

Robotarm och algebra

I denna laboration skall du lära dig lite mer om möjligheter att rita ut mer avancerade saker i MatLab, använda inbyggda matrisfunktioner samt öva på saker som är genomgådda i er kurs Linjär algebra.

Tidsåtgång

Denna laboration beräknas ta ca 7 timmar i anspråk (4 timmar bokade med assistent).

Mål

Du ska efter denna laboration känna till

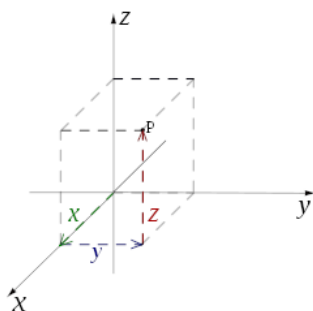
- ännu mer om hur MatLabs inbyggda grafiska möjligheter
- hur man gör rotationer, translationer m.m.
- diverse inbyggda saker i MatLab

Uppgift 1 - kopplad till er algebrakurs (ca 5 tim 20 min)

Du skall skriva en funktion som tar emot en parameter, som beskriver ett antal sektioner av en robotarm, och sen ritat ut robotarmen. Det är givet att robotarmen sitter fast i ett golv (som är ogenomträngligt) och att varje led är en "kul-led" som gör att två sektioner i princip kan ligga parallellt (d.v.s. sektionerna går åt varsitt håll) även om detta är fysiskt helt omöjligt. Den första sektionen (som sitter fast i golvet) har också en kulle, men denna sektion får givetvis inte gå neråt, d.v.s. genom golvet.

Ovanstående ger att vi kan vika alla sektioner (utom den första) 360 grader i alla riktningar. För att beskriva en sektion behöver vi 3 data som beskriver sektionen. Dessa är två vinklar som beskriver rotation runt z-axeln (första datat) samt rotation runt y-axeln (andra datat) samt längden på sektionen (tredje datat). I vilken ordning man roterar armen runt de olika axlarna är i denna uppgift mycket relevant. Startpositionen för varje ny sektion utgår givetvis ifrån föregående sektionens ändpunkt. Vinklarna beskrivs alltid utifrån att rotationerna sker runt origo.

Om den andra vinkeln för första sektionen är 0 grader står den första sektionen rakt upp från golvytan. Är denna vinkel 90 grader och den första vinkeln är 0 grader ligger första sektionen längs golvytan längs med x-axeln. Är båda vinklarna 90 grader ligger sektionen längs y-axeln. I detta system går alltså z-axeln uppåt. Figuren nedan visar hur axlarna ligger i förhållande till varandra.



Vill du ha detta mer visuellt kan du tänka dig att du står i origo och ritat x-axeln åt höger på golvet, y-axeln rakt fram och z-axeln går då rakt genom din kropp uppåt.

För att beskriva hela robotarmen lägger man alla sektionernas data efter varandra i en vektor av dimension $1 \times \text{”antal sektioner} \times 3\text{”}$ (3 sektioner ger alltså en 1×9 -vektor). Vi kallar denna för en specifikationsvektor i denna uppgift.

Vi håller oss alltid i samma koordinatsystem vilket leder till att alla sektioner kommer att roteras på samma sätt, men de har ett nytt ”origo” att rotera runt. Detta leder till att det räcker att hålla reda på var detta nya origo befinner sig i det fasta koordinatsystemet.

Tips 1: Roter alltid först runt y-axeln och sen runt z-axeln (moturs i båda fallen).

Tips 2: I MatLab finns det några saker som kan vara roliga för att rita saker. T.ex. *cylinder*, *sphere* och *surf* kan vara trevliga i denna uppgift. I den roterande kuben finns lite rotationer m.m.

Tips 3: Testa lite vad de olika funktionerna i tips 2 returnerar och planera för vad som behöver göras för att kunna rotera, translatera m.m. Det är inte säkert att man direkt kan applicera algebran på det man får från dessa funktioner.

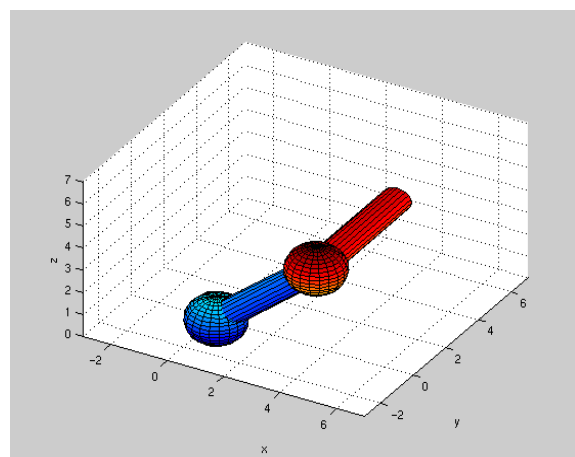
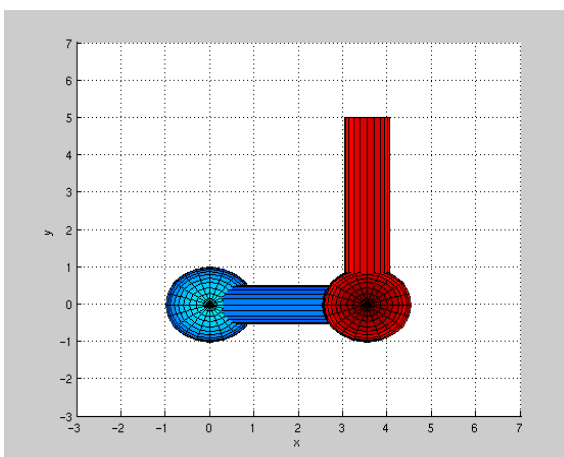
Uppgiften är att rita ut hela robotarmen givet specifikationsvektorn. Utritningsfunktionen skall heta *draw_robot_arm* och endast ta en specifikationsvektor som indata. Det är givetvis tillåtet att använda underfunktioner för att lösa delproblem (kanske till och med rekommenderat).

Exempel 1:

Antag att vi vill rita ut en robot som har 1 led förutom den som sitter fast i golvet. De två delarna skall vara 5 respektive 5 enheter långa. Den första sektionen skall inte roteras kring z-axeln, men den skall roteras 45 grader runt y-axeln. Motsvarande vinklar för den andra sektionen är ($z=90$, $y=90$). Anropet till din utritningsfunktion skall då innehålla 6 data enligt följande:

```
draw_robot_arm([0, 45, 5, 90, 90, 5]);
```

Detta anrop bör ge följande figur som resultat (sett uppifrån respektive som 3D-vy):



De olika färgerna i figurerna är något som MatLab själv valt. Man kan sätta färger och en massa annat på egen hand, men detta är något som ni inte behöver ge er in på.

I den vänstra bilden ser vi robotarmen uppifrån och den blå sektionens sfär har sitt centrum i koordinaten (0, 0, 0), d.v.s. hela koordinatsystemets origo. Z-axeln går ut ur figuren, d.v.s. mot dig. Den andra sektionen har sin sfär med sitt centrum i den koordinat där första sektionen slutar.

Hur ser man att detta är rätt? Vi tar det steg för steg och börjar med den första sektionen: Man tänker sig att man tittar mot origo från den positiva sidan av y-axeln och sen roterar den första sektionen 45 grader (motsols) runt y-axeln så blir den att luta så att den ligger mellan z-axeln och x-axeln. Detta verkar ok. Då rotationen kring z-axeln sen är 0 grader händer inget mer med den första sektionen.

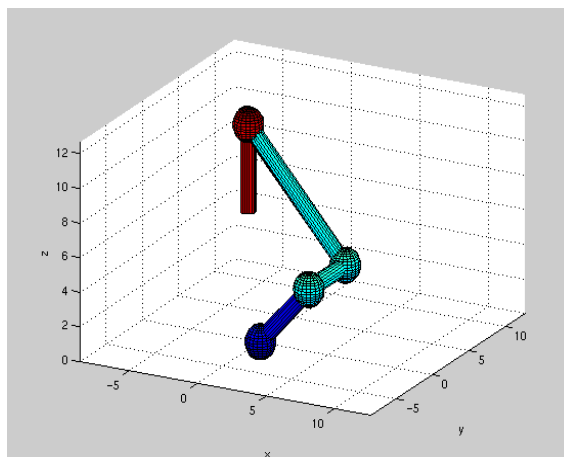
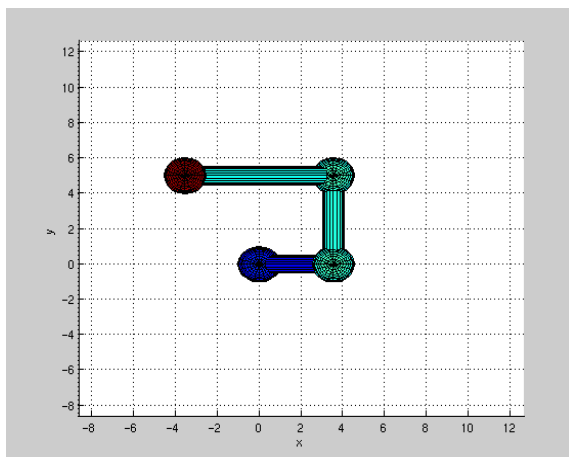
Den andra sektionen: Vi utgår återigen från en stående sektion placerad i origo (för att utföra rotationerna) och roterar denna 90 grader runt y-axeln. Detta ger att sektionen kommer att ligga längs med x-axeln. Efter detta tittar vi mot origo längs z-axeln och roterar sektionen 90 grader (motsols). Detta ger att sektionen ligger längs y-axeln. Sätter vi sen ihop delarna ser vi att vi får figuren ovan.

Exempel 2:

Antag att vi vill rita ut en robot som har 3 leder förutom den som sitter fast i golvet. De fyra delarna skall vara 5, 5, 10, respektive 5 enheter långa. Den första sektionen skall inte roteras kring z-axeln, men den skall roteras 45 grader runt y-axeln. Motsvarande vinklar för de tre återstående sektionerna är $(z=90, y=90)$, $(z=180, y=45)$ samt $(z=0, y=180)$. Vi har byggt på exempel 1:s robotarm med två sektioner till. Den vektor som skall skickas in till din utritningsfunktion skall då innehålla 12 data enligt följande anrop:

```
draw_robot_arm([0, 45, 5, 90, 90, 5, 180, 45, 10, 0, 180, 5]);
```

Detta anrop bör ge följande figur som resultat (sett uppifrån respektive som 3D-vy):



Uppgift 2 (ca 10 min)

Skriv funktionen `show_robot_arm_movements` som tar emot en matris bestående av ett antal rader med data, där varje rad beskriver hur robotarmen ser ut i ett givet ögonblick, och sen ritar ut robotarmen i varje position som beskrivs i matrisen. Det är viktigt att man kan se varje position (använd `pause`). Anropsexempel:

```
show_robot_arm_movements([0 0 5 0 0 5 ; -45 45 5 90 -90 5]);
```

Uppgift 3 (ca 1 tim 30 min)

Modifiera `show_robot_arm_movements` så att man får lite mjukare rörelse av robotarmen. Detta kan åstadkommas genom att man ritar ut armen i mellanpositioner till de rader som finns i den inkommande matrisen. T.ex. kan man låta vinklarna modifieras stegvis tills man nått nästa position.

2010-12-07