

текст2 – послідовність символів, що розміститься на місці тексту1, якщо вона буде знайдена у програмі.

Примітка. Якщо вводиться за допомогою R чи RA нова команда і вона виявилась невірною, то на дисплеї виводиться повідомлення про помилку, а команда редагування RA зупиниться на даному рядку програми.

Наприклад:

а) R 1 ^ 2 – заміняє 1, знайдену в рядку програми, на 2.

б) RA MOVE ^ MOVES – замінить усі команди програми MOVE на команду MOVES.

TGO і TGOS

Ці команди редагування використовується для створення програм за допомогою пульта ручного керування (TEACH GO і TEACH GOS). Формат команд:

TGO точка

TGOS точка

де точка – координатна, комбінована чи абсолютна точка.

Команда редагування TGO використовується для формування програмних команд GO&OPEN і GO&CLOSE (розтиснути/зжати схват маніпулятора). За допомогою команди редагування TGOS формуються команди GOS&OPEN і GOS&CLOSE. Кожна з цих 4-х команд складається натисканням кнопки STEP на пульті руч- ного керування. Одночасно запису-

ється поточне положення маніпулятора як величина точки. До назви точки додається індекс, що автоматично зросте на 1 при кожному натисканні кнопки STEP (крок) (Дивись моніторну директиву LTEACH).

Наприклад:

а) TGOS A – складає наступну програму (передбачається, що схват маніпулятора стиснутий).

GOS&CLOSE A0

GOS&CLOSE A1 і т.д.

### 5.6 Порядок виконання роботи

Лабораторну роботу виконують у наступній послідовності:

1) Ознайомитись з призначенням промислового робота, його технічною характеристикою й основними вузлами.
2) Ознайомитись з програмуванням промислового робота РМ-01.
3) Відповідно до виданого завдання скласти керуючу програму.
4) Під контролем викладача настроїти робот і відпрацювати по заданій програмі.
5) Скласти звіт про виконану роботу.

### 5.7 Зміст звіту

Звіт по лабораторній роботі повинний містити наступні розділи:

1) Назва лабораторної роботи.
2) Мета роботи.
3) Призначення робота і коротка його технічна характеристика.
4) Конструкція і принцип дії маніпулятора (з рисунками).
5) Система координат маніпулятора (з рисунками).
6) Запуск і навчання робота.
7) Порядок виконання роботи.
8) Текст керуючої програми з рисунком, який її пояснює.
9) Висновки по роботі.

## Рекомендована література

1. Промышленные роботы в машиностроении: Альбом схем и чертежей: Учеб. Пособие для технических вузов / Ю.П. Соломенцев, К.П. Жуков, Ю.А. Павлов и др.; Под общ. Ред. Ю.М. Соломенцева. – М.: Машиностроение, 1986. – 140 с.: ил.

2. Попов Е.П., Письменный Г.В. Основы робототехники: Введение в специальность: Учеб. для вузов по спец. “Робототехн. системы и комплексы” – М.: Высш. шк., 1990. – 224 с.: ил.

3. Металлорежущие станки: Учебник для машиностроительных втузов/ Под ред. В.Э. Пуша. – М.: Машиностроение, 1985. – 256 с., ил.

4. Автоматическая загрузка технологических машин: Справочник/ И.С. Бляхеров, Г.В. Варьяш, А.А. Иванов и др.; Под общ. ред. И.А. Клусова. – М.: Машиностроение, 1990. – 400 с.: ил.

5. Промышленные роботы агрегатно-модульного типа / Е.И. Воробъев, Ю.Г. Козырев, В.И. Царенко; Под общ. ред. Е.П. Попова. – М.: Машиностроение, 1988. – 240 с.: ил. – (Автоматические манипуляторы и робототехнические системы).

6. Козырев Ю.Г. Промышленные роботы: Справочник. – 2-е изд., перераб. и доп. – М.: Машиностроение, 1988. – 392 с: ил.

## Перелік посилань

1. Промислові роботи. Альбом до методичних вказівок до виконання лабораторних робіт з дисципліни “Промислові роботи: будова, програмування, експлуатація” для здобувачів другого (магістерського) рівня вищої освіти за спеціальністю 133 “Галузеве машинобудування” освітньо-професійної програми “Галузеве машинобудування” / Укл.: Кальченко В.В., Пасов Г.В. – Чернігів: НУ “Чернігівська політехніка”, 2021. – 55 с.

2. Промышленный робот М20П.40.01. Руководство по программированию робота. Стара Загора, 1988, – 490 с.

3. Промышленный робот РМ-01. Руководство по программированию робота. А/О “НОКИА”. Отдел робототехники. Хельсинки, 1987, – 130 с.

ЗМІСТ Стор