

**Tentamen i Robotteknik MPR160, 28 april 2000**

Lärare: Anders Boström ank 1526

Per Nyqvist 073-986 07 03

Tillåtna hjälpmedel: Typgodkända kalkylatorer och alla formelsamlingar.

Betygslista anslås torsdag 11/5 på Robotlaboratoriets anslagstavla. Granskning sker hos Gunvor Johansson fredagen 12/5 kl 9-11 på inst. för prod teknik.

**Betygsgränser: 30-39p=betyg 3, 40-49p=betyg 4, 50-60p=betyg 5**

- 1a Förklara hur AGVer (förarlösa truckar) navigerar med följande tre metoder:
- Dödräkning
  - Laser navigering
  - Induktiv slinga
- 3p
- 1b Beskriv tre koordinatsystem som man använder vid programmering av industrirobotar. Motivera vid vilka tillfällen de olika koordinatsystemen är lämpligast att använda för att positionera roboten.
- 3p
- 1c Vad betyder förkortningen tcp och varför används denna funktion.
- 2p
- 1d Förklara begreppen: absolutfel och repeterfel i samband med robotprogrammering, rita gärna en skiss.
- 2p
- 2a Förklara begreppen transportabilitet och spegling i samband med robotprogrammering.
- 2p
- 2b Beskriv:
- hur en SCARA robot är uppbyggd,
  - hur "SCARA"-effekten fungerar
  - vad förkortningen SCARA står för
- 3p
- 2c Beskriv detaljerat hur en "harmonic drive" är:
- uppbyggd,
  - vad som är ingående och utgående axel
  - hur den fungerar.
- 5p

3a Beskriv detaljerat hur följande robottyper är kinematiskt uppbyggda och vilken arbetsvolym de har:

- Articulated robotarm (universell)
- Cylindrisk robotarm
- Sfärisk robotarm

6p

3b Nämn två fördelar med offline programmering av industrirobotar.

2p

3c Nämn två fördelar med online programmering av industrirobotar.

2p

-----

#### Rotationsmatriser

$$\mathbf{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$$\mathbf{Rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$\mathbf{Rot}(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 4 En IBM7545 SCARA-robot utför montering i 2 separata stationer. I vardera stationen monteras i tur och ordning 4 st olika komponenter: "1", "2", "3" och "4". Roboten hanterar komponenterna "1" och "3" med ett visst verktyg samt använder ett annat verktyg för komponenterna "2" och "4" (dvs verktygsbyte måste ske efter montering av påföljande komponent i samma station).

En PLC styr kringutrustning och ger roboten klarsignal för montering på respektive station genom att ettställa motsvarande ingång (ingång 1 = 1 då station 1 är klar för montering och ingång 2 = 1 då station 2 är klar för montering). När roboten har monterat alla fyra komponenterna i en station annonseras detta genom att roboten ettställer motsvarande utgång (utgång 1 = 1 när montering är klar i station 1 och utgång 2 = 1 när montering är klar i station 2). PLC'n kvitterar en ettställd utgång genom att nollställa motsvarande ingång (1 eller 2) och roboten ska då i sin tur omedelbart nollställa den aktuella utgången (1 eller 2).

Vid programstart är roboten i sitt hemmaläge med hämtat verktyg för komponenterna "2" och "4" samt väntar på att få montera första komponenten i någon av stationerna.

Skriv ett AML/E-program för den ovan beskrivna robotuppgiften. Lösningen presenteras med en programlistning (se programexempel nedan) samt ska vara **optimerad** (dvs minimalt antal verktygsbyten). Använd endast nedan beskrivna rutiner och kommandon. Skriv tydligt och använd stora bokstäver! (10p)

**Fördefinierade rutiner (behöver inte deklarerars):**

Subrutin	Beskrivning
MONT_ST1(VAR);	Hämtar och monterar sedan en komponent VAR i station 1, dvs anropet MONT_ST1(3); innebär att roboten hämtar en komponent "3" och monterar denna i station 1.
MONT_ST2(VAR);	Hämtar och monterar sedan en komponent VAR i station 2, dvs anropet MONT_ST2(3); innebär att roboten hämtar en komponent "3" och monterar denna i station 2.
TOOL_13;	Lämnar först verktyget för komponenterna "2" och "4" samt hämtar sedan verktyget för komponenterna "1" och "3".
TOOL_24;	Lämnar först verktyget för komponenterna "1" och "3" samt hämtar sedan verktyget för komponenterna "2" och "4".

**Befintliga kommandon och syntax:**

Kommando	Beskrivning
WRITEO(DO,VAL);	Utgång DO sätts till värdet VAL.
TESTI(DI,VAL,HOPP);	Kontrollerar om en ingång DI har värdet VAL, vilket medför hopp till HOPP. Har ingången inte värdet VAL exekveras nästa instruktion.
HOPP::;	Markerar en hoppadress (label). Hopp får bara ske inom samma subrutin.
BRANCH(HOPP);	Ovillkorligt hopp till hoppadressen HOPP.
NAMN: STATIC COUNTER;	Definierar en global räknare med namnet NAMN.
SETC(NAMN,VAL);	Sätter en räknare med namnet NAMN till värdet VAL.
INCR(NAMN);	Räknaren NAMN ökas med 1.
TESTC(NAMN,VAL,HOPP);	Kontrollerar om en räknare NAMN har värdet VAL, vilket medför hopp till HOPP. Har räknaren inte värdet VAL exekveras nästa instruktion.
NAMN: SUBR(VAR,VAR1..);	Definierar en subrutin med namnet NAMN och variablerna med namnen VAR,VAR1.. blir lokala..
NAMN(VAR,VAR1..);	Anropar subrutinen med namnet NAMN med variablerna VAR,VAR1...
END;	Markerar slutet på subrutin.

**Programexempel:**

```

--Deklarationer
RAKNARE: STATIC COUNTER;

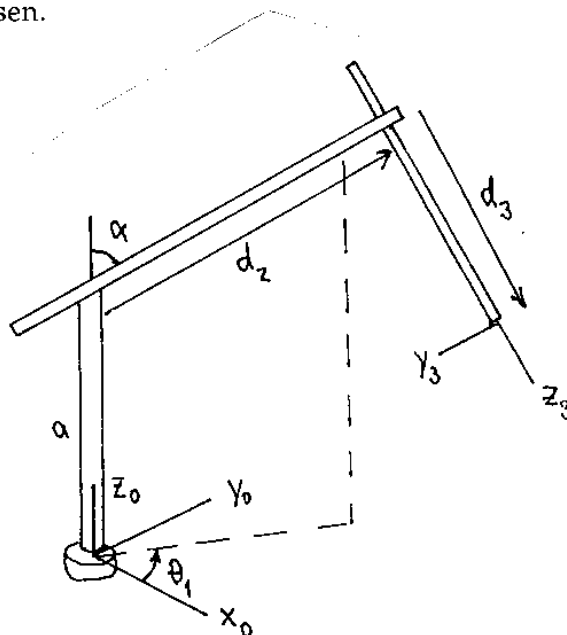
-- Definition huvudsubrutinen
PROGEX:SUBR;

-- Subrutiner
SUB:SUBR(VAR);
WRITEO(VAR,0);
SETC(RAKNARE,1);
INCR(RAKNARE);
MONT_ST1(RAKNARE);
END;

-- Sist kommer huvudprogrammet
TESTI(2,1,SLUT);
BRANCH(SLUT);
SUB(1);
SLUT;;
TESTC(RAKNARE,0,SLUT);
TOOL_13;
END;

```

5. Roboten i figuren har de tre länkvariablerna  $\theta_1$ ,  $d_2$  och  $d_3$ . Vinkeln  $\alpha$  mellan första och andra länken är fix.  $y_3$ -axeln är parallell med andra länken. Bestäm den homogena transformationsmatrisen. (8p)



6. Roboten i figuren har tre länkvariabler:  $d_1$ ,  $d_2$ ,  $\theta_3$ . Den andra länken fungerar som rotationsaxel för tredje länken.

- a) Bestäm  $d_1$ ,  $d_2$ ,  $\theta_3$  då verktygets läge är  $(p_x, p_y, p_z) = (a, a/2, a)$ . Endast en lösning behöver ges. (6p)
- b) Bestäm Jacobimatrisen ( $3 \times 3$ ). (6p)

