Mobile roboter

Elvi Khoguev

Universitetet for miljø- og biovitenskap

Institutt for matematiske realfag og teknologi, IMT

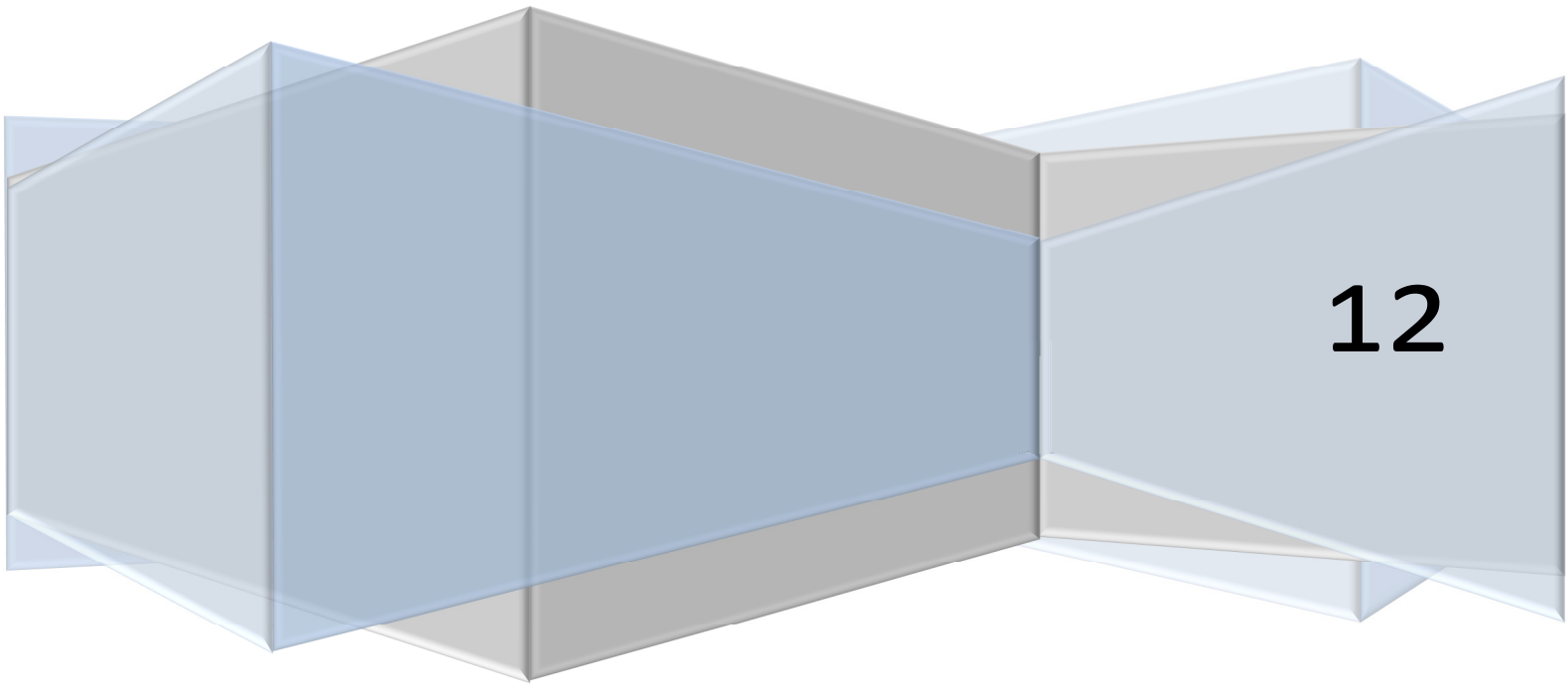
Masteroppgave 30 stp. 2012

Universitet for miljø- og biovitenskap, UMB
Institutt for matematiske realfag og teknologi, IMT

Mobile roboter

Masteroppgave

Elvi Khoguev



12

Masteroppgave av Elvi Khoguev

«Mobile roboter»

Innhold

Forord

Ideen om å skape robot-mekaniske enheter med menneskets utseende og handlinger eller noen levende skapninger, har fascinert menneskeheten siden tidenes morgen.

Selv i legender og myter har mennesker prøvde å skape et bilde av menneskeskapte vesener utstyrt med en fantastisk fysisk styrke og smidighet som i tillegg kan fly, leve under jorden og vann, handle selvstendig og samtidig adlyde mennesker og utføre det vanskeligste og farlige arbeidet. Selv i «Iliaden» av Homer (VI f.kr) sies det at lamme smeden Hefaistos, guden for ild og beskytter av smeder, smidde jenter av gull som utførte hans oppdrag.

Den første kunstige personen som har blitt nevnt i historien er en gigant laget av bronse, navngitte Talos, bygget av Hefaistos for å beskytte øya Kreta fra invasjon - dateres tilbake til 300 f.Kr. Det er mange legender; leire kolossen Golem som besatte enorm fysisk styrke og som ble en prototype av en robot. Ifølge Callistratus (300 f. Kr), ingeniøren og skulptøren Daedalus, skapte mange mekaniske skulpturer blant annet en statue av Afrodite som kan utføre ulike typer bevegelser; de var alle komplekse mekanismer.

Pålitelige data om mekaniske menn som var menneskelignende, tilhører og blir tilknyttet til navnet på den greske ingeniør Heron fra Alexandria (http://no.wikipedia.org/wiki/Heron_av_Alexandria#cite_note-1), som skapte flere mekaniske verker, særlig den berømte "Traktat om pneumatikk," der beskrev han et sett av bevegelige figurer og syngende fugler. I sitt arbeid på maskiner, skrev han at i antikken så var det noen som behersket kunsten å konstruere maskiner for templer til gudedyrkelse og teater. Ifølge Heron var det svært kompliserte hydrauliske og pneumatiske mekanismer.

Eolipile (Heronkula)- den første kjente varmekraftmaskin i historien.

Så blir den første historiske fasen av menneskehetens bevegelse mot opprettelsen av roboter preget av en overflod av myter og legender av mekaniske vesener, og etableringen av den første er ganske sofistikert for sin tid, menneskelignende maskiner - androids designet primært for religiøse og underholdning.

Blant de mest kjente oppfinnerne av mekaniske figurer på middelalderen var den franske ingeniøren Jacques de Vokanson (1709-1782). Han skapte en mekanisk and som kunne strakte halsen for å ta korn fra hånden, kavet i vannet og nøyaktig imiterte bevegelser til levende ender. Andre berømte modellen av Vokanson var «Pianist». «Pianisten» spilte piano og fløyte, løftete hode og simulerte puste.

Sveitsisk urmaker Pierre Jaquet Dro, (1721-1790) og hans sønn Henri Jacquet-Dro (1752-1791) nådde en høy perfeksjon i etableringen av maskiner – androids. På vegne av Henry Dro var etablert ord «android». Et eksempel på det høyeste tekniske ferdigheten kan være skapt av faren-Dro "Skriver". Han sitter ved et bord og med ryddig håndskrift skriver ned bokstaver og ord, sakte rister på hodet og senker øyelokkene i tid med bevegelse av hånden.

Skriver

«Skriver» består av 6000 detaljer.
I 1805 fransk oppfinneren Joseph Marie Jacquard (1752-1834) oppretter en automatisk maskin, som ved hjelp av kortstokk kan lage stoff med pre-programmert mønstre.

Programmerings metode ved hjelp av hullkort var deretter utviklet en engelsk matematiker, økonom og ingeniør Charles Babbage (1791-1871). I 37 år jobbet han for å oppfylle sin idé. I 1823, bygget han en differensiell maskin og startet arbeidet med en mer kompleks maskin. Den resulterende analytiske maskinen i sin strukturfunksjonene var nesten ferdigbygd datamaskin og hadde nesten alle de samme funksjonell blokkene som har moderne datamaskiner. Data innførtes ved hjelp av hullkort.

Selve begripelse «robot» kommer fra tsjekkiske ord «robota» som betyr tvunget arbeid og var brukt av tsjekkisk forfatter Karel Tsjapek (1890-1938) i hans skuespill « Rossum's Universal Robots» som handler om en person-Rossum. Han eide en fabrikk og ved en biologisk metode skapte roboter for å jobbe der.

Neste forfatteren som skrivet om roboter i sine vitenskapsdiktninger var Isaak Asimov. I 1942 prøvde han for første gang å formulere grunnleggende prinsipper for utførelse av roboter og deres samhandling med mennesker. Disse prinsippene er oppkalt etter robotikkens tre lover:

1. En robot tillates ikke å skade et menneske eller passivt la et menneske komme til skade.
2. En robot må følge ordre gitt av mennesker bortsett fra når slike ordre kommer i konflikt med første lov.
3. En robot må verne om sin egen eksistens såfremt slikt vern ikke kommer i konflikt med første og andre lov.

Grunnlegger og president i selskapet «Unimation" som startet første produksjon av industrielle roboter og gjenkjent som "faren til moderne industriell robotikk", Joseph Engelberger mente at robotikkens tre lover fra I. Azimov er standarder for å etterfølges av spesialister ved produsering av moderne roboter. Fantastisk ideer og bilder av forfattere i stor grad påvirket på utviklingen av vitenskapelige og teknologiske fremskritt, og de nye konsepter av "robot" har en viktig rolle, ikke bare i litteratur og kunst, men også i vitenskap, teknologi og produksjon.

Innledning

Utviklingen av det moderne samfunnet og arbeidslivet har ført til fremveksten og utviklingen av en ny klasse av maskiner - roboter - og tilsvarende forskningsområdet - robotikk. Robotikken er rask utviklede vitenskapelige og tekniske disiplin, studerer ikke bare teori, metoder for beregning og konstruksjon av roboter, systemer og komponenter, men må også løse problemene med komplekse automatisering av produksjon og forskning med bruk av roboter. Det bør bemerkes at begrepet "robotikk" brukes også i en annen sammenheng. Det betyr komplekser av maskiner og aggregater utstyrt med robot-enheter, eller komplekser som brukes i samband med roboter i én teknologisk prosess.

Vitenskapelig og industriell aktivitet er bestemt av to behovet: på den ene siden behovet for utvikling og forbedring av sosial produksjon som grunnlag for å oppnå økonomisk makt, på den andre - nødvendigheten av forskning og utvikling av nye utradisjonelle områder og aktiviteter. Historien om vitenskapelige og teknologiske prestasjoner er en perfekt illustrasjon

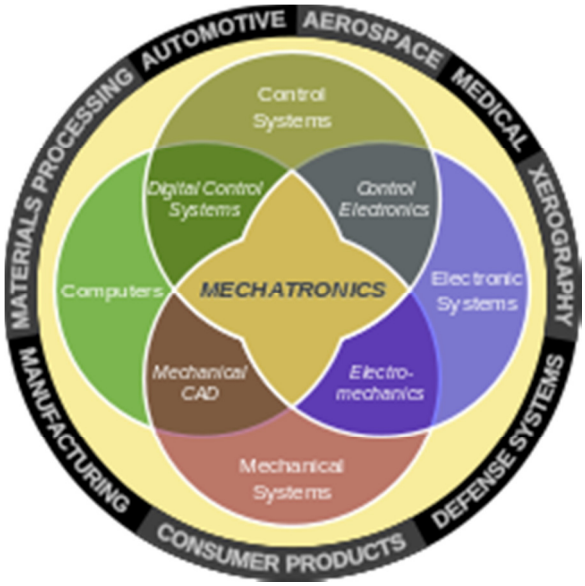
av de stadier i denne pågående prosessen. Fremveksten og utviklingen av roboter og robotikk - et levende eksempel på offentlig etterspørsel etter radikal økning i effektiviteten av produksjon og utvikling av nye områder og aktiviteter. En av de avgjørende faktorene i økonomisk utviklingen er intensivering av industriell produksjon som basert på komplekse mekanisering og automatisering.

Generelle prinsipper for robotikk

I nær fremtid ventes mer intensivt introduksjon av robotikk i produksjon. For å unngå feil i denne prosessen man må kritisk vurdere erfaringen og velge mest tilpassende metoder og teknologi. Så, professor L.I. Volchkevich anbefalte de generelle prinsippene for forskningspolitikk for industriell robotikk:
Første prinsipp - prinsippet om å oppnå resultater - sier at roboter skal ikke bare simulere eller erstatte mennesker, men gjøre menneskers funksjoner raskere, mer pålitelig og bedre. Bare i den tilfelle blir det virkelig effektivt.
Det andre prinsippet - prinsippet om komplekse løsninger - dikterer behovet for å håndtere de viktigste komponentene i en kompleks produksjonsprosess: produksjonsanlegg, teknologi, grunnleggende og tilleggsutstyr, styringssystemer og tjenester, bemanning, samarbeid med eksterne instanser og andre.
Tredje prinsipp - behovet - definerer bruk av roboter. Selve de mest moderne og avanserte roboter må ikke brukes hvor man kan tilpasse dem, men der hvor det er nødvendig.
Fjerde prinsipp - prinsippet om aktualitet - tillater ikke gjennomføring og realisering av umodne tekniske løsninger og design. Innføring av dyre, upålitelig og uproduktivt roboter og andre automatisert verktøy kan føre bare til å diskreditere dem.

Mekatronikk

En viktig del i utviklingen av robotikk som en fundamentalt ny tekniske retningen er effektiv forskning og prestasjoner. I de siste årene, dannet en ny, rask utviklede vitenskapelig og teknisk retning, kalt "Mekatronikk". Mekatronikk representerer en organisk sett av vitenskapelige ideer og prinsipper for mekanikk, elektronikk, datateknikk og reguleringsteknikk



Oversikts Venn-diagram fra nettsiden til Rensselaer Polytechnic Institute som beskriver de forskjellige disiplinene som utgjør faget mekatronikk.

Fremveksten og utviklingen av de grunnleggende konturene av denne disiplinen startet i Japan på grunn av den stadig voksende og svært produktiv bruk av maskiner av ulike elektroniske enheter i form av miniatyr elektroniske enheter, integrerte kretser og mikro dataenheter – mikroprosessorer. Formålet med Mekatronikk er ikke bare roboter som en bestemt enhet, men mekatroniske systemer som består av mekaniske og elektroniske enheter, hvor skjer utveksling av energi og informasjon. Mekatronikk innebærer også automatisering av planlegging og business management, industriell automasjon og robotikk, automatisert transport og lufttrafikken kontrollsystemer. Derfor, videreutvikling og forbedring av ny teknologi og organisasjonsformer for produksjon - fleksible produksjonssystemer med industrielle roboter - er direkte avhengig av prestasjoner av mekatronikk. Roboter har blitt en realitet av verdens økonomiske system, og det finnes ikke alternativer til deres bruk i industri og forskning. Roboter og robotsystemer tillater oss å undersøke og utforske områder som er utilgjengelige for mennesker. De var det manglende leddet som gjør mulig å knutte sammen høy ytelse med høy fleksibilitet av produksjonen. Derfor vil roboter og robotsystemer fortsette å vokse og være en stor del av industri, forskning og daglig menneskelig aktivitet. Ingen tvil om at de vil bli pålitelige assistenter av mennesker i forskning, utvikling og fremgang.

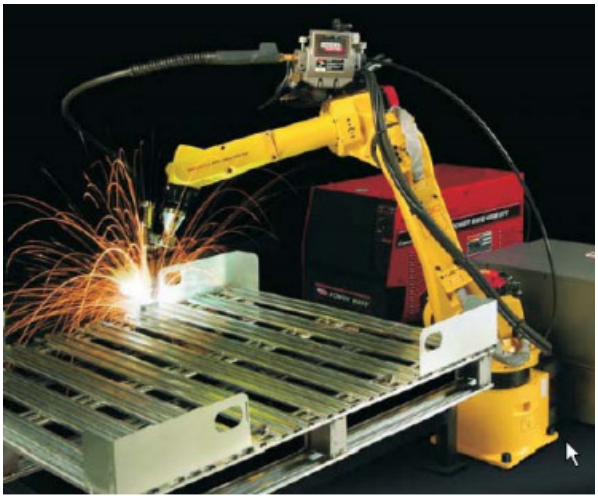
Klassifisering

Industrielle roboter – i sin tur er delt i tre grupper i henhold til produksjonsprosess; produksjons eller teknologiske roboter som utfører grunnleggende operasjoner, hjelperoboter som utfører transport, løfte-, flytte-, osv. operasjoner, universelle roboter som kan utføre begge funksjoner.

Eksempel:

Industriell robot FANUC Arc Mate 100iBe disse budsjettmessige roboter er utviklet for å maksimere effektiviteten av sveisings systemer. God balanse mellom størrelsen og arbeidsarealet, høy aksial hastighet er fordelen på design. Kontrolleren er integrert i base av

roboten for å forenkle installasjon og spare plass. All programvare for sveising gir god ytelse i enkle og sammensatte robotikk systemer.



Industriell robot FANUC Arc Mate 100iBe

Landsbrukts roboter – er utviklet for å automatisere tidkrevende og ensformig prosesser i jordbruksproduksjonen som tradisjonelt krever betydelige lønnskostnader. I tillegg til melking operasjonen, den mest automatiserte i de siste årene, blir det mulig å skape spesialtransport og teknologiske midler, som for eksempel traktorer som betjenes uten drivere, og brukt for såing, pløying, gjødsling, sprøyting avlinger, skjæring osv.

Eksempel:

Vision Robotics, et selskap basert i San Diego, har demonstrert en prototyp vinranke-beskjærings robot. God beskjæring krever dyktighet å balansere veksten av vintreet. Vinrankene må også trimmes på visse steder og på presise vinkler for å vokse de beste druene for vinlaging. Roboten er litt tregere enn en god menneskelig beskjærer, men det vil øke hastigheten. Den burde være i stand til å beskjære vinrankene på omtrent halvparten av kostnadene ved manuell arbeidskraft, sier Derek Morikawa, administrerende direktør i Vision Robotics (http://www.economist.com/node/15048711).

Landsbrukt robot

Transports roboter - designet for automatisert transportfasiliteter og styre ulike transportsystemer. Forskning og utvikling på bygging av transports roboter foregår intensivt over hele verden. Mens det er fire svært forskjellige typer - terrestriel, flyende, svømmende og underjordiske. Teori og praksis av den siste typen av roboter rekket ikke ennå stor

utviklingen. Største praktisk utvikling fikk transports roboter som bruker hjul, beltegående og vandrende. I stor industriell automatisert frakt, lagringssystemer, automatisert produksjon brukes hjul roboter i form av mobile kraner, automatiserte transport enheter utstyrt med forskjellige manipulatorer. Enkleste formen av slike roboter er skinnegående transportmidler, eller kan lages ruter ved hjelp av kabler som monteres under overflaten. Frekvens generator gjennom kabler oppretter et magnetisk felt og sensorer på transport roboten sender den til ønskede ruten. Slike enkle systemer kan inneholde flere grener og looper ved hjelp av forskjellige frekvenser for hver bane. Mer komplekse systemer kan ha roboter som er utstyrt med egen datasystem og sensorer.

Eksempel:

Autonomous Platform Demonstrator eller APD er en militær transport-robot som er utviklet av US Army Tank Automotive Research, Development and Engineering center - TARDEC. Dette er 9.6-tonn tungt og 4,5 meter langt UGV (unmanned ground vehicle) utformet for å vise siste teknologiske prestasjoner i dette feltet. Den har en hybrid elektriske drivverket med seks elektriske motorer som drives av li-ion-batterier som lades ved hjelp av en innebygd diesel generator. APD er skli-styrt kjøretøy som kan svinge på plass. Andre teknologier inkluderer et lett karosseri, en avansert bæresystem osv. Fra et kontrollpunkt den kan styres av en soldat, eller den kan operere autonomt. Autonomt den kan operere med hastigheter opptil 80 km/t. Den kan reise fra punkt til punkt ved hjelp av GPS og unngå hindringer i veien. Robotikk kjøretøyet kan overvinne 1 meter høy hindringer og navigere 60-graders bratte skråninger.

Autonomous Platform Demonstrator

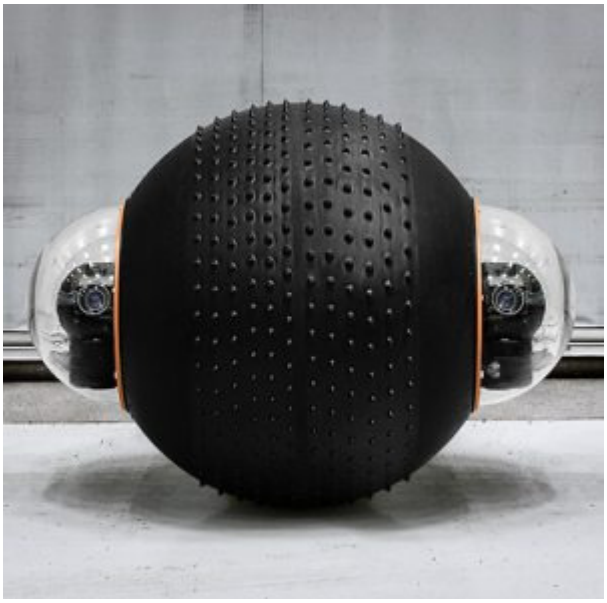
Spesielle roboter - kan brukes til å utføre ulike typer reparasjoner, restaureringer og redninger i ekstreme forhold og situasjoner, samt forebygging av ulykker, naturkatastrofer og fjerning av deres effekter. Utviklingen av slike roboter er rettet til å ta opp viktige saker av sikkerhet av mennesker og miljøet.

Virkeområdet for spesielle roboter er svært mangfoldig. De skapes til å fungere under ekstreme forhold for eksempel, kjernereaktorer, kjernefysiske ubåter og bedrifter, dekontaminasjon av lokaler, bygninger og områder av radioaktive, kjemiske, biologiske og andre utslipp, nøytralisering av ulike eksplosive innretninger; søk og redning av folk i ulykker,

krasjer og naturkatastrofer; kontrol over brann, kampen mot terrorisme og organisert kriminalitet, patrulje tjeneste og mye mer.

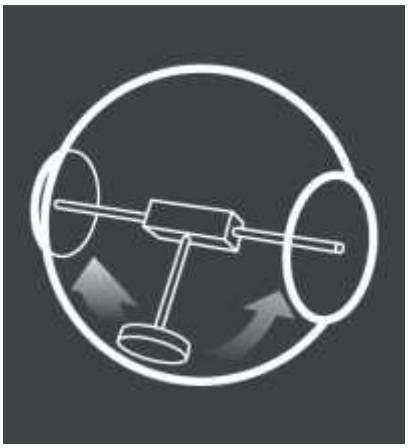
Eksempel 1:

GroundBot



GroundBot - © 2008 Rotundus

GroundBot er en selvstendig autonom Robot utviklet av Rotundus (www.rotundus.se). GroundBot har blitt utformet som en lukket enhet uten eksterne bevegelige deler. Istedenfor en intern gyroskop vekten brukes til å drive Robot fremover, som vist i den nederste diagrammet.



GroundBot er designet for bruk i tøffe miljøer, der tradisjonelle hjul eller beltegående roboter ikke kan fungere skikkelig. Disse miljøene er: etterforsket gasslekkasjer, interplanetariske ekspedisjoner og lokalisere jordskjelvet overlevende. Roboten kan operere i de fleste terreng inkludert dyp snø, is, gjørme og sand. På grunn av den forseglede utformingen av roboten (innkapsling IP67) er den også i stand til å flyte på vann. GroundBot er 60 cm. Høy og 80 cm. bred. Har 25 kg. vekt og kan jobbe i temperatur diapasonet fra – 30 til +40 ^{o}C. Hastighet opp til 10 km/t og har to kamera som gir 360 ^{o} synsfelt. Driftstid er 8-16 timer avhengig av

oppdrag. Robot har trådløs tilkobling med Wi-Fi bånd, har to GPS mottakere, akselerasjonsmåler, gyrosensor, magnetometer. To kameraer gir mulighet for høykvalitets video strømmen både i 2D og 3D.

Eksempel 2:

I et fengsel i Sør-Koreas byen Pohang, har myndighetene sammen med forskeres gruppe Asian Forum For Correction, som er spesialister på kriminalitet og fengsel politikk, startet et prosjekt. Tre roboter overvåker innsatte og vil bidra til å redusere arbeidsbelastning for andre vakter. De har 1,5 meters høyde, beveger seg på fire hjul og er utstyrt med kameraer og sensorer som tillater dem å oppdage risikofylt atferd som vold og selvmord. De kan varsle menneskelige vakter hvis de oppdager et problem.

Den fangevokter roboten prototype.

Roboter har spesielt operativsystem som tilpasset til arbeid inne i fengsel og kan være vennlige mot de insatte. (http://www.bbc.com/news/technology-15893772)

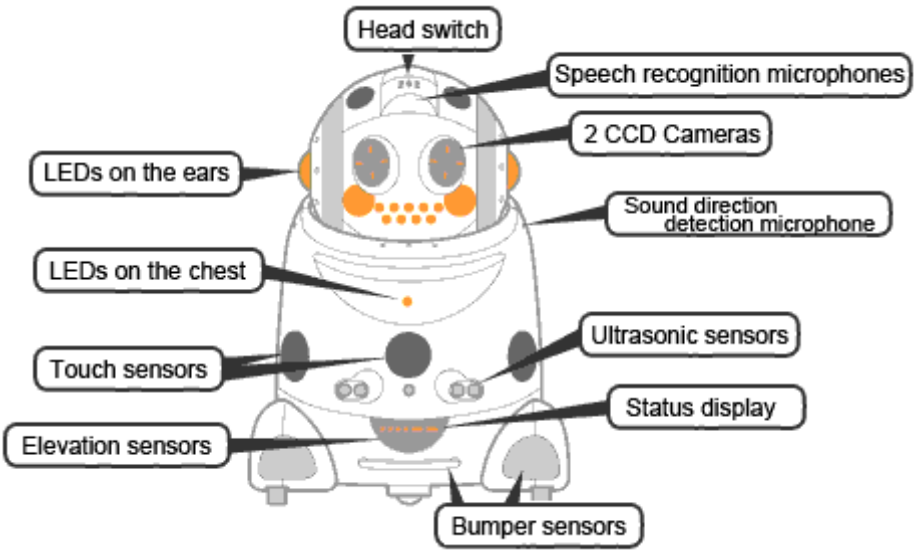
Service roboter - utformet for å automatisere forskjellige operasjoner i hjemmet og i service-sektoren. De har stor sosiale utfordring. De må forenkle hverdagsliv og gi mer tid til mennesker for kreativitet. Å skape service roboter er svært komplekse vitenskapelige og tekniske oppgaver. Det krever universelle fleksible systemer. Det vil si at de må ha nok intelligens for å være i stand til å utføre en rekke enkel arbeid - matlaging, oppvask, rengjøring, sy og reparasjon av klær, barnepass, opplærings ferdigheter.

Eksempel:

PaPeRo er en personlig Robot er utviklet av NEC Corporation (Japan). Den lille størrelsen og fargerik design av PaPeRo er ment å appellere til både barn og voksne, med sitt ansikt designet for å se både vennlige og rolige. Utvikling startet i 1997 med den første prototypen R100, og i 2001 navnet PaPeRo, som står for " Partner-type-Personal-Robot ble vedtatt.

PaPeRo roboter

Ved hjelp av talegjenkjenning teknologi så kan PaPeRo gjenkjenne menneskelig språk. I utgangspunktet forstår han ca. 200 ord, og det vil øke tilsvarende når nye funksjoner er tatt med. PaPeRo bruker nyeste talesyntese teknologi og den kan snakke med naturlig og søt stemme. Selv om omgivelsene er bråkete, lydene er for høyt eller for myk, kan PaPeRo kommunisere med brukeren.



Mekaniske strukturen av PaPeRo roboten

Med innebygd ansiktsgjenkjenning, kan PaPeRo lokalisere mennesker fra bildene på sin CCD-kamera ved hjelp av image gjenkjennelse teknologien. Det ansikts identifikasjon systemet kan skille mellom og huske ansiktene til opptil 30 personer. Roboten kan samhandle med en person og se på ansiktet hans. Når han ikke kan se et ansikt godt eller

hvis avstanden eller sted er ikke tilfredsstillende, justerer han avstanden selv eller sier til personen, "Det er for mørkt".

Ni berørings-sensorer er montert på hodet og kroppen av roboten for å oppnå ulike samspill med barn fra et bredt spekter av aldre, inkludert spedbarn. For eksempel, PaPeRo blir glad når du klapper hodet og PaPeRo kiler når du berører buken.

Når PaPeRo ikke snakker med folk den engasjerer seg i autonome aktiviteter. Han går rundt i rommet med vilje eller danser på egen hånd. Ved bruk av sine sensorer, for eksempel kameraer og ultralyd sensorer, kan PaPeRo gå uten å kollidere med gjenstander på gulvet.

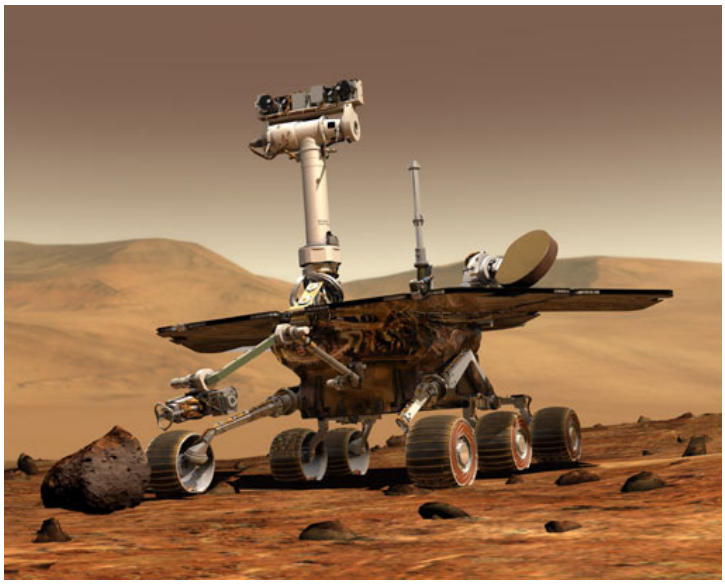
Når PaPeRo batteri nivået synker, ser han for sin ladestasjon og går til ladning selv. Når batteriet er ladet, kommer PaPeRo ut fra ladestasjonen.

Designer roboter - utformet for automatisk beregning og design av maskiner og strukturer, utviklingen av teknologiske prosesser, systemadministrasjon, informasjons management, etc. Nå finnes de bare i form av separate systemer, fortsatt langt fra perfekt, men raskt voksende. Problemet med robotisert design (design ved hjelp av roboter) er spesielt relevant i forhold til utvikling og etableringen av fleksible produksjonssystemer som består av flere robotikk moduler og inneholder så mange variabler at mennesker-designer kan ikke koble dem sammen. Et ideelt verktøy for å løse slike oppgaver er datasimulering.

Forsknings roboter – det er roboter som brukes til søk, innsamling, forberedelse og overføring av undersøkte objektene. Slike objekter kan være vanskelig å få tak i, eller å være utilgjengelig for mennesker; ute i verdensrommet, dypt i havet, under jorden, under ekstrem forhold i laboratorium. De kan brukes i områder som krever identifikasjon, behandling og analysering av store mengder informasjon, for eksempel informasjonshenting og leting, kunst og litteratur.

Eksempel1:

Spirit, er den første av to rovere i forbindelse med NASAs Mars Exploration Rover-program. Den 4. januar 2004 landet NASA Spirit i Gusev-krateret på Mars, tre uker senere landet tvilling-roveren Opportunity på den andre siden av Mars. Før landing var målet for hver av Mars-roverne å kjøre opptil 40 meter på en dag, totalt opp til 1 kilometer. Begge roverne har passert disse målene. Roveren fortsatte å fungere effektivt over 10 ganger lengre enn det NASA-planleggere ventet, noe som har tillatt en å gjennomføre store geologiske undersøkelser av stein og overflate-trekk på Mars-overflaten.
http://no.wikipedia.org/wiki/Spirit_(Mars-rover)



Spirit (Mars – rover)

Rovere var utstyrt ved veldig avansert vitenskapelig instrumenter:

«Panoramic Camera» (Pancam): for å bestemme mineralogien, overflaten og struktur på det lokale terrenget.

«Miniature Thermal Emission Spectrometer» (Mini-TES): for å identifisere mulige stein og steder en ønsker å undersøke nærmere og for å bestemme prosessene som formet Mars-steinene. Instrumentet vil også se mot himmelen for å lage temperatur-profiler på Mars sin atmosfære.

«Mössbauer Spectrometer» (MB): for nærmere undersøkelser av mineralogien til steiner og jordprøver.

«Alpha Particle X-Ray Spectrometer» (APXS): for nærmere analyser av elementene i steiner og jordprøver.

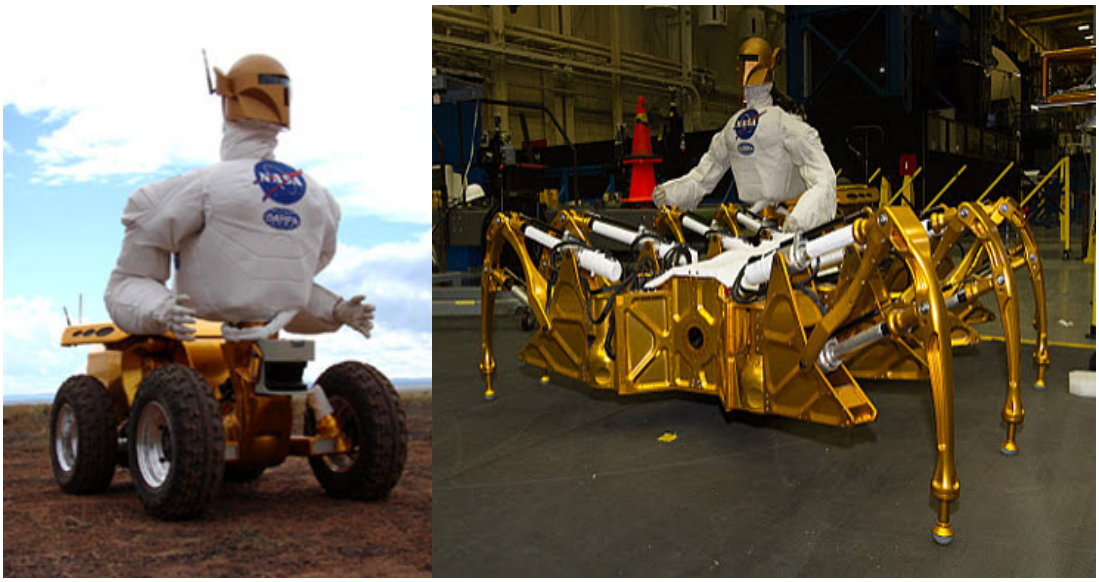
Magneter: for å samle magnetisk støv-partikler. MB og APXS vil analysere prøvene.

«Microscopic Imager» (MI): for å ta nærbilder med høy resolusjon (oppløsning) av steiner og jord.

«Rock Abrasion Tool» (RAT): for å bore i steiner.

Eksempel 2:

Et godt eksempel på forsknings robot er andre prosjekt – Robonaut. Det er en humanoid robot konstruert og bygget ved NASA Johnson Space Center i Houston, Texas. Det er maskinen som kan hjelpe mennesker arbeide og utforske i verdensrommet. Det som er unikt med den roboten er manipulatorer som helt like med menneskenes hender og det gir mulighet til å bruke samme arbeidsområde og verktøy med bedre effektiviteten og fjerner behovet for spesialiserte robot kontakter.



Robonaut på en hjul- og på en gående mobile baser.

Kontrollmetoder av roboter

Fjernkontrollstyring: I denne kontrollmetoden, kontrollerer menneskelig operatør en robot fra avstand som er for stor for operatøren for å se rett hva roboten gjør. Informasjonen om robots handling kommer fra robotens sensorer til styringssystemet. Stort sett sitter den menneskelige operatør på en arbeidsstasjon og regisserer en robot gjennom en slags grensesnitt. Roboter som ble lagd i NASA og sendt til Mars er et eksempel på fjernstyrte roboter.

Fjernstyrte mobil robot «Daksh» designet for nøytralisering av sprenglegemet (Indisk hær).

Fjernkontrollstyring har noen ulemper fordi mange oppgaver er repeterende og kjedelig. Det kan også hende at kameraet ikke kan overføre nye bilder svært raskt fordi kommunikasjonen har begrenset båndbredde. På grunn av disse og mange andre problemer som fører til tidsforsinkelse trenger fjernkontrollstyring utvikling av andre kontrollmetoder.

Teleprence - det som populært kalles virtuell virkelighet, der operatøren har full sensors tilbakemeldinger og føler som føler roboten. Dersom operatøren snur seg i en bestemt retning, roboten gjør det samme. Dersom operatøren presser på en joystick, robot beveger seg fremover. Hvis hjulene glir, vil operatøren høre og føle at motorene strammer seg og samtidig se at det var ingen visuell endring. Dette gir veldig naturlig grensesnitt til menneske, men det er veldig dyrt utstyr og krever svært stor båndbredde. Det krever også en person per robot. Dette er bedre enn tradisjonell fjernkontrollstyring, men gir ikke mulighet til en operatør å ha én kontroll over flere roboter.

Autonomt kontroll - kontroll av datamaskin, med sensors tilbakemeldinger, uten menneskelig inngrip. I denne type kontroll, kan datamaskinen sende roboten til forhåndsprogrammerte posisjoner og manipulere hastigheten og retningen av roboten som er basert på sensors tilbakemeldinger. Datamaskinen kan også kommunisere med andre enheter for å hjelpe roboten gjennomføre sine oppgaver.

Semi-autonomt kontroll - det kalles ofte overvåkende kontroll, hvor fjernkontrollen for en instruks eller en del av en oppgave som den kan trygt gjøre på egen hånd.

Offline programmering - bruk av dataprogrammer med realistisk grafikk for plan og programmere bevegelser uten bruk av robots maskinvare.

Kjør-gjennom programmering (Lead-Through Programmering). Poboten settes i «lære-modus» og en menneskelig operatør viser roboten nøyaktig hva slags bevegelser må gjøres for en oppgave, mens datamaskinen husker bevegelser (lagring av felles posisjoner, lengder og / eller vinkler, for å bli avspilt under utførelsen av oppgave).

Lær programmering(Teach Programmering) – robot flyttes til nødvendige for oppgaven stillinger ved en lærer anheng (teach pendant) datamaskin husker disse konfigurasjonene, og spiller dem tilbake i robots bevegelses rekkefølge. Ved hjelp av en lærer anheng, kan en arbeide med en robot uten å være bundet til en fast terminal. Disse enhetene brukes til å styre roboter i et bredt utvalg av innstillinger, og av mennesker som forsker og utvikler roboter. Den lærer anheng er en kontroll boks som den menneskelige operatøren bruker til å plassere roboten på sted ved å manipulere knappene på boksen. Flere funksjoner er inkludert på en typisk lærer anheng. Enheten har vanligvis en nødstopp-knapp, slik at operasjoner kan straks slå av operasjonen hvis det er et problem, som kan oppstå når en robot ser ut til å være ødelagt. Det er også en skjerm som kan brukes til å se og redigere kommandoer, og for å se gjennom historien til kommandoer som gitt til roboten, sammen med et tastatur for kommando innspilling. Den type tastatur kan variere, avhengig av hvilken type roboten lære anheng er laget for å styre

Fanuc RJ Series Spot Welding Teach Pendant.

Hva er robot ?

For å forstå hva en robot er må vi starte fra begynnelsen.

En robot er et kompleks system som består av flere delsystemer. Det er en kombinasjon av elektronikk, mekanikk og programvare.
Enkleste robotikk system vist på figur 1. Kontroller sender et signal til roboten. Signal gjennom en aktuator går til mekanisme som ferdiggjør kommandoen og resultatet av handlingen kommer tilbake til kontroller.

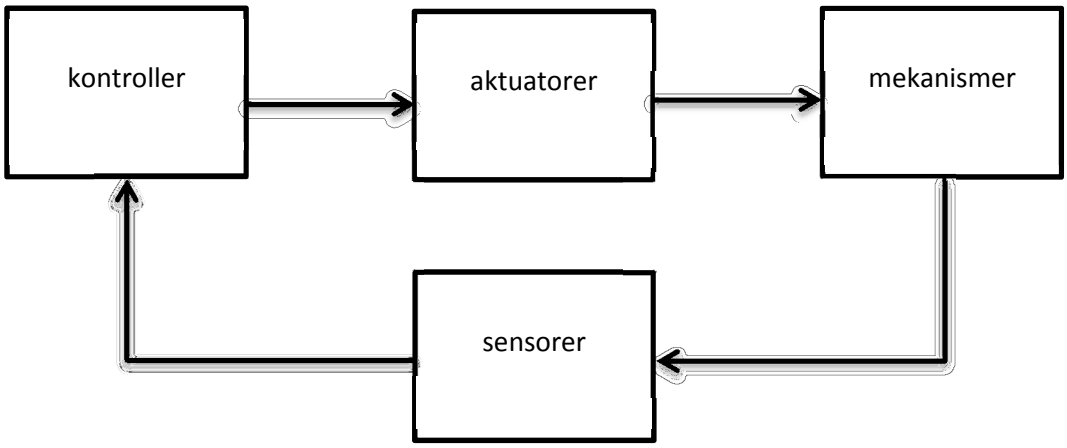


Fig.1.1. Komponenter av robotikk system.

Aktuator- det er en mekanisk innretning for å bevege eller kontrollere en mekanisme eller en prosess. Typisk mottar den mekaniske innretningen energi i form av luft, elektrisitet eller væske, og konverterer dette til en bevegelse. Eksempelvis kan en motor betegnes som aktuator når sirkulære bevegelser er nødvendig. Lineære motorer gir rette bevegelser.

Sensor er et instrument som registrerer en viss påvirkning, f.eks. lys, lyd varme eller kulde, og konverterer registreringen til et signal som kan leses av en observatør eller et annet instrument. Fotocelle er et eksempel på en sensor som registrerer bevegelse eller berøring og fra det kan en trigger eller en detektor utløse en alarm eller et elektronisk varsel.

Mekanisme - det kan være veldig bredt utvalg av verktøy, manipulatorer, drivende hjuler osv.

Det sentrale trekk ved en robot er dens mekaniske struktur. Roboter kan bli klassifisert som de med en fast base, robot manipulatorer, og de med en mobil base, mobile roboter. I det følgende er de geometriske egenskapene til de to klassene er presentert.

Robot manipulator

Den mekaniske strukturen av en robot manipulator består av en sekvens av stive, sammenkoplete legemer(ledd); en manipulator - en *arm* som sikrer mobilitet, et *håndledd* som gir behendighet, og en *slutt-effektor* som utfører oppgaven som kreves av roboten.

Robot manipulator

Den grunnleggende strukturen i en manipulator er serie- eller åpent kinematisk kjede. En kinematisk kjede betegnes som åpent når det er bare en sekvens av lenker som kobler de to endene av kjeden. Alternativt en manipulator inneholder en lukket kinematisk kjeden når en sekvens av linker danner en loop.

En manipulators bevegelighet bestemmes av antall ledder. Artikulasjonen mellom de to tilkoblete leddene kan realiseres ved hjelp av enten en prismatisk eller en roterende tilkobling. I en åpen kinematisk kjede, hver prismatisk eller roterende tilkobling gir strukturen med en enkelt grad av frihet (DOF). Prismatisk tilkobling skaper en relativt translasjonell bevegelse mellom de to leddene, mens en roterende tilkobling skaper en relativ

rotasjonsbevegelse mellom de to koblinger. Roterende tilkoblinger blir vanligvis foretrukket fremfor prismatiske leddene på grunn av deres kompakthet og pålitelighet. På den annen side, i en lukket kinematisk kjede, antall frihetsgrader er mindre enn antall ledd på grunn av de begrensninger pålagt av loopen.

De frihetsgradene skal være riktig fordelt langs den mekaniske strukturen for å ha et tilstrekkelig antall til å utføre en gitt oppgave. I det mest generelle tilfellet en oppgave som består av vilkårlig posisjonering og orientering av ett objekt i tredimensjonale (3D) plass, er seks frihetsgrader nødvendig, tre for å posisjonere et punkt på objektet og tre for å orientere objektet med hensyn til en gitt koordinatsystemet. Hvis flere frihetsgrader enn nødvendig for å fullføre en oppgave er tilgjengelig, sies det at manipulator er overflødig fra et kinematisk utsiktspunkt.

Et arbeidsområde for en manipulator representerer det område hvor manipulators end-effektor kan få tilgang til. Dens form og volum avhenger av manipulators struktur samt tilstedeværelse av mekaniske felles grenser. Armen må posisjonere håndleddet sånn at slutt-effektor kan få en riktig orientering. Typen og rekkefølgen i armens frihetsgrader, som starter fra felles basen, gir en klassifisering av manipulatorer som kartesisk, sylindrisk, sfærisk, SCARA, og antropomorfisk.

Kartesisk geometri realiseres av tre prismatiske ledder som akser, og de er gjensidig ortogonale (Fig. 1.2). Hver DOF tilsvarer en kartesiansk plass variabel og dermed er det naturlig å utføre rette bevegelser i rommet. Den kartesiske struktur gir meget god mekanisk stivhet. Håndleddets posisjoneringsnøyaktighet er konstant overalt i arbeidsområdet. Dette er volumet omgitt av et rektangulært prisme.

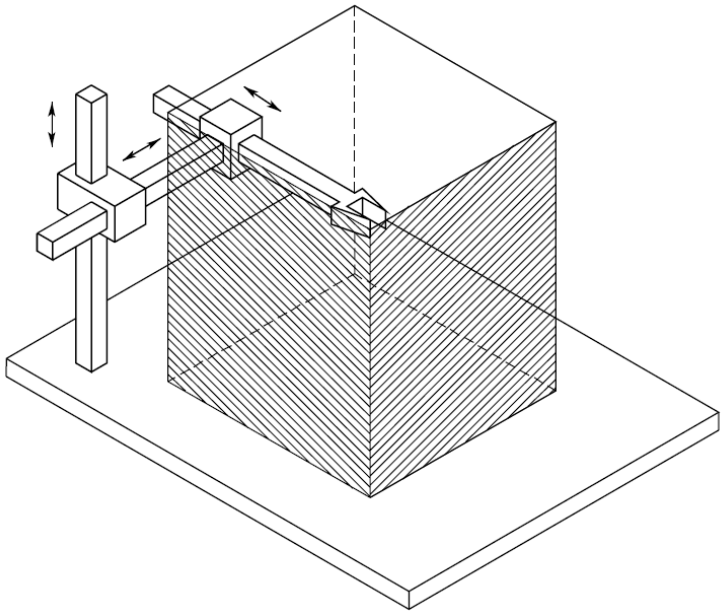


Fig. 1.2. Kartesisk manipulator og dens arbeidsområde.

I motsetning til høy nøyaktighet, har strukturen lav behendighet siden alle leddene er prismatiske. Retningen for å manipulere et objekt er fra siden.

På den annen side, hvis det er ønskelig å manipulere et objekt fra toppen, kan den kartesiske manipulatoren realisere dett ved en *gantry* struktur som illustrert i fig. 1.3. En slik struktur gjør tilgjengelig en arbeidsplass med stort volum og gjør det mulig manipulering av gjenstander ved store dimensjoner og tung vekt. Kartesiske manipulatorer er ansatt for materialhåndtering og montering. Motorene som drever leddene i en kartesiansk manipulator er vanligvis elektriske og tidvis pneumatiske.

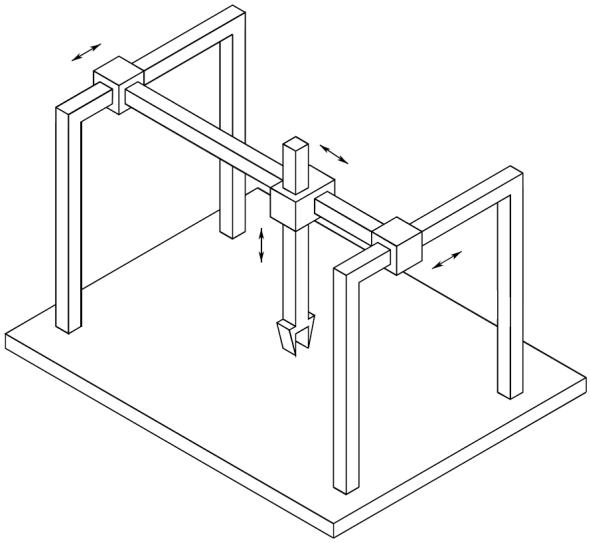


Fig. 1.3. Gantry manipulator

Sylindrisk geometri er forskjellig fra kartesiansk i at den første prismatisk tilkobling er erstattet med en revolute(roterende) tilkobling (Fig. 1.4). Hvis oppgaven er beskrevet i sylindriske koordinater, i dette tilfellet hvert DOF svarer også til en kartesiansk plass variabel. Den sylindriske struktur gir god mekanisk stivhet. Posisjonerings nøyaktighet for håndledd reduseres når den horisontale avstanden fra støtteaksen øker. Arbeidsområdet er en del av en sylinder (Fig. 1.4). Den horisontale prismatiske tilkobling gjør håndleddet på en sylindrisk manipulator egnet til å få tilgang til horisontale hulrom. Sylindriske manipulatorer er i hovedsak ansatt for å bære gjenstander ved store dimensjoner, i et slikt tilfelle bruk av hydrauliske motorer foretrekkes enn bruk av elektriske motorer.

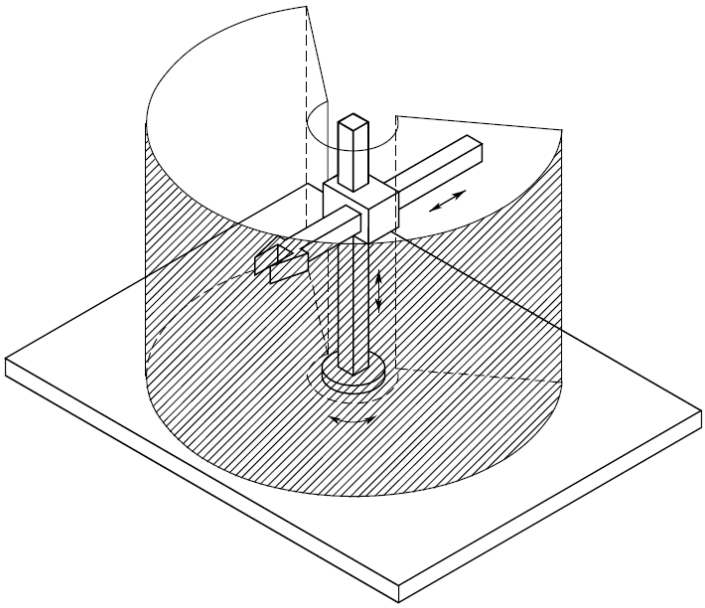


Fig 1.4. Sylindrisk manipulator og dens arbeidsområde.

Sfærisk geometri er forskjellig fra sylindrisk ved at andre prismatisk tilkobling er erstattet med en *revolute*(roterende) tilkobling (Fig. 1.5). Hver DOF tilsvarer en kartesisk plass variabel. Forutsatt at oppgaven er beskrevet i sfæriske koordinater. Mekaniske stivheten er lavere enn de to ovennevnte geometrier og mekaniske konstruksjonen er mer komplisert. Håndleddets posisjoneringsnøyaktighet minker, når radial avstand øker. Arbeidsområdet er en del av en sfære(Fig. 1.5). Det kan også inkludere bunnen av manipulators støtte og det kan manipulere gjenstander på gulvet. Sfæriske manipulatorer er hovedsakelig brukt for maskineri. Vanligvis brukes elektriske motorer.

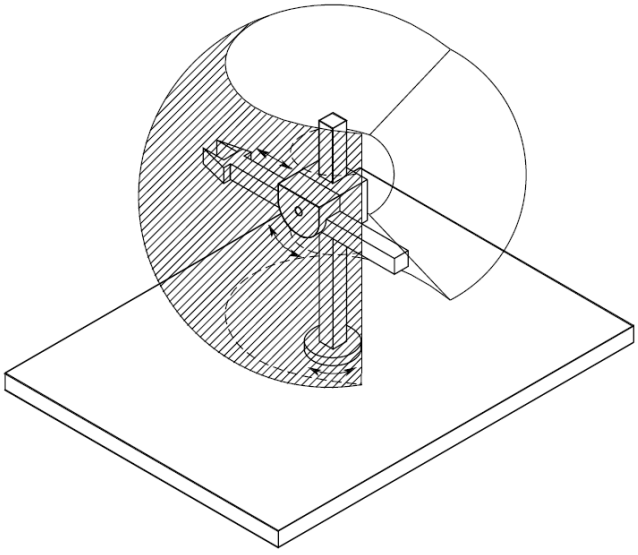


Fig. 1.5. Sfærisk manipulator og dens arbeidsområde.

En spesiell geometri er SCARA geometri som kan realiseres ved avhending to ledd med roterende tilkoblinger og et ledd med prismatiske tilkoblinger på en slik måte at alle aksene av bevegelser er parallelt (Fig. 1.6).

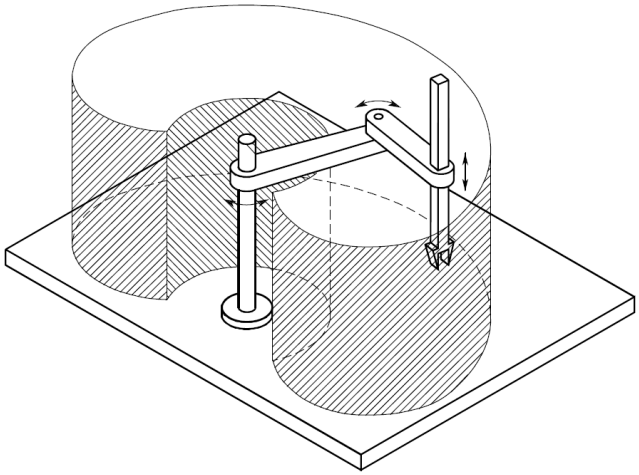


Fig. 1.6. SCARA manipulator og dens arbeidsområde.

Forkortelsen SCARA står for *Selective Compliance Assembly Robot Arm* (selektiv samsvar montert robots arm) og karakteriserer de mekaniske funksjonene av en struktur som tilbyr høy stivhet til vertikale laster og overholdelse til horisontale belastninger. Som sådan er den SCARA struktur velegnet til vertikale monterings oppgaver. Håndleddets posisjoneringsnøyaktighet minker når øker avstanden til håndleddet fra den første felles aksen øker. Det typiske arbeidsområdet er illustrert på Fig 1.6. SCARA manipulator brukes for små objekter og drives av elektriske motorer.

Antropomorfisk geometri er realisert ved tre roterende ledder; den roterende aksen i første leddet er ortogonal til aksene i de andre to leddene som er parallelt (Fig. 1.7).

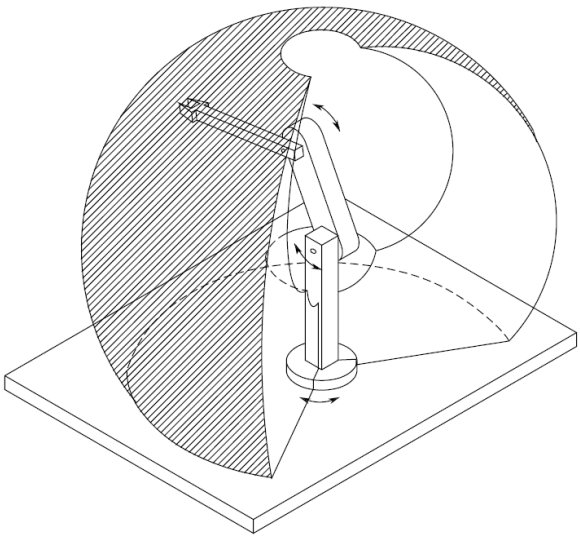


Fig. 1.7. Antropomorfisk manipulator og dens arbeidsområde

I likhet med en menneskelig arm, det andre leddet kalles skulderleddet og det tredje leddet albueleddet siden det knytter "arm" med "underarmen." Den antropomorfe strukturen er den mest behendige, siden alle leddene er roterende. På den andre siden, korrespondansen mellom frihetsgrader og de kartesiske plassvariabler er tapt, og håndleddets posisjoneringsnøyaktighet varierer innenfor arbeidsområdet. Dette er en del av en sfære (Fig. 1.7). Leddene er vanligvis drevet av elektriske motorer. Utvalget av industrielle anvendelser av antropomorfe manipulatorer er veldig bredt.

Alle de tidligere manipulatorer har en åpen kinematisk kjede. Når større nyttelast er nødvendig, vil den mekaniske strukturen ha høyere stivhet for å garantere nødvendige posisjoneringsnøyaktighet. I et slikt tilfelle, anbefales en lukket kinematisk kjede. For eksempel, for en antropomorf struktur kan vedtas parallellogram geometri mellom skulder og albue ledd, slik som å lage en lukket kinematisk kjede (Fig. 1.8).

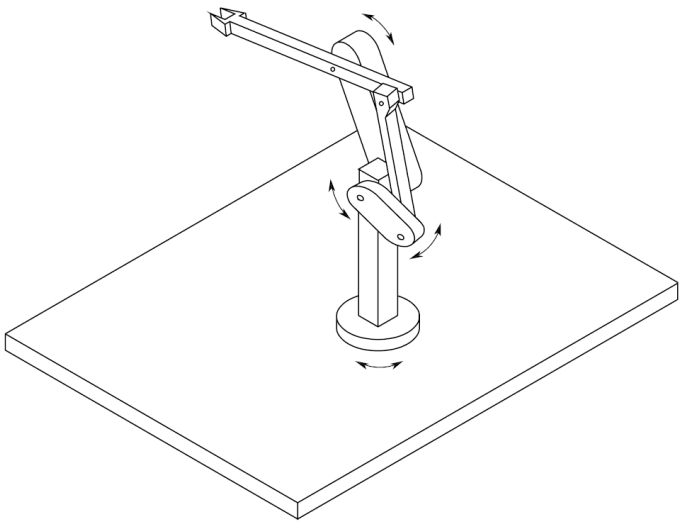


Fig 1.8. Manipulator med parallellogram geometri.

En interessant lukket kjede geometri er parallell geometri (Fig. 1.9) som har flere kinematiske kjeder som forbinder base og slutt-effektor. Den fundamentale fordelen er høy stivhetsgrad, i forhold til åpen-kjeden manipulatorer, og dermed muligheten til å oppnå høy operasjonelle hastigheter. Ulempen er en redusert arbeidsområde.

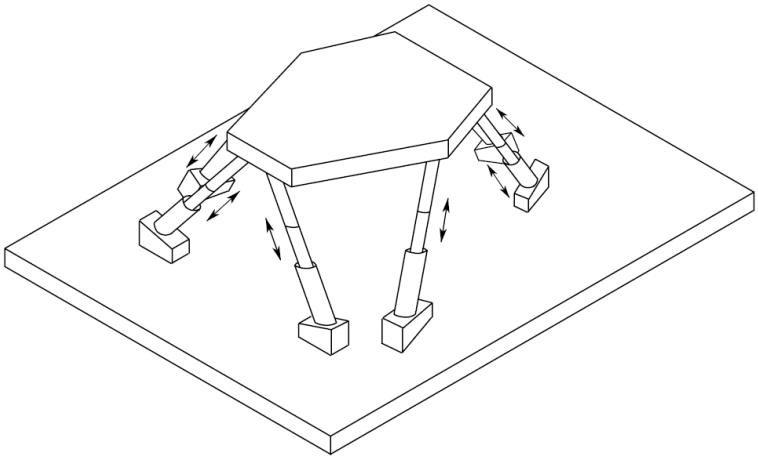


Fig. 1.9. Parallell manipulator

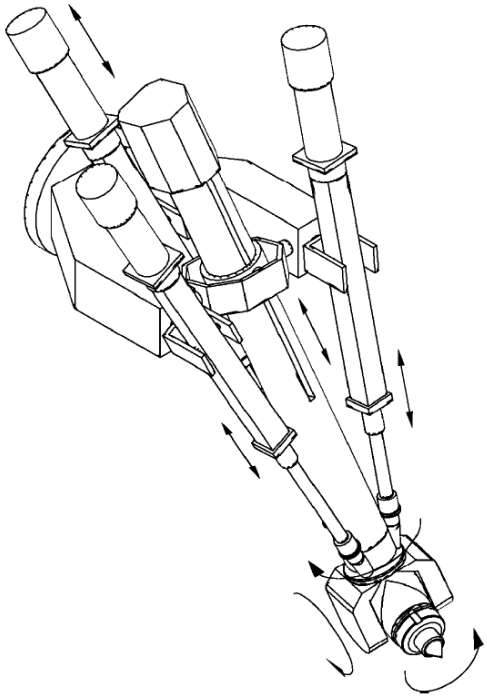


Fig. 1.10. Hybrid parallell-serie manipulator.

Geometrien illustrert i Fig. 1.10 er av hybrid type, siden den består av en parallell arm og en serie kinematisk kjede. Denne strukturen er egnet for utførelsen av manipulasjons oppgaver som krever store verdier i kraft langs vertikal retning.

Manipulatorens strukturer presentert ovenfor er nødvendig for å posisjonere håndleddet for å orientere manipulators slutt-effektor på nødvendige mote. Dersom vilkårlig orientering i 3D-rom er ønskelig, må håndleddet ha minst tre frihetsgrader som tilbys av roterende ledd. Siden håndleddet er siste delen av en manipulator, må den være kompakt. Dette ofte kompliserer mekanismens design. Uten å gå inn konstruksjonsdetaljer, for å ha høyeste ferdighet må håndleddets tre roterende akser krysse hverandre i ett punkt. I et slikt tilfelle, blir håndleddet til et sfærisk håndledd, representert i fig. 1.11. Det sentrale trekk ved sfærisk håndleddet er dekobling mellom posisjon og orientering om end-effektor, armen har oppgaven med å posisjonere over skjæringspunktet, mens håndleddet bestemmer end-effektors orientering. Disse erkjennelser der håndleddet er ikke sfærisk er enklere fra et mekanisk synspunkt, men stilling og orientering er koblet, og dette kompliserer samordningen mellom bevegelse av armen og håndleddet for å utføre en gitt oppgave.

Slutt-effektor er spesifisert i henhold til oppgaven roboten skal utføre. For materialhåndterings oppgaver består end-effektor av et griperområde av riktig form og dimensjoner som bestemmes av objektet som skal gripes (Fig. 1.11). For maskinerings og monterings oppgaver, er slutt-effektor et verktøy eller en spesialisert enhet, f.eks. en sveisepistol, en spray pistol, en mølle, en drill, eller en skrutrekker.

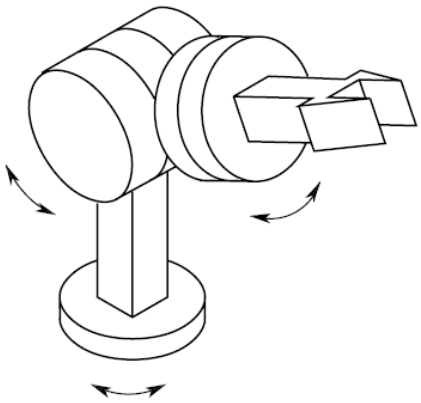


Fig. 1.11. Sfærisk håndledd

Allsidigheten og fleksibiliteten til en robot manipulator skal ikke fremkalle overbevisningen om at alle mekaniske strukturer er tilsvarende for å gjennomføre av en gitt oppgave. Valget av en robot er faktisk bestemt av kravet som setter begrensninger på arbeidsområdets dimensjoner og form, maksimalt nyttelast, posisjoneringsnøyaktighet, og dynamisk ytelse av manipulator.

Fig.1.12. Forskjellige typer av slutt-effektor.

Mobile roboter

Hovedtrekket av mobile roboter er tilstedeværelsen av en mobil base som lar roboten bevege seg fritt i miljøet. I motsetning til manipulatorer, slike roboter er mest brukt i tjenesteytende applikasjoner, hvor omfattende, autonome bevegelsesevner er nødvendige. Fra et mekanisk synspunkt, en mobil robot består av ett eller flere stive legemer utstyrt med et mobilt system.

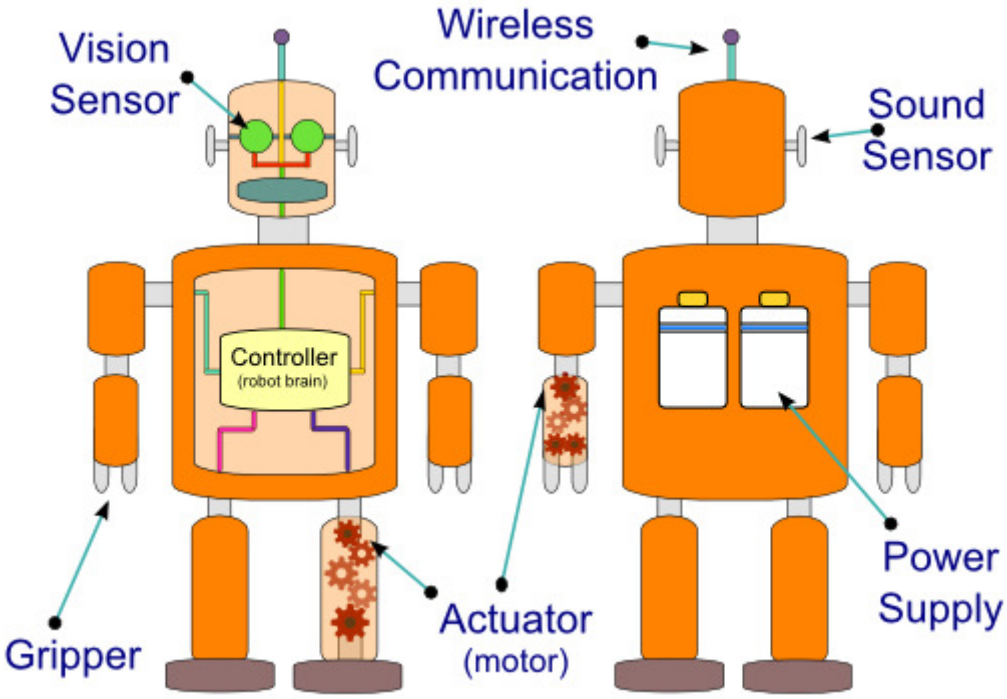


Fig 1.13. Mekaniske struktur av mobile robot

Mobile roboter må kunne bevege seg frem og tilbake, snu til høyre og venstre. Det er en fordel hvis robot kan snu seg på plass, fordi ofte de må jobbe på begrensete områder.

Mobile roboter kan deles på to hovedklasser:

Hjulgående mobile roboter vanligvis består av en stiv kropp (base eller chassis) og et system av hjul som gir bevegelse i forhold til bakken. Andre stive legemer (trailere), også utstyrt med hjul, kan kobles til basen ved hjelp av de roterende leddene.

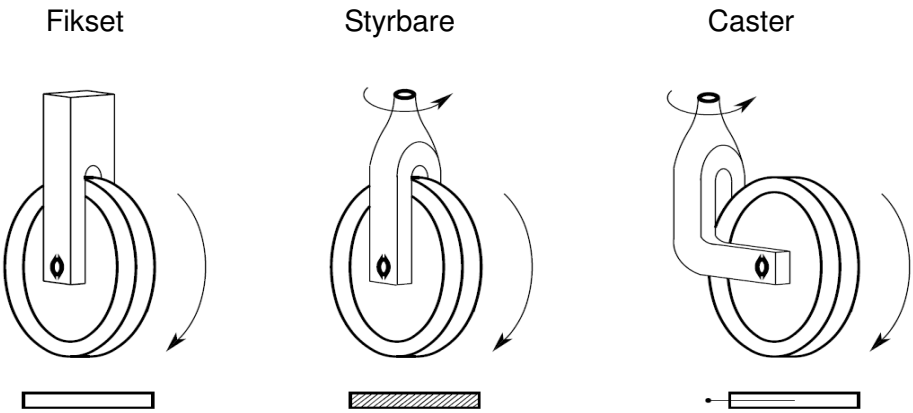


Fig 1.14. Tre typer konvensjonelle hjul med sine respektive ikoner.

Legged mobile roboter er laget av flere stive kropper, sammensatte ved prismatiske tilkoblinger eller, oftere, ved roterende tilkoblinger. Noen av disse legemer danner underekstremitetene og de med jevne mellomrom kommer i kontakt med bakken for å realisere bevegelse. Det er et stort utvalg av mekaniske strukturer i denne klassen. Design er ofte inspirert av studiet av levende organismer (*biomimetic robotics*); fra to føtters humanoider til seksføtters roboter som tar sikte på å kopiere de biomekaniske effektiviteten fra insekter.

Fig. 1.15. KONDO Multi Legged ROBOT "KMR-M6" PV1

Hjulgående kjøretøy er de mobile roboter som faktisk brukes aller fleste. Det grunnleggende mekaniske elementet av slike roboter er hjulet. Tre typer konvensjonelle hjul finnes, som vist i fig. 1.14 sammen med ikonene som vil bli brukt til å representere dem:

• Den faste hjulet kan rotere om en akse som går gjennom sentrum av hjulet og er ortogonal til hjulets bevegelsesretning. Hjulet er strengt festet til understellet, og derfor hjulets orientering er konstant.

• Det styrbare hjulet har to akser i rotasjon. Den første er den samme som et fast hjul, mens den andre er vertikal og går gjennom sentrum av hjul. Dette gjør at hjulet kan endre sin orientering i forhold til chassis.

• Den caster hjulet har to økser av rotasjon, men den vertikale aksen ikke går gjennom sentrum av hjulet. En slik ordning fører til at hjulet dreies automatisk, etter bevegelsesretningen av kabinettet. Denne typen hjulet er derfor innført for å gi et støttepunkt for statisk balanse uten at det påvirker mobilitet av basen, for eksempel, caster hjul blir ofte brukt i handlekurvene.

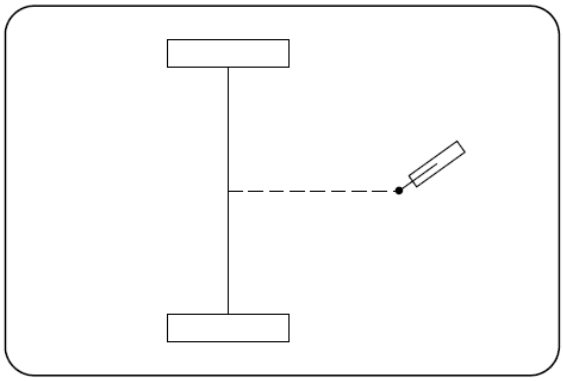


Fig. 1.16. Differensiell-drivende mobile robot.

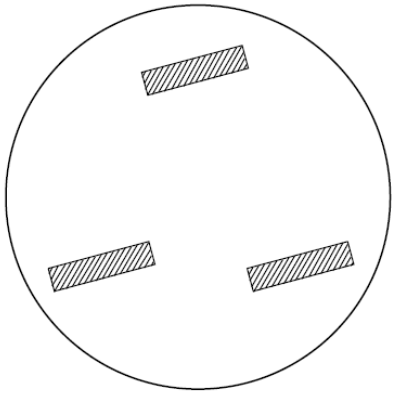


Fig. 1.17. Synkro-drivende mobile robot.

Variasjoner av kinematiske strukturer som kan oppnås ved å kombinere de tre vanlige hjulene er store. I det følgende de mest relevante ordninger er kort undersøkt.

I en differensial-kjøretøy er det to faste hjul med en felles rotasjonsaksen, og ett eller flere caster hjul, som er vanligvis mindre og deres funksjon er å holde roboten statisk balansert. (Fig. 1.16). De to faste hjul kontrolleres separat, og kan ha ulike verdier av vinkelfarten mens caster hjulet er passiv. En slik robot kan rotere på stedet (dvs. uten å flytte midtpunktet mellom hjulene), forutsatt at vinkelhastigheten på de to hjulene er like og har motsatt retning.

Et kjøretøy med tilsvarende mobilitet er innhentet ved hjelp av en *synchro-drive* kinematisk ordning (Fig. 1.17). Denne roboten har tre styrbare hjul som er synkront drevet av bare to motorer via en mekanisk kopling, f.eks. en kjede eller en overførings belte. Den første motoren styrer rotasjon av hjul rundt den horisontale aksen, og dermed gir den drivende kraften (trekkraft) til kjøretøyet. Den andre motoren styrer rotasjon av hjulene rundt den vertikale aksen, dermed påvirker deres orientering. Merk at selve basen endrer ikke retning under bevegelse. Ofte brukes en tredje motor i denne type robot til å rotere uavhengig øvre del av kabinettet i forhold til den nedre delen. Dette kan være nyttig for å orientere vilkårlig en retningsbestemt sensor (f.eks. et kamera), eller i alle fall for å gjenopprette en orienterings feil.

I et *trehjuls* kjøretøy (Fig. 1.15) er det to faste hjul montert på en bakaksel og et styrbart hjul foran. De faste hjulene er drevet av en singlemotor som styrer deres trekkraft, mens styrbare hjul blir drevet av en annen motor som skifter retning. Det fungerer da som en styringsgruppe- enhet. Alternativt kan de to bakhjulene være passiv og forhjulet kan gi veigrep samt styring.

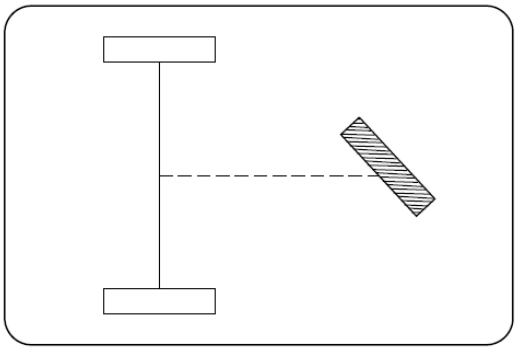


Fig 1.18. En trehjuls mobil robot.

En bil-lignende firehjuls kjøretøy har to faste hjul montert på en bakaksel og to styrbare hjul montert på en foraksel, som vist i fig. 1.19. Som i forrige sak, gir en motor (foran eller bak) trekkraft, mens den andre endrer orientering av forhjulene i forhold til kjøretøyet. Det er verdt å påpeke at for å unngå sluring, må de to forhjulene ha forskjellige retninger når bilen beveger seg langs en kurve, spesielt det interne hjulet er litt mer styrt i forhold til eksterne. Dette garanteres ved bruk av en bestemt enhet kalt Ankermanns styring.

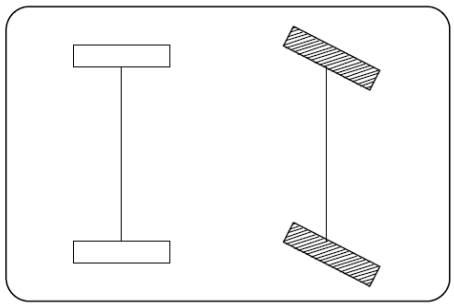


Fig. 1.19 Fire hjuls mobil robot.

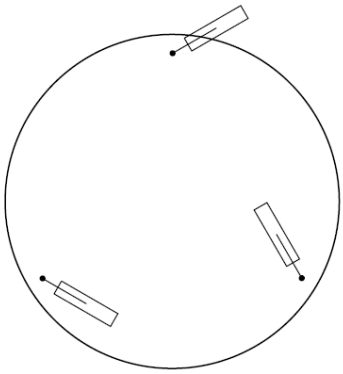


Fig. 1.20. En retningsuavhengig mobil robot med tre uavhengig drevet caster hjul.

Til slutt kan vi se på roboten i Fig. 1.20. som har tre caster hjul vanligvis arrangert i et symmetrisk mønster. Trekkraftens hastigheter på de tre hjulene er uavhengige. I motsetning til de tidligere saker, er dette kjøretøyet omnidireksjonell: faktisk, kan den flytte umiddelbart i alle kartesiske retninger, samt re-orientere seg på stedet.

I tillegg til ovennevnte konvensjonelle hjulene, det finnes andre spesielle typer hjul, blant disse er spesielt Mecanum (eller svensk) hjul, vist i Fig. 1.18. Dette er et fast hjul med passive valser plassert langs ekstern rim, rotasjonsaksen av hver valse blir vanligvis tilbøyelig med 45 ◦ med i forhold til bevegelsesretning av hjulet. Et kjøretøy utstyrt med fire slike hjul montert parvis på to parallelle aksler er også omnidireksjonell.

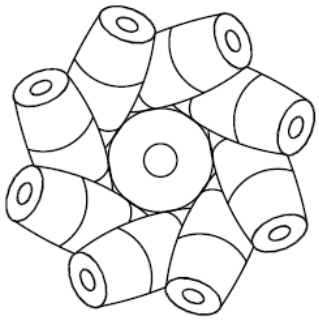


Fig. 1.21. Meganum eller Svensk hjul.

I utformingen av en hjulgående robot, den mekaniske balanse av strukturen representerer ikke et problem generelt. Spesielt er en tre-hjuls robot statisk balansert så lenge dens tyngdepunkt faller inne i støttens trekant, som er definert av kontaktpunktene mellom hjulene og bakken. Robot med mer enn tre hjul har en støtte polygon, og dermed er det lettere å beholde balansert tilstand. Det bemerkes imidlertid at når roboten beveger seg på ujevnt terreng en suspensjon system må holde kontakten mellom hvert hjul og bakken.

I motsetning tilfelle av manipulatorer, arbeidsområdet av en mobil robot (definert som den delen av omgivelsene som roboten kan få tilgang) er potensielt ubegrenset. Likevel, den lokale mobiliteten til en ikke-omnidireksjonell mobil robot er alltid redusert, for eksempel, trehjuls roboten i Fig. 1.18. kan ikke flytte momentant i en retning parallelt med bakakselen. Til tross for dette faktum, kan *tricycle* manøvrere slik at til slutten den oppnår ønsket forskyvning i den retningen. Med andre ord, mange mobile roboter er begrenset på de tillatte øyeblikkene bevegelser, uten egentlig å hindre muligheten for å oppnå alle posisjoner og orientering i arbeidsområdet. Dette innebærer også at antall frihetsgrader for roboten (ment som antall tillatte momentant bevegelser) er lavere enn antall DOF av sine konfigurasjons variabler.

Det er selvsagt mulig å montere en manipulator på en av mobil kjøretøy. Slik robot kalles en mobil manipulator og kombinerer fingerferdighet av leddet arm med ubegrenset mobilitet av basen. Et eksempel på slik robot er en mekanisk struktur som vist i Fig. 1.22. Men utformingen av en mobil manipulator innebærer ytterligere vanskeligheter relatert, for eksempel, til den statiske og dynamiske mekaniske balanse av roboten, samt til aktivering av de to systemene.

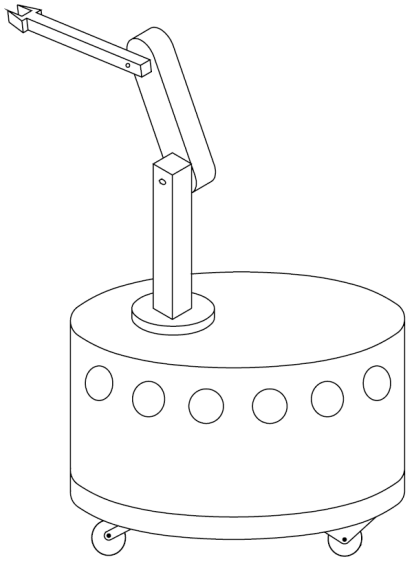


Fig. 1.22. En mobil manipulator oppnådd ved montering av en antropomorfisk arm på en differential-drive kjøretøy.

Mikrokontroller

Førhistoriske folk begynte å telle ved hjelp av sine ti fingre, så det var et desimaltall system. Videre ble lagt til andre systemer avgjørende ved sine lokale problemer. Med oppfinnelsen av datamaskiner og videre utvikling av dem startet bruk av binære system. Informasjon uttrykkes ved hjelp av elektrisk spenning. Best er å ha to nivå av spenningen som har binær kode; 0 og 1. De kalles en bit. En bit er den minste regneenheten i en datamaskin, og den har enten verdien ett eller null. En byte er definert som åtte biter, det vil si en kombinasjon av åtte ettall eller nuller, som 10001101. Tidligere var hver bokstav i alfabetet representert av en kode på nettopp 8 biter i datamaskinspråket, og en byte tilsvarte derfor den datamengden som en enkelt bokstav brukte på datamaskinen. Etter hvert er man imidlertid gått over til koder på 16 biter eller mer, fordi det gjør det mulig å beskrive flere tegn. I datamaskinens ungdom lå bitene rent fysisk lagret i såkalte ferrittlagre, det vil si små ringer av jern. Ringene kunne magnetiseres i to forskjellige retninger når det ble sendt strøm gjennom dem, og hver retning representerte henholdsvis 1 eller 0. I dag består datamaskinens arbeidsminne, RAM, av et enormt antall transistorer, og informasjonen avleses ved å måle om det er spenning på transistoren eller ikke. Ingen spenning tilsvarer verdien 0, og spenning verdien 1. Arbeidslageret fungerer imidlertid bare så lenge det er strøm på datamaskinen. Skal informasjonen lagres, må den oppbevares på harddisken eller i et eksternt lager, for eksempel en CD. På harddisken oppbevares informasjonen som små magnetiske ”klatter”, mens de på en CD oppbevares som fordypninger. En fordypning tilsvarer 0, et område uten fordypning 1. Laserlysstrålen som avleser CD-en, reflekteres forskjellig avhengig av om den treffer en fordypning eller ikke, og refleksen kan oversettes til rekker av 0 og 1. Den lagrede informasjonen er gjenskapt.

En **mikroprosessor** grupperer biter til ord. Første mikroprosessorer jobbet med ord som bestod av fire bit. En mikroprosessor er den komponenten som inneholder hele sentralenheten (CPU) i et datasystem. Mikroprosessorer stammer fra 1971, da den første mikroprosessoren Intel 4004 så dagens lys. I dag kjenner vi mikroprosessorene som bl. a. CPU-enhetene i PC-maskiner. Intel Pentiumbrikker er en av de ledende her.

En **mikrokontroller** inneholder en mikroprosessor i tillegg til de fleste andre funksjoner som en nødvendige i et lite datasystem. Enkelt sagt er en mikrokontroller en liten, (nesten) komplett datamaskin med:

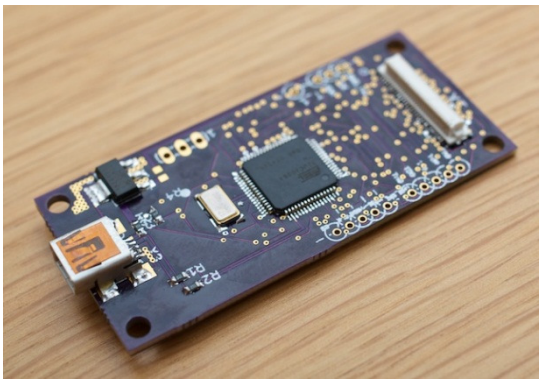
- CPU (Central Processing Unit)

- Hukommelse

- IO-enheten

- Interne og eksterne databusser

CPU-en er selve hjernen i mikrokontrolleren. Denne enheten utfører alle beregningene som mikrokontrolleren kan utføre. Det er også CPU-en som bestemmer hvordan og i hvilken rekkefølge programmene skal utføres, og som tar alle kritiske avgjørelser som er nødvendig for at mikrokontrolleren skal kunne fungere. Vi sier den utfører aritmetiske og logiske operasjoner. I en mikrokontroller er det vanlig å benytte forkortelsen MCU (Microcontroller CPU) som navn på CPU. MCU er integrert på samme brikken (IC) som internminne (program og data) og IO-funksjoner.



Mikrokontroller Atmel AVR32

Hukommelsen er absolutt nødvendig for å få mikrokontrolleren til å fungere. Alle programmer som skal utføres og alle data som behandles, må lagres i hukommelsen.

De hukommelsestypene som er vanlige i mikrokontrollere er:

1. FLASH PROM (for permanent programlager). FLASH-teknologien lar oss programmere en hukommelsesblokk permanent. Det er også mulig å nulle ut området igjen og reprogrammere det.
2. UV-PROM (for permanent programlager). UV – ultra violet. Dette er en hukommelsestype som kan programmeres igjen og igjen. Ulempen i forhold til FLASH er at vi må benytte UV-lys for å slette hukommelsen før reprogrammering. Dette kan ta 15 – 20 minutter. I tillegg er det en ulempe at IC-en må utstyres med et kvartsvindu over brikken for at lyset skal kunne slippe inn. Dette fordyrer komponenten.
3. SRAM (for generell datalagring). Statisk RAM. Dette betyr at vi kan lese og skrive til hukommelsen og at innholdet beholdes så lenge spenningsforsyningen til komponenten er på.
4. EEPROM (for permanent lagring av viktige systemparametre). Elactrically Erasable PROM. Vi kan slette innholdet elektrisk, men i motsetning til FLASH slettes en og en celle. Dette er en omstendelig prosess som tar lang tid, men som til gjengjeld gir oss anledning til å oppdatere innholdet av bare ett ord uten å måtte slette hele PROM-blokken.

IO-enheter som harddisker og CD-lesere benyttes sjelden i forbindelse med mikrokontrollere, men IO-enheter for å kommunisere med annet datautstyr blir mer og mer vanlig. I teorien kan en mikrokontroller virke uten sine IO-enheter. Uten IO-enhetene vil MCU-en fint kunne utføre programmer som allerede ligger i hukommelsen og hente data fra og skrive data til hukommelsen. Men ingenting ville noen sinne kunne påvirke programmene. Likeledes ville heller ingen resultater kunne vises fra beregningene. Det er dette IO-enhetene dreier seg om – å mate informasjon til maskinen og å kunne presentere informasjon fra beregningene i maskinen. Informasjonen som utveksles med maskinen, kan utveksles med brukere direkte, med teknisk måle- og kontrollutstyr eller med andre maskiner.

Informasjon må utveksles mellom enhetene som inngår i mikrokontrolleren. Dette kan skje via separate datakanaler mellom CPU-en og hver av de resterende enhetene i

systemet. Dette hadde blitt både komplisert plasskrevende og dyrt fordi det som regel finnes flere hukommelsesenheter og mange IO-enheter. En enklere og mer systematisk tilnærming til problemet, er å benytte en databuss bestående av et sett av elektriske ledere som snor seg innom alle enhetene i maskinen. CPU-en kan utveksle data med de forskjellige enhetene som etter tur kan benytte databussen. Data som sendes ut på databussen kan egentlig ses av alle komponentene som er tilkoblet bussen. Imidlertid benytter CPU-en en spesiell teknikk som kalles adressering for å sikre at data bare sendes til den ønskede enhet. Bussene i datasystemene fører derfor både adresse- og datasignaler.

Sensorer

Robots sensorer er en av de viktigste komponentene i en funksjonell robot, de er robotens øyne, ører, følelse av kontakt, balanse og eneste kilde til innspill fra det miljøet. Robots avstand og rang sensorer hjelper for navigasjon og unngå hindringer, akselerometre og gyroskoper gir tilbakemelding for å opprettholde balanse og orientering, berørings og kraft sensorer gir tilbakemelding på fysisk kontakt med gjenstander, miljømessige sensorer gir temperatur, fuktighet, og vibrasjon informasjon, og en rekke navigasjonssensorer hjelper roboten bevege seg gjennom verden intelligent.

Det finnes veldig bredt valgmulighet på sensorer som bruker forskjellige metoder for å få nødvendige data. Vi kan se på noen eksempler:

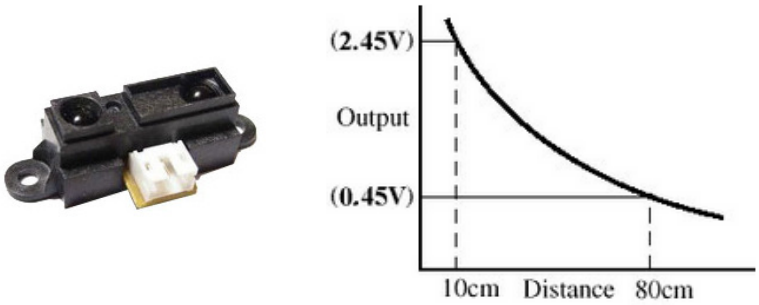


Fig. 2.1. Sharp IR Distance Sensor - GP2D12 og volt x distanse grafikken.

Sharp IR Distance Sensor - GP2D12. (Fig.2.1.) Bruker infrarød stråling for å måle avstanden. Kan brukes fra 10 til 80 cm og gir analog utgang.

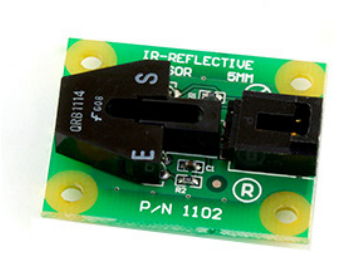


Fig. 2.2. Phidgets IR Reflective Sensor.

Phidgets IR Reflective Sensor. (Fig.2.2.) Denne sensoren kan oppdage et objekt på 5 mm ved hjelp av en fotoreflektor . Kan bestemme forskjellen mellom svart (lave reflekterende forhold) og hvit (høy reflekterende forhold).

Fig. 2.3. Parallax PING Ultrasonic Range Sensor.

Parallax PING Ultrasonic Range Sensor. (Fig.2.3.) Ultrasoniske sensor, en enkel metode for avstandsmåling. Denne sensoren er perfekt for en rekke applikasjoner som krever at du utfører målinger mellom bevegelige eller stillestående objekter. Sensoren er svært populær fordi kan være nyttig i sikkerhetssystemer eller som en infrarød erstatning hvis ønskelig. Sensor måler avstanden ved hjelp av sonar, en ultrasonisk (godt over menneskelig hørsel) puls overføres fra enheten og avstand til målet bestemmes ved å måle tiden det tar for ekkoets retur. Utgang fra PING sensoren er en variabel bredde puls som tilsvarer avstanden til målet.

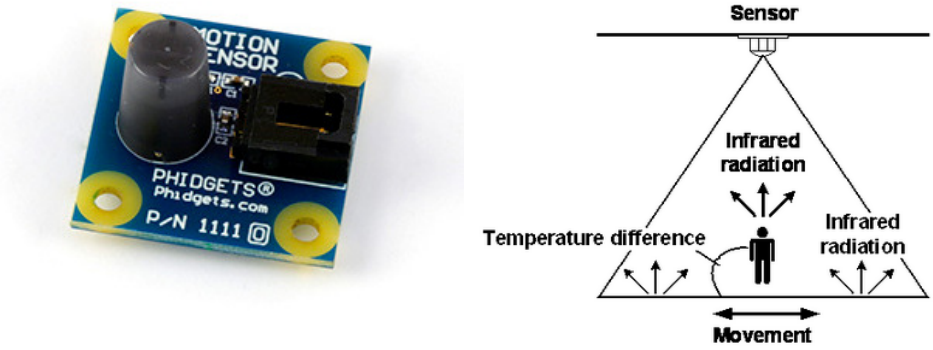


Fig. 2.4. Phidgets Motion Sensor

Phidgets Motion Sensor. (Fig.2.4.) Denne sensoren registrerer endringer i infrarød stråling som oppstår når det er bevegelse ved en person (eller objekt) som er forskjellig i temperatur fra omgivelsene. Som denne sensoren registrerer temperaturforskjeller, er det godt egnet til å oppdage bevegelse av personer av kroppstemperaturen. Sensoren er også preget av en smal sensing område.

Fig.2.5. PhidgetSpatial 3/3/3, måler 9 Axis IMU

PhidgetSpatial 3/3/3, måler 9 Axis IMU (Inertial Measurement Unit) (Fig.2.5.) måler statisk og dynamisk akselerasjon i 3 akser, opp til 5g. Det måler også magnetfelt i 3-akser opp til ± 4 Gauss og til slutt måler rotasjonsvinkel i tre akser, opp til ± 400 ° per sekund.

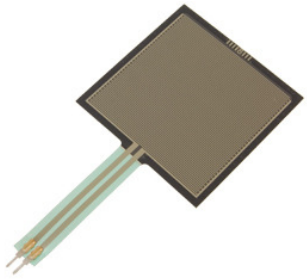


Fig.2.6. Force Sensing Resistor (FSR).

Force Sensing Resistor (FSR). (Fig.2.6.) Den er perfekt sensitive sensor for robots grip, robots bein og like applikasjoner.

Fig.2.7. LifeCam Cinema Webcam-720 HD.

LifeCam Cinema Webcam-720 HD. (Fig.2.7.) Høy oppløsnings USB kamera som kan brukes I alle robots prosjekter.

Fig.2.8. Phidget Temperature Sensor IR.

Phidget Temperature Sensor IR (Fig.2.8.) – infrarød termometer for å måle temperaturen på avstande. Kan måle temperaturen fra -70 til 380 ^{o}C.

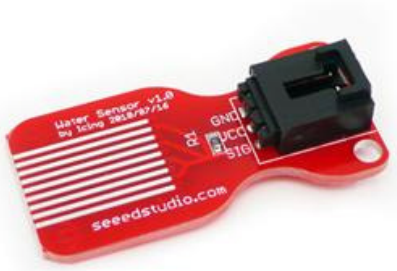


Fig.2.9. Electronic Brick – Water Sensor.

Electronic Brick – Water Sensor. (Fig.2.9.) Vann sensor er laget for vann deteksjon, som kan bli brukt i målinger for nedbør, vann nivå, selv på smelte lekkasje. The Brick er i hovedsak består av tre deler: en elektronisk kontakt, en 1 MΩ rasister, og flere linjer med blanke ledende ledninger.

Det finnes mye mer forskjellige typer av sensorer som mobile robot kan bruke for orientering, navigasjon osv. Det er bare noe få eksempler som var nevnt.

Motorer og aktuatorer

Motorer og aktuatorer gir poboten trekkraft for å bevege seg og utføre oppgaver. På dagens marked kan man finne alle slags motorer og utstyr for roboter. Vi skal se på noen av dem.

Fig.3.1. 14,4V, 720RPM 300oz-in Planetary Gearmotor

En robust rimelig motor som inkluderer planetgir og en kraftig RS775 motor. Girkassen utgående aksel er 0,5 "diameter og 3" lang med et kilespor for enkel montering av hjul og kjedehjul. Den Magnum775 inkluderer et planetgir med en 20:1 reduksjon forhold. Dette

forholdet gir ca. 700 om/min utgående hastighet ved lav belastning på 14.4V. Akse, planetgir og støttende ring laget av herdet stål. Dette hjelper Magnum775 å håndtere den høye dreiemoment produsert av motoren ved tung last.

Fig.3.2. Soyo SY42STH38-0406A Unipolar Stepper Motor

Soyo SY42STH38-0406A Unipolar Stepper Motor en stepper motor ved veldig høy presisjon (0,9 grader/step). Det er en spesialisert form for en elektrisk motor, som roterer med høy presisjon på datamaskinens ordre (mottar en elektrisk puls) til en fast vinkel. For en fullstendig omsetning, bør han utføre et antall trinn gitt av produsenten. Brukes for nøyaktig posisjonering av enheten under datastyring. Mye brukt i robotikk, skrivere og enheter i utskriverens hodebevegelser.

Fig.3.3. Firgelli Automations 2" Stroke 150lb Force Linear Actuator.

Den Firgelli Automations 2 " Stroke 150 Force lineær aktuator er skapt for applikasjoner som krever ro og jevnt lineær drift. De kommer med innebygd endebrytere (ikke bevegelig), aluminium skall, 2 klemme én på hver ende.

Programmering av mobile roboter.

Programmeringen av en robot må være spesialisert og tilpasset dens atferdsmessige modellen. Rodney Allen Brooks fra Massachusetts Institute of Technology(MIT) var den første som foreslo en ny tilnærming til programmering av roboter. Ved bruk eksempel fra insekter, foreslått han å minimere tankegang før handling. Sin algoritme til å utføre robot Brooks kalt «Arkitektur av prioriterte interaksjoner."

Arkitektur av prioriterte interaksjoner kan samle inn en sammenhengende (koherent) system av alle elementene i den mobile robotens kontroll. Fordelen med denne metoden, som kombinerer persepsjon og bevegelse, er at det krever bare beskjedne matematiske ressurser.

Kunstig intelligens kan være tilgjengelig for alle ...

Tradisjonelt programmering.

Ved tradisjonell programmering man prover å modellere omgivelsen som tilpasser for roboten. Programmet inkluderer en rekke underprogrammer som utføres i en tradisjonell rekkefølge.

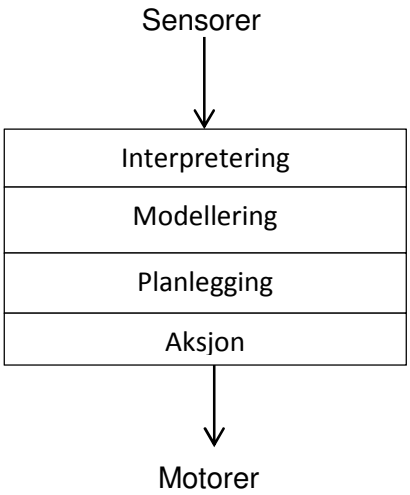


Fig.4.1. Tradisjonell programmering av roboter.

Fordeler.

Ved hjelp av sensorer samles alle data om omgivelsen. For å unngå feil, alle data interpreteres. Modellering av omgivelsen må inkludere geometriske parametere av hindringer og dens plassering. Denne modellen av omgivelsen lar roboten utgjøre oppgaven steg for steg og gir mulighet optimere handlingen og unngå feil.

Ulemper.

Denne metoden krever store matematiske beregninger. I reelle situasjoner finnes veldig mye data som må beregnes og det krever stor hukommelse. Sensorer må være presise. Men alle sensorer i praksis har noen teknologiske feil bestemte av målingsmetoder. Det fører til at innkomme data fra forskjellige sensorer er ikke i samsvar med hverandre.

Prioriterte interaksjoner

Denne metoden kombinerer styringen av robot i real tid med robots reaksjoner som kommer av sensorens data. Istedenfor å kontrollere innkommende data fra forskjellige sensorer robot reagerer på data som har høyest prioritet. Hvis forsvinner data fra sensoren som har høyeste prioritet robot reagerer på data fra sensoren som har lavere prioritet. På den måten robot reagerer på hver hendelse.

For eksempel vi kan se på en robot som har forskjellige sensorer: locator – gruppe sensorer som bestemmer lokalisasjon, IR avstandsmåler.

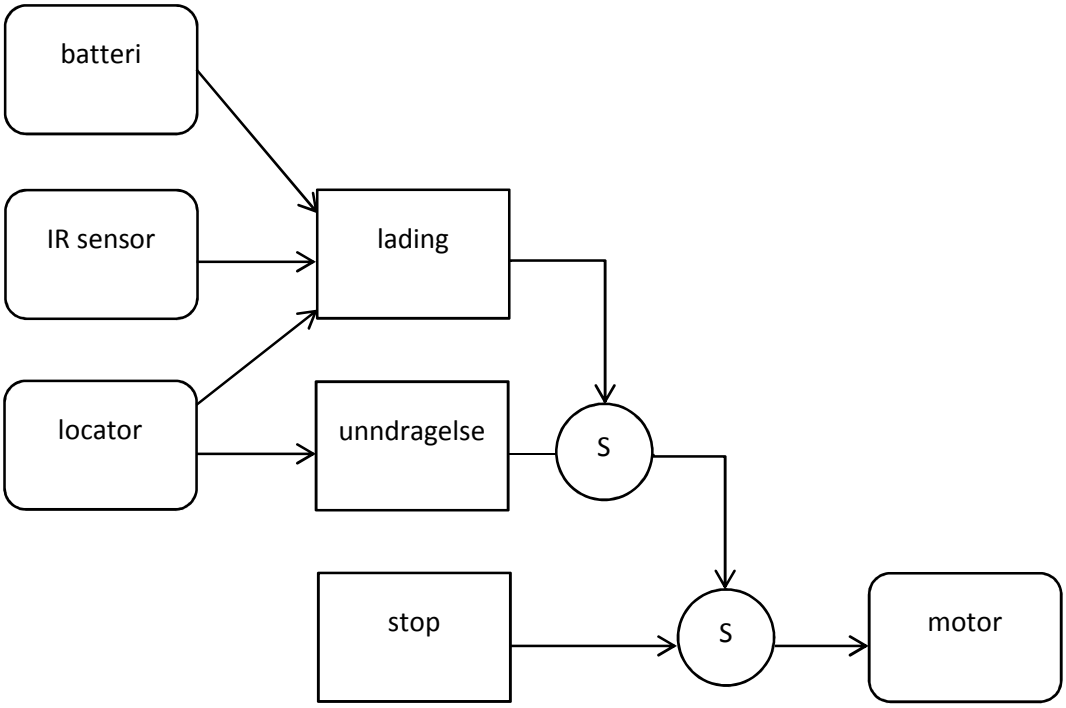


Fig.4.2. Programmering ved prioriterte interaksjoner.

Mål for roboten er å bevege seg gjennom et rom og unngå hindringer. Nede navnet underprogrammer og robots adferds respons plassert fra lav til høy prioritets rekkefølge.

- Stop – adferds respons som stopper robot
- Motor – underprogrammet som aktiverer motor
- Locator – underprogrammet som måler avstanden i nåtiden
- Unndragelse – adferds respons som beholder en bestemt avstand mellom roboten og hindringen. Robot snur seg og fortsetter bevegelse.

- Lading – ved nesten tomt batteri robot beveger seg til ladestasjon. Den bruker informasjon fra locator, IR sensor og batteri. Ladestasjon har en IR transmitter som bestandig sender stråling. Punkt S er stedet for samhandling av de prioriterte.

Denne programmerings metode har særskilte parallelle oppgaver. Det gir mulighet å programmere uavhengig hver oppgave og sette til roboten ny funksjoner uten å miste tidligere. Man må bare binde de sammen og vurdere prioriteter.

Største problemet med programmeringen er å sette sammen forskjellige programmer uten at de går til å konfrontere med hverandre. Det finnes også mekaniske feil som må beregnes ved konstruering av roboten. Forskjellige typer av sensorer gir ulike resultater og man må velge og prioritere de riktig.

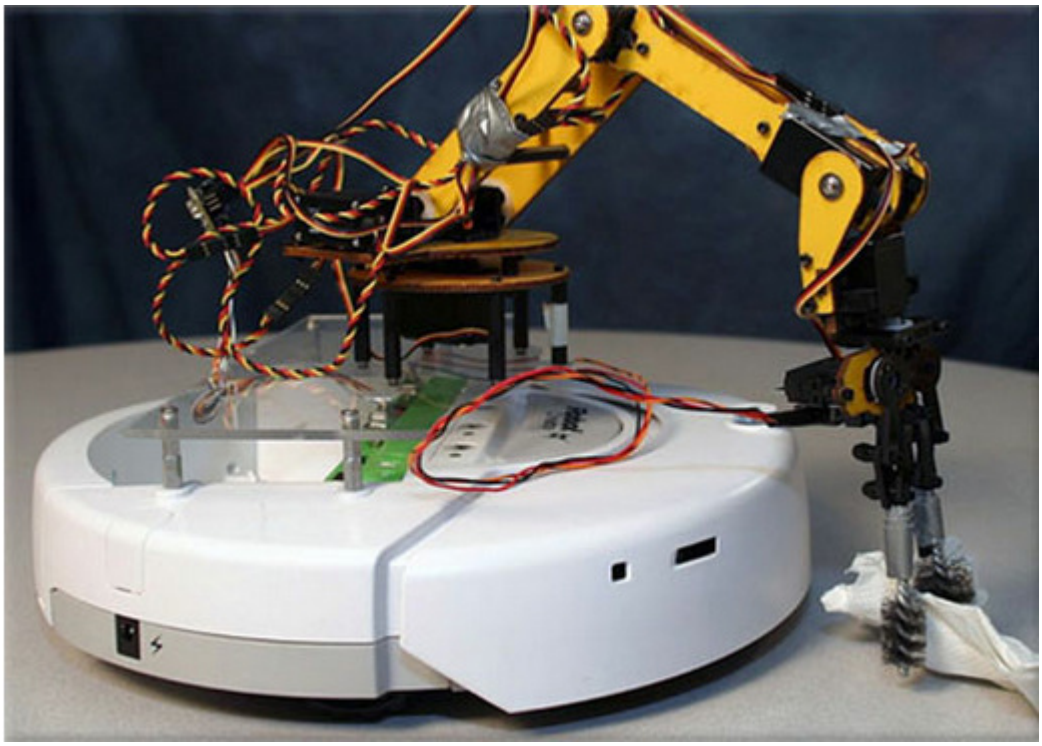
En mobil robot må orientere seg i omgivelsen. Å gjøre det i praksis er ikke så lett. Hvis robot beveger seg inne i et rom, den kan dra langs vegger og reagere på lys kilder. Det krever få sensorer og programmeres lett. Men hvis robot må bevege seg på et åpent og særlig ujevnt terreng uten noen orienterings punkter? Førprogrammering som forutsetter bestemte operasjoner som bevegelses lengde, snurring osv. virker ikke. To motorer kan ikke jobbe helt synkront eller med like hastighet. Denne forskjellen fører til at roboten går ikke rettfram. Ved sånne små feil roboten mister orientering. Dette kan låses hvis man bruker en turteller for hjul eller motorer. Men når robot starter, stopper eller snur hjulene sklir. Man kan ikke unngå dette. En løsning er kontinuerlig korrigering av posisjon. Dette krever ekstra utstyr; sensorer, GPS antenner, magnetisk kompass osv. og selvfølgelig alt må programmeres, bindes sammen og prioriteres. Man må finne riktige algoritmer for roboten i helheten.

Praktisk del

I den praktiske delen kan vi bruke en tjeneste fra kompaniet «iRobot» som har navnet iRobot Create ®.

Det er en rimelig mobil robot plattform for lærere, studenter og utviklere som kan brukes til å eksperimentere, lære og spille, lære det grunnleggende av robotikk, informatikk og ingeniørvitenskap, programmets atferd, lyder og bevegelser. Man kan feste sensorer, elektronikk, kameraer, grippers og mer.

Nybegynnere kan observere robotens oppførsel i noen av de 10 demonstrasjons modusene eller ved å laste ned et skript med noe grunnleggende terminal programmer. Avanserte brukere kan skrive et tilpasset programvare ved hjelp av en rekke metoder som kan dra nytte av robotens "streaming sensor data"-modus for mer kontroll over roboten. Og svært avanserte brukere kan skrive programmer for helt selvstendig atferd.



«iRobot» er grunnlagt i 1990 av Massachusetts Institute of Technology roboticists. iRobot designer og bygger noen av verdens viktigste roboter. Mer enn 8 millioner hjemme roboter har blitt solgt over hele verden. Mer enn 4500 roboter har blitt levert til militære og sivile forsvar over hele verden. De utfører tusenvis av farlige søk, rekognosering og bombe-deponerings oppdrag.

iRobots hovedkontor ligger i Bedford, Massachusetts. Selskapet har også kontorer i Virginia, North Carolina, Florida, California, Storbritannia, Frankrike, Kina og Hong Kong.

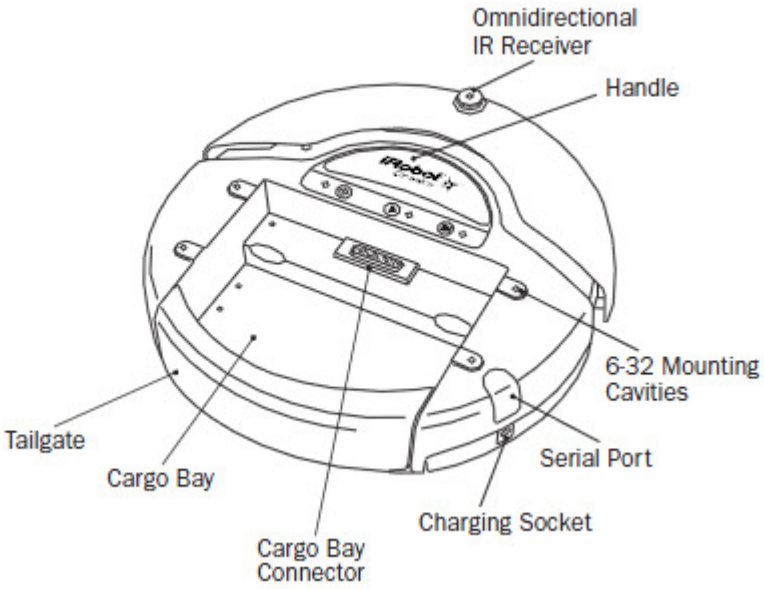
Som en pioner i robot bransjen, har iRobot hatt som mål å drive innovasjon, tjene som en nærings katalysator, og forandre verden ved å starte opp en æra av roboter.

For å støtte og oppmuntre til utvikling av robotteknologi, tilbyr iRobot omfattende ressurser for tredjeparts utviklere, gir informasjon og produkter som letter etablering og enkel integrasjon av nye nyttelast, atferd og evner på iRobot plattformer. iRobot samarbeider med eksterne utviklere fra offentlige etater, akademiske institusjoner, og små og store bedrifter for å skape og bringe til markedet innovasjoner som forbedrer hverdagslivet og hjelper fagfolk å utføre farlige oppdrag ved å skape mindre risiko.

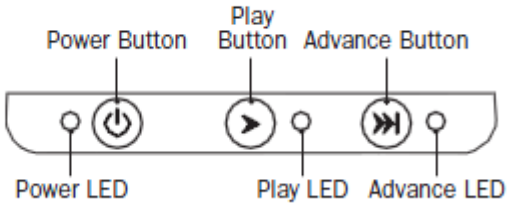
Med mer enn tjue års lederskap i robotbransjen, forblir iRobot forpliktet til å gi plattformer for oppfinnelse og oppdagelse, utvikle viktige partnerskap for å fremme teknologisk utforskning og bygge roboter som forbedrer livskvalitet og sikkerhet over hele verden.

Anatomi

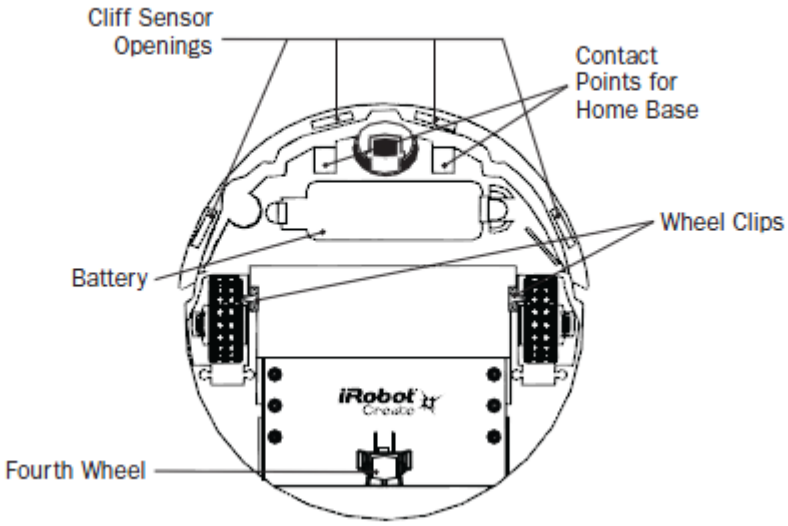
Top View



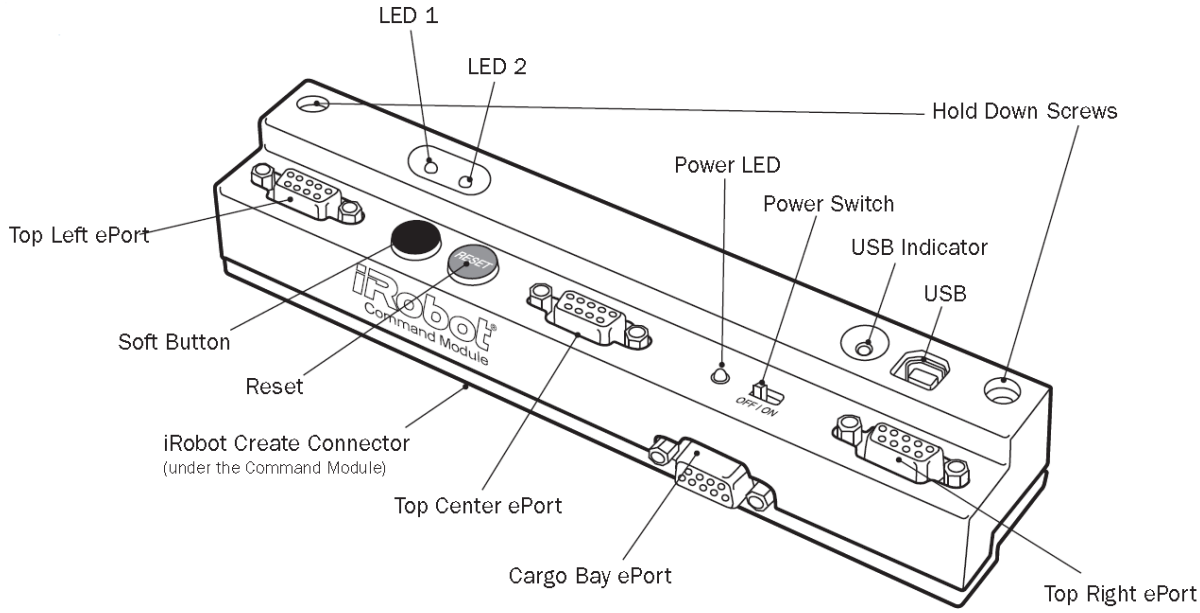
Buttons and Lights



Bottom View



iRobot Command Module



IRobot Command Module arbeider med iRobot Create, og gir mulighet til å skrive egne programmer i C eller C + +, og man kan legge de til maskinvare og utvide Create ved tilknytting roboten til en PC. Kommandoen Modul plugges inn i Cargo Bay Connector, og den har fire ekspansjonsporter som gir flere innganger og utganger for den tilpassede maskinvaren. Tre av kontaktene er på overflaten, det gir plass for enkle festepunkter for en rekke sensorer, og en kontakt på baksiden for enkel tilgang fra lasterom. Command Module drives av en Atmel AVR ATMega168 mikrokontroller som kan omprogrammeres ved å laste ned programmer fra Windows - datamaskinen med den medfølgende USB-kabelen. Selvskrevet programmer kan bruke iRobot Åpen Interface seriell protokoll for å kontrollere Creates motorer, lys, og høyttaler, og få informasjon fra sensorene. Samtidig kan mikrokontrolleren direkte samhandle med din egen tilpasset maskinvare gjennom sine I / O tilkoblinger.

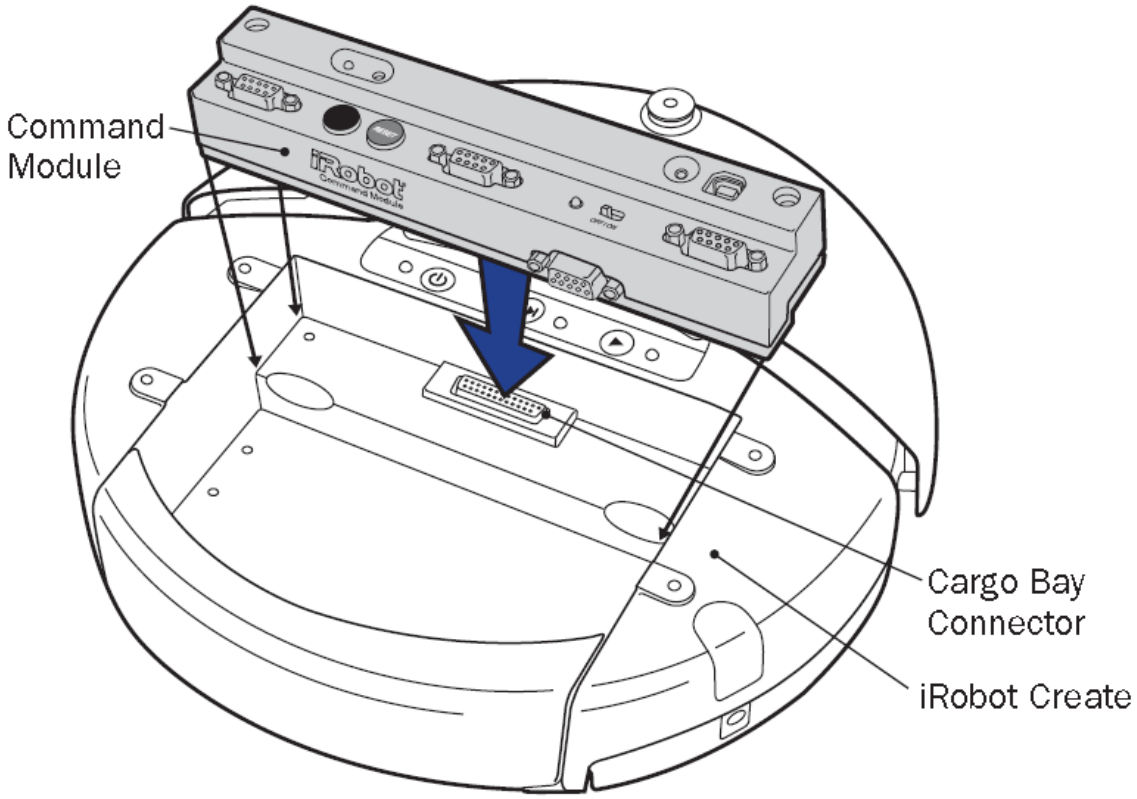
Man kan starte med et av eksempel programmene, utvide og endre det ved å legge til sine egne funksjoner. Oppdateringer og flere informasjon er tilgjengelig på www.irobot.com.

Command Module åpner mange spennende muligheter. Programmene er kun begrenset av sin egen fantasi. Man kan lære robotikk og programmering ved videregående skoler eller universiteter. iRobot Create er en robust, rimelig robot plattform som gir elevene mulighet til å ha sin egen robot: Legge til nye sensorer og utføre egne robotikks eksperimenter, lage en lavkost sverm av roboter for å undersøke kollektiv atferd, ha det gøy med robot "kunst" utstillinger, sang og dans eller andre underholdende atferd, legge til et kamera og internettforbindelse for å skape en vakt robot.

Command Module kommer med pre installert demo program som kan vise at alt fungerer som det skal. Drivere installeres først på datamaskinen, det konfigureres et bestemt USB port. Command Module kan da kobles til den USB porten.

Plugge inn Command Module

Command Module kobles til Cargo Bay Connector som visst på Figur 1



Innsetting av Command Module inn Create kontakt.

Programmer

Man kan begynne å bruke Command Module ved hjelp av eksempel programmer som er forhåndsprogrammert på Command Module. De kalles drive og input og dens kilde koden er i "Sample Projects" katalog på Command Module Produkt CD.

Den Command Modul bruker WinAVR suite av åpen kildekode utviklingsverktøy for å kompilere og laste ned dine C eller C + +-programmer. WinAVR programmen inkluderer GNU GCC-kompilatoren, avrdude nedlaster og Programmers Notepad IDE.

Det drive eksempel programmet viser de grunnleggende funksjonene av Command Module og iRobot Create Open Interface ved å kjøre rundt, reagere på sensorer, og bruke ulike inndata og resultater. Det gjør en serie stigende piper slik at du vet det Command Module virker og blinker Create sine lysdioder (strøm lys har oransje farge). Deretter, hvis du plasserer Create på gulvet og trykker på Commando Module soft knapp begynner det å kjøre på gulvet, rygge og snu når den treffer noe med støtfangeren sin, blinke med alle lysdioder i et mønster. For å stoppe programmet, trykk på knappen igjen eller bare løft den opp. Koden gir nyttig funksjon for å initialisere Command Module og iRobot Create, kontinuerlig lese

Create sin sensor verdier, drive Create, slå på Create, og endre overføringshastighet. Du kan addere evner til den hoved funksjonen.

Den input eksempel programmet illustrerer de tre typene av input og output som du kan bruke med Command Module. Du kan samhandle med maskinvare på iRobot Create gjennom Open Interface, bruker den innebygde Command Module maskinvare, og legge til din egen maskinvare til Command Module kontakter. For dette enkle programmet blinker tre lamper, en på hver type av programmet. Hvis det blir et tastetrykk, lyser den tilhørende LED og det spilles en sang. Du kan legge til din egen knapp og LED og prøve å bruke alle tre typer input og output.

Light det er et eksempel program som viser en mulig benyttelse av Command Module og iRobot Create. Den undersøker huset for lys som er igjen på og spiller en alarm når den finner et lys slik at du kan slå den av. Det programmet holder alle Create og Command Modul lysdioder slått på (merk at strøm lys vil være oransje) så det er lett å finne den i mørket.

Start den opp i en relativt mørkt rom ved å trykke på soft knapp. Det vil vurdere lysnivået når det starter og vil slå på alarm når den oppdager en betydelig lysere rom. For å oppdage lysets nivå, bruker programmet en CDS lys sensor som du må legge til Command modulen øverst på midten ePort.

Prosessor

Prosessoren i Command Module er en Atmel ATMega168 8-bits mikrokontroller. Alle operasjoner på prosessoren er utført med 8-bits registre (32-bits og 64-bits registre i de fleste hjemme datamaskiner).

For best effektivitet, bruk 8-bits variabler og beregninger så langt det er mulig. 16-bits variabler må brukes bare der hvor det høyere range er nødvendig. Ikke bruk flyttall eller divisjon i din Command Module kode. ATMega168 har ikke maskinvare for å håndtere flyttall eller dele operasjoner. Hvis du bruker disse, blir din kompilert kode stor veldig fort. I stedet bruk heltall og riktig skift (>>).

Prosjekt

Den enkleste måten å starte et prosjekt er å kopiere en av eksempel programmer og endre den til din egen applikasjon. For eksempel, rammeverk for å sende kommandoer og lese signaler fra sensorer, som tilbys av drive eksempel kan være nyttig for de fleste bruksområder. For å opprette ditt eget prosjekt basert på drive eksempel utfør følgende trinn:

1. Kopier drive.c, oi.h, og makefile fra drive eksemplet katalogen til det nye prosjektet katalogen.

2. I den nye katalogen, endre navn drive.c til ønsket navn for det nye programmet, som myapp.c

3. Rediger makefile for å gjenspeile det nye navnet på applikasjonen. Spesielt, må du endre verdien av TARGET til navnet på den nye applikasjonen. På linje 59 av makefile, vil du se linjen:

TARGET = drive

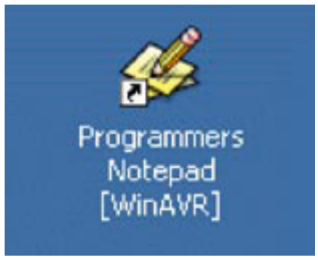
Dette må endres til det nye navnet på appen.

For eksempel:

TARGET = myapp

Deretter endres oppførselen til programmet som passer til dine nye applikasjoner ved å endre C-kode. I *drive.c*, atferden av Create er definert i *mine* funksjonen.

Alle operasjoner med programmer gjennomføres ved hjelp av Programmers Notepad. Dette er et hjelpeprogram som følger med iRobot Create og kan installeres på datamaskin.



Du kan importere eksempelprogrammer til en ny mappe, så endre programmer eller lage dine egne programmer med Programmers Notepad, gjennomføre kontroll over endrede eller nye programmer og etterpå laste de til Command Module gjennom en USB kabel.

iRobot Create som jeg jobber med har bare støtte og lys sensorer og det begrenser variasjoner som jeg kan bruke ved programmeringen.

Arbeids beskriving.

Jeg starter iRobot Create for å se på robots oppfølging. Etterpå laster jeg til Command Module eksempel program fra filen **drive** (se vedlegg). Programmer er identiske, men kommandoer fra Command Module virker saktere enn fra basen.

Algoritmen ser ut sånn: etter starten gir robot startelyd og begynner å bevege seg lans divergerende spiralen fram til den treffer første hindringen, gir lydsignal, rygger litt, snur seg, fortsetter bevegelse rettfram til neste treff osv. På denne måten støvsuger dekker først åpent område og videre beveger seg helt kaotisk uten noen fastsatte system. Etter treffet vinkelen til rotering i programmet satt til tilfeldige tall fra 53 til 180 grader.

Det ser ut som visst på Fig 5.1

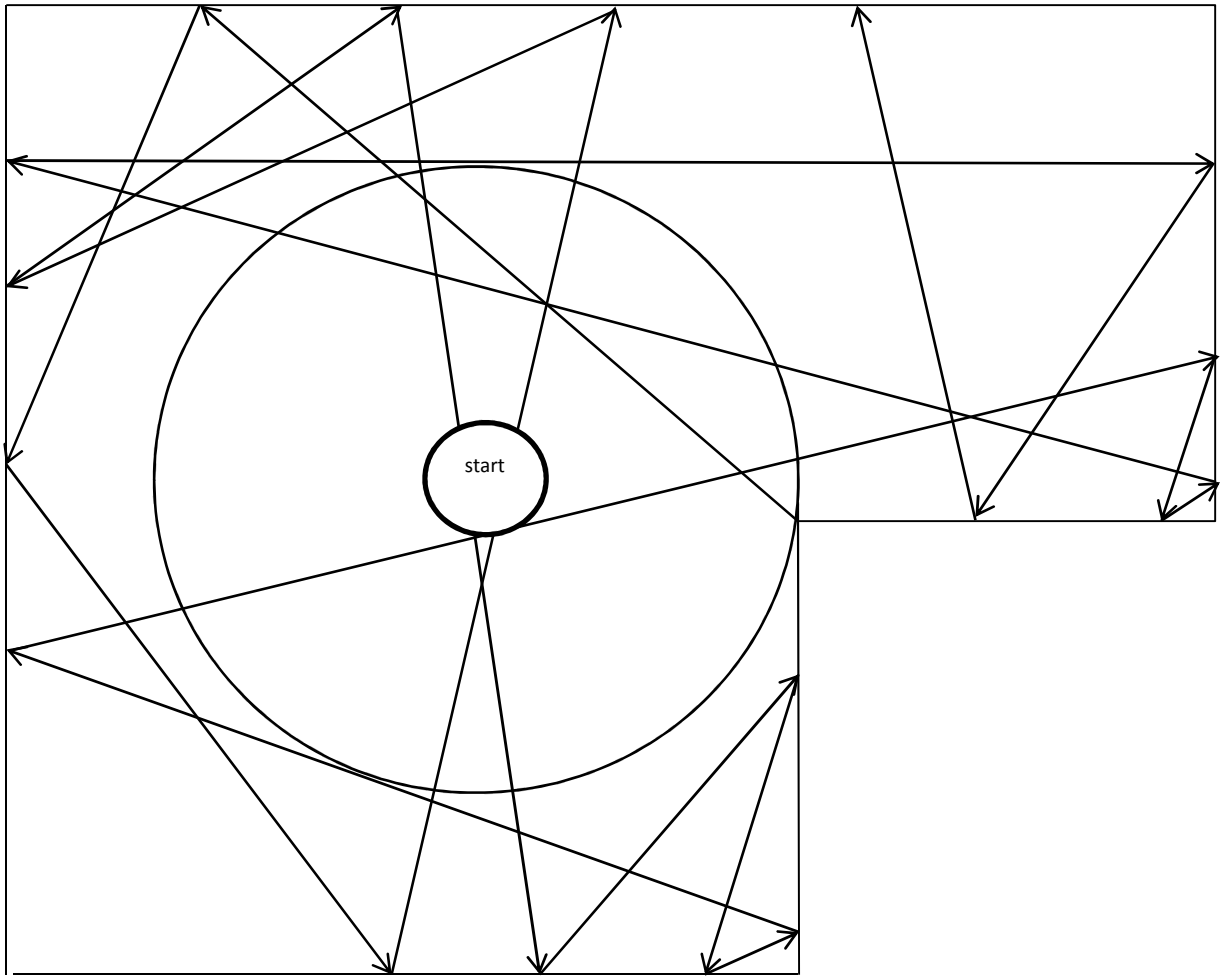


Fig.5.1. Robots bevegelse skjema.

Denne algoritmen fører til at støvsuger krysser det samme område mange ganger. Det er ikke en veldig energisparende metoden. Deretter omprogrammerer jeg Command Module for å oppnå resultaten som visst på Fig.5.2.

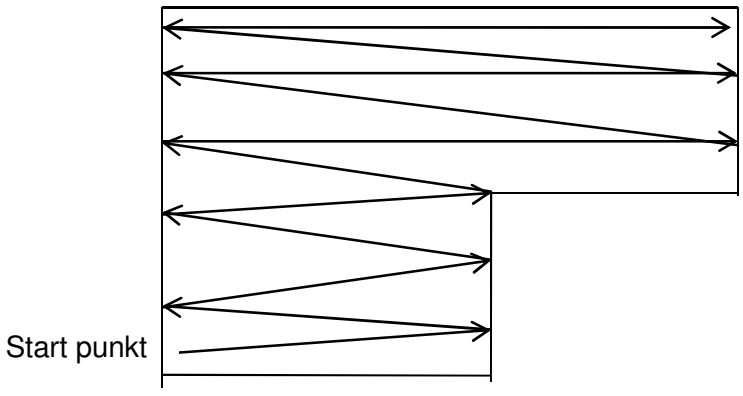


Fig.5.2 Robots bevegelse ved fast rotasjonsvinkel.

I praksis ser det ikke så rett ut som på grafikken. Det var nevnt tidligere feil med førprogrammering. Roboten beholder ikke presis rotasjons vinkel, etter treff av veggen støvsuger bytter retningen litt. Feil summeres og målet oppnås ikke.

Det samme skjer hvis jeg plasserer en hindring midt i området.

Førprogrammering virker ikke. Jeg har ikke mulighet til å bruke parallelt prioritetsprogrammering pga. manglende resurser.

Jeg fant annet løsning.

Jeg kjøpte en selvgående robotstøvsuger Samsung Navibot Silencio SR8895.

Robotstøvsuger Samsung Navibot Silencio SR8895.

Det er siste utgave fra Samsung og som produsenten siterer selv: «Takket være det intelligente navigasjonssystemet klarer Navibot Samsung SR8895 alt selv. Takket være sitt innovative system kan den rengjøre alle slags gulv på egen hånd, bestemme den optimale ruten og nå ut til alle hjørner. Takket være systemet Visionary Mapping™ kan Navibot velge den mest effektive rengjøringsruten i hjemmet ditt. Med hjelp av det innebygde kameraet som kan ta 30 bilder i sekundet lager den et komplett kart over hjemmet og lagrer den beste veien. I tillegg vet den alltid hvor den er og hvor den er på vei: den er så avansert at etter en tur til ladestasjonen husker den hvor den sluttet, og vender tilbake dit. Vi ville lage et apparat med lysende egenskaper, som også gjør gulvene dine skinnende rene. «

Den har mye sensorer, et kamera, nåtids respons, 8 moduser og ser ut så avansert!

Jeg prøvde den hjemme. Jeg har en leilighet med forskjellige møbler mørke og lys. Gulvet dekket av laminat og på noen steder ligger tepper.

Det var en skuffelse. Støvsuger orienterer seg veldig dårlig. Ofte ignorerer bein av borer og stoler. Noen ganger merker ikke mørke møbler og vegger. Lys sensorer var kanskje ikke så bra, men etter at den kjørte inn i veggen, rygger den og kjører inn igjen. Navibot finner ladestasjonen fra et meters avstand men ikke fra andre kanten av rommet. Da kan jeg

konkludere at IR sensor også var ikke helt perfekt. Felles grunnen til alle feil kan være også programmering feil eller svak prosessor som er overbelastet av stor mengde informasjon.

Neste eksperiment var iRobot Roomba 780 robotstøvsuger.

iRobot Roomba 780 robotstøvsuger

Reklamen fra selgere og produsenten handler om robots gode vaske og ryddings evner og ikke noe om gode orienterings metoder.

Oppfølgings algoritme var det samme som på iRobot Create. Først går rundt med større og større radius fram til første hindringen. Den treffer hindringen veldig sjelden, oftest robot beholder en liten avstand. Roboten fullfører oppgaven.

Konklusjon

Mest avanserte robotstøvsuger Samsung Navibot Silencio SR8895 virker dårlig. Hvorfor? Jeg har ikke tilgang til programmet for Navibot. For å finne svar prøvde jeg en tidligere utgave fra Samsung, robotstøvsuger Samsung Navbot SR8855.

Robotstøvsuger Samsung Navbot SR8855.

Den var ikke så avansert som siste utgave, har mindre antall av sensorer, men gjør oppgaven sin. Går litt saktere enn SR8895 treffer møbler og vegger ikke så ofte og ikke så hardt. Ved svak batteri finner ladestasjonen.

For å lage en avansert robot man må starte fra en sterk prosessor som tilfredsstiller gitt oppgaven. For å ha mer informasjon om omgivelsen og om oppgaver må roboten ha stor minne, store beregningsmuligheter og nok antall av presise og pålitelige sensorer. Alt må virke i enighet uten konflikter ved hjelp av god softverk. På andre siden en sterk prosessor og stor mengde av utstyr øker energibruk og kostnader.

Enkelt støvsuger iRobot virker bra og tilfredsstiller gitt oppgaven. Det bekrefter uttrykk det enkle er ofte det beste. Kanskje Navibot kunne være den klokeste, men var begrenset av kostnader.

Produktutvikling inne i robotikken og programmeringen går veldig fort. Kommer ny teknologier og materialer som gir mulighet til å lage konstruksjoner vi kan ikke forestille i dag.

Et godt eksempel på det (ikke akkurat fra robotikken) er V-22 Osprey – konvertoplan som kombinerer individuelle evner av fly og helikopter.

Konvertoplan V-22 Osprey

Utviklet i USA i over 30 år med Boeing og Bell. Flyet er utstyrt med to motorer, T406 Allison, plassert på vingen i en gondol, som kan roteres nesten 98 grader. Det tok så langt tid fra projekten til produksjon fordi da var ikke materialer som dekker krav.

Kanskje det er ikke så langt fra tida når man kan komme hjem og si fra til personal roboten hva han vil spise på kvelden…

Litteraturreferanser

Internettressurser:

http://www.irobot.com/us/

http://store.irobot.com/shop/index.jsp?categoryId=3311368

http://www.razorrobotics.com/

http://www.robotechnicas.ru/

http://www.pcweek.ru/themes/detail.php?ID=66917

http://www.robotiksistem.com/mobilerobot.html

http://www.galileo.org/robotics/design.html

http://www.robotshop.com/sensors.html

http://www.samsung.com/no

http://www.allonrobots.com/index.html

http://uctrl.net/files/e/e0/Introduksjon_til_Microkontrollere.pdf

Bøker:

Robotics: Modelling, Planning and Control
Av Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo. 2008.

Robot Builder's Cookbook: Build and Design Your Own Robots av Owen Bishop. 2010.

Samling og programmering av mobile roboter hjemme av Frederic Giamarchi. 2007.

Vedlegg

Program (en del av program drive.c).

Et program drive.c kopiert til Micfrosoft Word tar 41 sider. Derfor plasserte jeg her bare en del av programmet som beskriver robots bevegelse.

```
if(UserButtonPressed)
  {
    // Play start song and wait
    byteTx(CmdPlay);
    byteTx(START_SONG);
    delayAndUpdateSensors(2813);

    // Drive around until a button or unsafe condition is detected
    while(!(UserButtonPressed)
          && (!sensors[SenCliffL])
          && (!sensors[SenCliffFL])
          && (!sensors[SenCliffFR])
          && (!sensors[SenCliffR])
          && (!sensors[SenChAvailable])
      )
    {

      // Keep turning until the specified angle is reached
      if(turning)
      {
        if(backing_up)
        {
          if((-distance) > 5)
            backing_up = 0;
          drive(-200, RadStraight);
        }
```

```
  else
  {
    if(turn_dir)
    {
      if(angle > turn_angle)
        turning = 0;
      drive(200, RadCCW);
    }
    else
    {
      if((-angle) > turn_angle)
        turning = 0;
      drive(200, RadCW);
    }
  }
}
else if(sensors[SenBumpDrop] & BumpEither)  // Check for a bump
{
  // Set the turn parameters and reset the angle
  if(sensors[SenBumpDrop] & BumpLeft)
    turn_dir = 0;
  else
    turn_dir = 1;
  backing_up = 1;
  turning = 1;
  distance = 0;
  angle = 0;
  turn_angle = randomAngle();

  // Play the bump song
  byteTx(CmdPlay);
```

```
      byteTx(BUMP_SONG);
    }
    else
    {
      // Otherwise, drive straight
      drive(300, RadStraight);
    }
// Flash the leds in sequence
    if(++leds_cnt >= 10)
    {
      leds_cnt = 0;
      if(turning)
      {
        // Flash backward while turning
        if(leds_state == 0)
          leds_state = 4;
        else
          leds_state--;
      }
      else
      {
        if(leds_state == 4)
          leds_state = 0;
        else
          leds_state++;
      }

      if(leds_state == 0)
      {
        // robot Power LED Amber
        byteTx(CmdLeds);
```

```
  byteTx(0x00);
  byteTx(128);
  byteTx(255);
  LEDBothOff;
}
else if(leds_state == 1)
{
  // Play LED on
  byteTx(CmdLeds);
  byteTx(LEDPlay);
  byteTx(0);
  byteTx(0);
  LEDBothOff;
}
else if(leds_state == 2)
{
  // Advance LED on
  byteTx(CmdLeds);
  byteTx(LEDAdvance);
  byteTx(0);
  byteTx(0);
  LEDBothOff;
}
else if(leds_state == 3)
{
  // Robot LEDs off, CM left LED on
  byteTx(CmdLeds);
  byteTx(0x00);
  byteTx(0);
  byteTx(0);
  LED2On;
```

```
      LED1Off;
    }
    else if(leds_state == 4)
    {
      // Robot LEDs off, CM right LED on
      byteTx(CmdLeds);
      byteTx(0x00);
      byteTx(0);
      byteTx(0);
      LED1On;
      LED2Off;
    }
  }

  // wait a little more than one robot tick for sensors to update
  delayAndUpdateSensors(20);
}

// Stop driving
drive(0, RadStraight);
```