

**Tentamen i Robotteknik MPR160, 19 december 1998**

Lärare: Rolf Berlin ank 1286

Anders Boström ank 1526

Tillåtna hjälpmedel: Typgodkända kalkylatorer och alla formelsamlingar.

OBS bifogat finns AML/E-instruktioner och formler på sidorna 5-7!

Betygslista anslås tisdag 12/1 på Robotlaboratoriets anslagstavla. Granskning sker hos Gunvor Johansson onsdagen 13/1 kl 13-15 på inst. för prod teknik.

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

- 1a Beskriv det karakteristiska och ge ett exempel på följande tre robottyper:
- Telemanipulator
  - Servicerobot
  - Industrirobot 3p
- 1b Näm tre branscher som med sina produktionsbehov kraftigt drivit utvecklingen och användandet av robotteknik framåt. 3p
- 1c Möjligheten att robotisera processer är beroende av kunskap om processen och möjligheten att med sensorer styra processen, utvecklingen av nya robotapplikationer kan därför ta mycket lång tid. Beskriv med bakgrund av detta hur utvecklingen av bågsvetsning stegvis realiserades, beskriv minst tre faser i denna utveckling. 4p
- 2a Beskriv 12 funktioner (knappar) som finns på en programmeringslåda för en industrirobot. (Enskilda bokstäver eller siffror duger ej som svar) 6p
- 2b Beskriv detaljerat det karakteristiska med följande två programmeringsmetoder för industrirobotar:
- On-line programmering
  - Off-line programmering 4p

3a Förklara hur AGVer (förlösa truckar) navigerar med följande fyra metoder:

- Optisk slinga
- Laser navigering
- Induktiv slinga
- Dödräkning

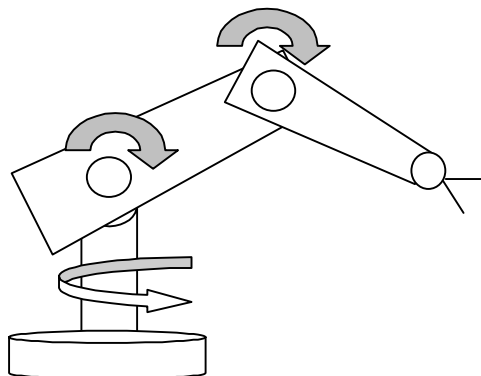
4p

3b Gör ett histogram av följande gråskalebild från ett kamerasystem. Förklara även vad ett histogram används till.

0	2	3	3	2	1	1	0	0	0
1	2	3	5	4	3	2	1	1	0
0	2	4	6	7	5	4	4	2	1
1	3	4	4	6	6	5	5	4	2
0	2	4	6	7	7	5	4	2	1
1	3	3	5	5	5	2	2	2	0
0	0	2	3	3	4	2	2	1	0
0	1	2	2	1	1	0	0	0	0

3p

3c Beskriv hur robotarmen nedan kan nå samma position med verktyget på fyra olika sätt, rita gärna.



3p

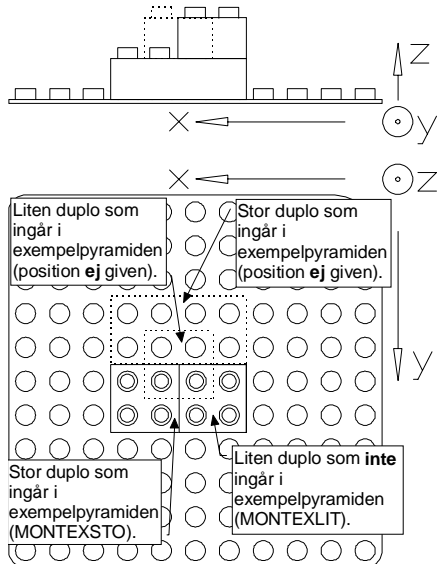
4. Skriv ett **optimerat** AML/E program för en IBM7545 SCARA "DUPLO-robot". Roboten ska bygga 4st homogena "pyramider" bestående av en liten och två stora duplobitar. De stora duplobitarna sitter intill varandra och bildar ett kvadratisk rätblock. Den lilla duplobiten sitter centrerad på toppen av rätblocket (se figur nedan).

Robotcellen är identisk med den som användes i IBM-laborationen förutom att en PLC nu är installerad och kan positionera indexeringsbordet samtidigt som roboten förflyttar sig (multitasking). Duploplattorna är redan utlagda och på dessa monteras de stora duplobitarna. Gripdonsfingrarna griper kraftfullt på sidan av bitarna (endast långsidan för stor duplo), vilket innebär att den andra stora duplobiten inte kan monteras direkt i sin korrekta position.

Från början befinner sig roboten i hemmaläget (HEM) med duplogripdonet öppnat och där skall roboten befinna sig när "pyramiderna" är klara. Nya positioner uträknas med hjälp av befintliga positioner samt mått på duplobitarna - se nedan. Förflyttning i sidled sker på Z-nivån=UPP och i Z-led sker förflyttning endast med ZMOVE-kommandot.

Lösningen presenteras med en programlistning (se programexempel nedan). De globala deklARATIONERNA som finns angivna nedan behöver inte ingå i lösningen. Skriv tydligt och använd stora bokstäver! (10p)

### Geometri:



Längdenhet är [mm]. Monteringspositionen (x, y, z, r=vridningsvinkeln på handleden) är för den stora duplobiten, MONTEXSTO -> (0, 540, NIVA1Z, 0) och för den lilla duplobiten, MONTEXLIT -> (-15, 540, NIVA2Z, 0).

Måtten på bitarna är:

**Stor duplo:** 30X60 mm, Höjd: 20 mm.

**Liten duplo:** 30X30 mm, Höjd: 20 mm.

De utritade axlarna är parallella med motsvarande axlar i robotens baskoordinatsystem.

De två **streckade** bitarna tillsammans med den stora duplobiten visar ett exempel av pyramiden.

### Roboten använder PLC'n enligt följande:

Signal	Beskrivning
Utgång 11	<b>Puls</b> (=1) under 1 s ger PLC'n order om att ställa indexeringsbordet i position 1.
Utgång 12	<b>Puls</b> (=1) under 1 s ger PLC'n order om att ställa indexeringsbordet i position 2.
Utgång 13	<b>Puls</b> (=1) under 1 s ger PLC'n order om att ställa indexeringsbordet i position 3.
Utgång 14	<b>Puls</b> (=1) under 1 s ger PLC'n order om att ställa indexeringsbordet i position 4.
Ingång 11	=1 om bordet är i position 1, =0 annars.
Ingång 12	=1 om bordet är i position 2, =0 annars.
Ingång 13	=1 om bordet är i position 3, =0 annars.
Ingång 14	=1 om bordet är i position 4, =0 annars.

**Deklarationer.** Följande positioner är globalt definierade och kan anropas med kommandona PMOVE, ZMOVE och som variabler till en subrutin:

Deklaration	Beskrivning
HEM	Hemmaläget för roboten (z-värde = UPP)
UPP	Enda tillåtna z-värde innan roboten förflyttar sig i sidled med PMOVE
LITDU	Grovposition för hämtning av en liten duplo
LITDUZ	Absolutvärde i z-led för hämtning av en liten duplo
STODU	Grovposition för hämtning av en stor duplo
STODUZ	Absolutvärde i z-led för hämtning av en stor duplo
MONTEXSTO	Grovposition för montering av den stora duplobiten i figurexemplet ovan
NIVA1Z	Absolutvärde i z-led för montering av den stora duplobiten i figurexemplet ovan
MONTEXLIT	Grovposition för montering av den lilla duplobiten i figurexemplet ovan
NIVA2Z	Absolutvärde i z-led för montering av den lilla duplobiten i figurexemplet ovan

### **Befintliga kommandon och syntax:**

Kommando	Beskrivning
PMOVE(POS);	Roboten går till den globalt definierade positionen POS.
ZMOVE(ZPOS);	Roboten rör sig endast i z-led och går till det globalt definierade z-värdet ZPOS.
GRASP;	Gripdonet stängs - ingen fördröjning är nödvändig.
RELEASE;	Gripdonet öppnas - ingen fördröjning är nödvändig.
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.
DELAY(TID);	Uppehåller exekveringen av programmet under TID sekunder.
NAMN: NEW....;	Definierar en global position eller z-värde med namnet NAMN.
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:**

```
--Först deklaras alla positioner och z-värden
LAMNA: NEW PT(15,540,UPP,90);
ONODIGZ: NEW -130;
RAKNARE: STATIC COUNTER;
-- Sedan definieras huvudsubrutinen
PROGEX:SUBR;
-- Därefter kommer subrutiner
SUB:SUBR(VAR,VARZ);
PMOVE(VAR);
ZMOVE(VARZ);
GRASP;
END;
```

```
-- Sist kommer huvudprogrammet
DELAY(2);
TESTI(12,1,SLUT);
BRANCH(SLUT);
SUB(LITDU,UPP);
SUB(LAMNA,NIVA2Z);
SUB(HEM,UPP);
SUB(LITDU,UPP);
WRITEO(12,0);
SLUT.;;
SETC(RAKNARE,1);
INCR(RAKNARE);
TESTC(RAKNARE,0,SLUT);
END;
```



## FORMLER

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

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

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

### Development of a determinant. Cofactor

Subdeterminant  $D_{ij}$  of order  $(n-1) \times (n-1)$  is formed by deleting the  $i^{\text{th}}$  row and the  $j^{\text{th}}$  column of  $D$ .

Cofactor  $A_{ij} = (-1)^{i+j} D_{ij}$ .

- |   |
|---|
| 11. $\det A = a_{i1}A_{i1} + a_{i2}A_{i2} + \dots + a_{in}A_{in}$ (development by $i^{\text{th}}$ row)<br>12. $\det A = a_{1j}A_{1j} + a_{2j}A_{2j} + \dots + a_{nj}A_{nj}$ (development by $j^{\text{th}}$ column) |
|---|

**Example.**

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 5 & 0 & 2 & 0 \\ 3 & 4 & 1 & 1 \\ 2 & 3 & 4 & 5 \end{vmatrix} = [\text{development by } 2^{\text{nd}} \text{ row}] = 5 \cdot (-1)^{2+1} \begin{vmatrix} 2 & 3 & 4 \\ 4 & 1 & 1 \\ 3 & 4 & 5 \end{vmatrix} + 2 \cdot (-1)^{2+2} \begin{vmatrix} 1 & 2 & 4 \\ 3 & 4 & 1 \\ 2 & 3 & 5 \end{vmatrix}$$

### Matrix inversion

13.  $A^{-1}$  exists  $\Leftrightarrow \det A \neq 0$ .

**Calculation of  $A^{-1}$**

a.  $[A^{-1}]_{ij} = \frac{1}{\det A} A_{ji}$  (cofactor method)

b. Assume, by elementary row operations,

$$[A|I] = \begin{bmatrix} a_{11} & \dots & a_{1n} & 1 & \dots & 0 \\ \dots & & & & & \\ a_{n1} & \dots & a_{nn} & 0 & \dots & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & \dots & 0 & b_{11} & \dots & b_{1n} \\ \dots & & & & & \\ 0 & \dots & 1 & b_{n1} & \dots & b_{nn} \end{bmatrix} = [IB].$$

Then  $B = A^{-1}$ . (Jacobi's method)

Special case,  $n=2$ :  $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$ .