

Tentamen i Robotteknik MPR160, 16 december 2000

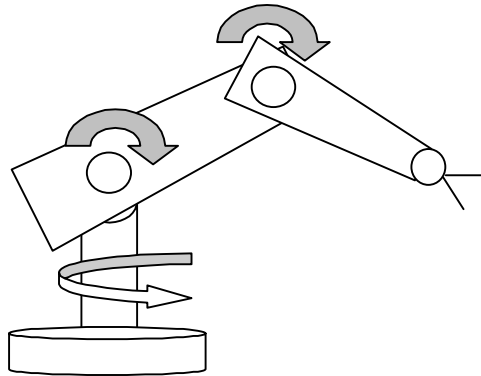
Lärare: Rolf Berlin ank 1286; 0707-99 24 89
Anders Boström ank 1526

Tillåtna hjälpmedel: Typgodkända kalkylatorer och alla formelsamlingar.

Betygslista anslås måndag 15/1 på Robotlaboratoriets anslagstavla. Granskning sker hos Gunvor Johansson onsdagen 17/1 kl 13-16 på inst. för prod teknik.

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

- 1a Beskriv hur robotarmen nedan kan nå samma position med verktyget på fyra olika sätt, rita gärna.



2p

- 1b Vad betyder förkortningen tcp och varför används denna funktion vid robotprogrammering. 2p
- 1c Förklara hur följande styrprinciper för robotrörelse fungerar, rita gärna: 6p
- synkroniserad PTP styrning
 - osynkroniserad PTP styrning
 - CP- styrning
- 2a Robotsäkerhet är ett centralt problem i alla robotceller, förklara hur ett hålldon används och fungerar i samband med att någon vistas i farligt område. 2p
- 2b Vid projektering av en robotinstallation genomför man som produktionstekniker normalt 4-5 faser, nämn fyra faser och beskriv översiktligt vad de innehåller. 2p
- 2c Förklara vad som menas med direkt drivning av robot länkarmar, nämn även en fördel och en nackdel med metoden. 2p

- 2d Beskriv detaljerat hur en "harmonic drive" är: 4p
- uppbyggd,
 - vad som är ingående och utgående axel (vid normal nedväxling)
 - hur den fungerar.
- 3a Nämn 6 krav på ett avancerat off-line system för robot programmering 3p
- 3b Förklara hur ett normalt industriellt vision system används för att identifiera ett objekt bland en mängd tidigare definierade. Beskriv minst två egenskaper som var och en eller tillsammans kan användas för identifiering. 2p
- 3c Förklara följande robotbegrepp : 3p
- märklast
 - lägesrepetitbarhet
 - arbetsvolym
- 3d Förklara hur AGV-navigering med "induktiv slinga" fungerar och hur man programmerar AGV'n att välja rätt väg för ett visst uppdrag i en layout med många korsningar och alternativa vägar 2p

=====

Rotationsmatriser

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

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

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

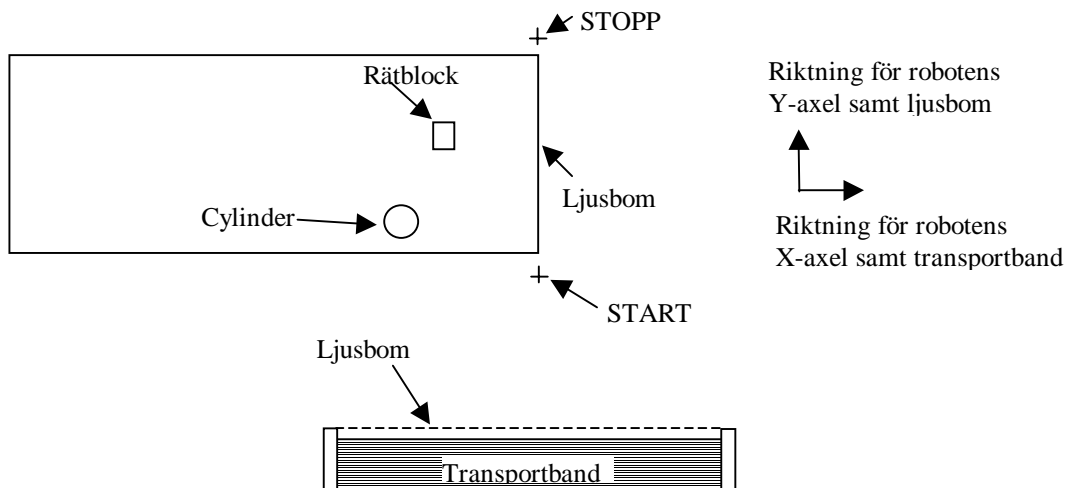
- 4.a Rätta nedanstående RAPID-program som innehåller 6 fel (svara genom att stryka felaktiga rader och skriva de korrekta i den högra kolumnen i svarsformuläret på nästa sida - **avdrag** görs för felaktigt svar). 5p

En ABB Irb1400 (S4) laddar en fräs med rätblock (6x8cm, höjd=8cm) och en svarv med cylindrar (radie = 5cm, höjd = 8cm). Roboten hämtar föremålen från ett transportband, vars riktning är parallell med baskoordinatsystemets X-axel. Vid änden av transportbandet finns en ljusbom (som är parallell med baskoordinatsystemets Y-axel). Ljusbommen är placerad 1cm över transportbandet samt är kopplad till ingången LJUSBOM, vilken ettställs då ljusbommen bryts (=0 annars).

Robotens "gripdon" utgörs av en sugpropp, som innehåller en sensor. Sensorn är kopplad till ingången SENSOR samt sitter centrerad i sugproppen. Förutsatt att sensorn är riktad rakt mot transportbandet ettställs SENSOR omedelbart då en höjdförändring detekteras (=0 då ingenting finns mellan sensorn och transportbandet). För att sökning (SearchL) ska fungera finns systemdefinierade positioner (START, TRÄFF och STOPP) där höjden över transportbandet är anpassad för både detektering och uppsugning samt läget i förhållande till ljusbommen endast är ett offset i z-led. Dessutom finns utgången INVERTERARE som vid ettställning inverterar (logiskt) SENSOR. Sugproppen aktiveras vid ettställning av utgången PROPP (=0 innebär inaktivering). Plockning (uppsugning) av ett föremål måste ske i ytcentrum på ovansidan.

Transportbandet styrs m h a utgången MOTOR (=1 innebär att transportbandet startas, =0 stoppar transportbandet). Föremålen ligger slumpvis (dock efter varandra och med visst mellanrum) på transportbandet och detta innebär att roboten först måste detektera position och typ av föremål, för att sedan hämta detta och ladda in i rätt maskin. Laddningen utförs genom att anropa 2st fördefinierade rutiner (SVARV och FRÄS).

Verktygsdata **TCP** används för alla positioner. Bortse ifrån eventuella fördröjningar samt antag att rätblocken är orienterade så att antingen kortsidan (6cm) eller långsidan (8cm) är parallella med ljusbommen.



Svarsformulär uppgift 4a

MODULE cylindrar_och_rätblock	
PROC main()	
Reset PROPP;	
Reset INVERTERARE;	
Set MOTOR;	
IF LJUSBOM = 1 THEN	
Reset MOTOR;	
sökning_längs_ljusbom;	
ELSE	
Set MOTOR;	
sökning_längs_ljusbom;	
END IF	
ENDPROC;	
PROC sökning_längs_ljusbom()	
MoveL START,v100,fine,TCP;	
SearchL \Stop,SENSOR,TRÄFF,STOPP,v100,TCP;	
kontroll;	
ERROR	
TRYNEXT;	
ENDPROC	
PROC kontroll()	
MoveL Offs(TRÄFF,-90,0,0),v100,fine,TCP;	
IF SENSOR=0 THEN	
cylinder;	
ELSE	
rätblock;	
END IF	
ENDPROC	
PROC cylinder()	
MoveL Offs(TRÄFF,-50,0,0),v100,z10,TCP;	
Set PROPP;	
MoveL Offs(TRÄFF,-50,0,100),v100,z10,TCP;	
SVARV;	
ENDPROC	
PROC rätblock()	
MoveL Offs(TRÄFF,-10,70,0),v100,fine,TCP;	
IF SENSOR=1 THEN	
MoveL Offs(TRÄFF,-30,40,0),v100,fine,TCP;	
Set PROPP;	
MoveL Offs(TRÄFF,-30,40,100),v100,fine,TCP;	
END IF	
Set INVERTERARE;	
IF SENSOR=0 THEN	
MoveL Offs(TRÄFF,-30,40,0),v100,fine,TCP;	
Set PROPP;	
MoveL Offs(TRÄFF,-30,40,100),v100,fine,TCP;	
END IF	
FRÄS;	
ENDPROC	
ENDMODULE	

- 4.b Skriv ett AML/E-program som får en IBM7545-robot att först gå till positionen (100,100,-100,0) och därefter gå så **linjärt som möjligt** (dvs med minsta möjliga avvikelser från den matematiskt perfekta linjära banan) till positionen (150,300,0,0). 5p

Befintliga kommandon och syntax:

Kommando	Beskrivning
PMOVE(POS);	Roboten går till den globalt definierade positionen POS.
ZMOVE(POSZ);	Roboten rör sig endast i z-led och går till det globalt definierade z-värdet ZPOS.
PMOVE(PT(100,100,-100,0));	Roboten går till positionen (100,100,-100,0).
ZMOVE(-100);	Roboten rör sig endast i z-led och går till z-värdet (-100).
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(15,540,0,90);
ONODIGZ: NEW -130;
RAKNARE: STATIC COUNTER;
-- Sedan definieras huvudsubrutinen
PROGEX:SUBR;
-- Därefter kommer subrutiner
SUB:SUBR(VAR,VARZ);
PMOVE(VAR);
ZMOVE(VARZ);
PMOVE(PT(15,VARZ,ONODIGZ,90));
END;
```

```
-- Sist kommer huvudprogrammet
DELAY(2);
BRANCH(SLUT);
SUB(LAMNA,100);
SLUT;;
SETC(RAKNARE,1);
INCR(RAKNARE);
TESTC(RAKNARE,0,SLUT);
END;
```

