

Upravljanje robotima

Opšti stavovi o upravljanju robotima

- Da bi robotski sistem, opremljen senzorima i pokretačkim jedinicama, mogao ostvariti željeno kretanje sa krajnjim ciljem izvršenja postavljenog zadatka, potrebno je upravljačko djelovanje.
- Jedno od osnovnih svojstava robota, s obzirom da savremeno tržište zahtjeva brze odzive proizvođača i raznolikost proizvoda, jeste fleksibilnost, odnosno mogućnost brzog prilagođenja novim zadacima.
- Ovo svojstvo robota je omogućeno, u prvom redu, zahvaljujući upravljanju kompjuterom čiji se program relativno lako mijenja.
- Jedna od funkcija upravljačkog sistema jeste da omogući uspješnu primjenu robota na različitim zadacima.

Postoji mnoštvo upravljačkih strategija koje se primjenjuju zavisno od stepena složenosti zadatka koje robot obavlja.

Sa aspekta kompleksnosti upravljačkih tehnika, u robotici se razmatraju dva pristupa upravljanju i to:

- slobodno kretanje robota u operativnom prostoru I
- kretanje robota u operativnom prostoru uz postavljena ograničenja.

U prvom slučaju se definiše kretanje hvataljke. Izvođenjem planiranog kretanja robot će izvršiti i planirani zadatak, dok se u drugom slučaju, osim definisanja kretanja koje robot izvodi, postavljaju zahtjevi o iznosu kontaktne sile kojom hvataljka djeluje na okolinu. Drugim riječima, nastoji se upravljati interakcijom između robota i okoline (engl. hibrid control). Jasno je da je upravljanje u drugom slučaju mnogo kompleksnije i prelazi okvire ovog kursa.

Ulogu upravljačke jedinice robotskog sistema vrši kompjuter. Kompjuter je "mozak" robotskog sistema. On omogućuje da robot – spoj mehaničkog, energetskog i senzorskog sistema svrsishodno djeluje.

Zadatak upravljačke jedinice, u najjednostavnijem slučaju kada je potrebno pozicionirati hvataljku u određenu tačku manipulativnog prostora, sastoji se u izračunavanju komponenti vektora unutrašnjih (upravljanih) koordinata, odnosno rješavanju inverzne kinematike i genesanju upravljačkih signala koji djeluju na aktuatore zglobova radi izvođenja željenog kretanja.

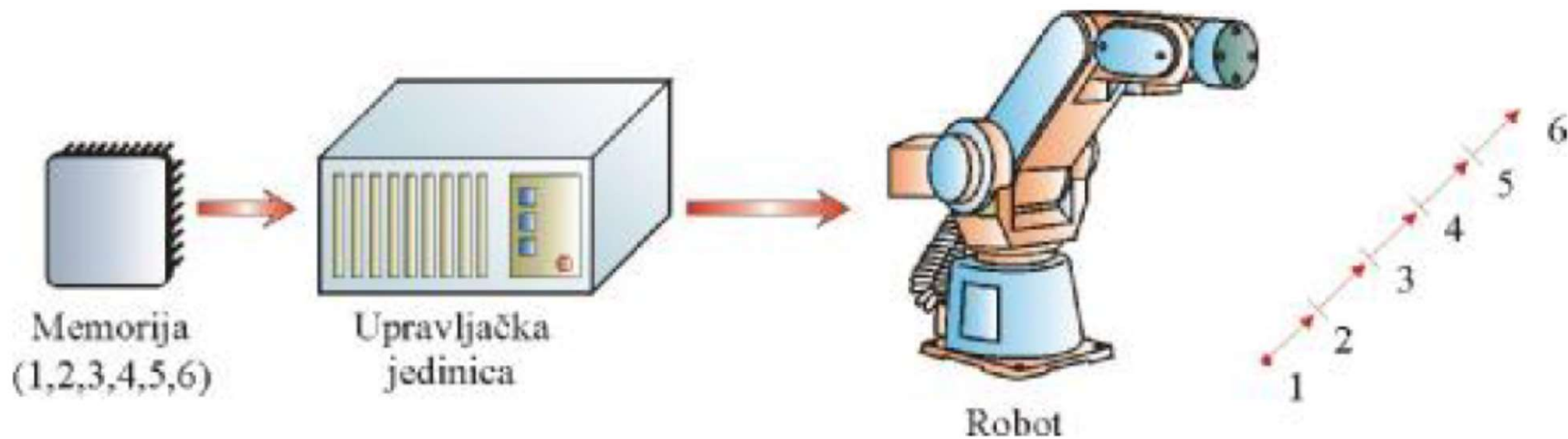
Kretanjima u pojedinim zglobovima za iznose izračunatih koordinata hvataljka (završni element) će dosegnuti željenu tačku manipulacijskog prostora.

Problem upravljanja znatno je složeniji ukoliko robot treba da slijedi planiranu trajektoriju. Trajektorija može biti definisana i u prostoru unutrašnjih i u prostoru spoljašnjih koordinata, pri čemu je definisanje u prostoru spoljašnjih koordinata znatno složenije, jer zahtijeva izračunavanje inverzne kinematike.

Ukoliko se sva potrebna izračunavanja obave prije nego li robot pristupi kretanju, robot radi u off-line režimu. U upravljačkoj jedinici robotskog sistema memoriše se trajektorija hvataljke definisana nizom tačaka koje se na njoj nalaze, odnosno iznosima upravljanih koordinata za određene tačke trajektorije. Pri izvođenju robotskog programa i kretanju robota nisu moguća nikakava izračunavanja.

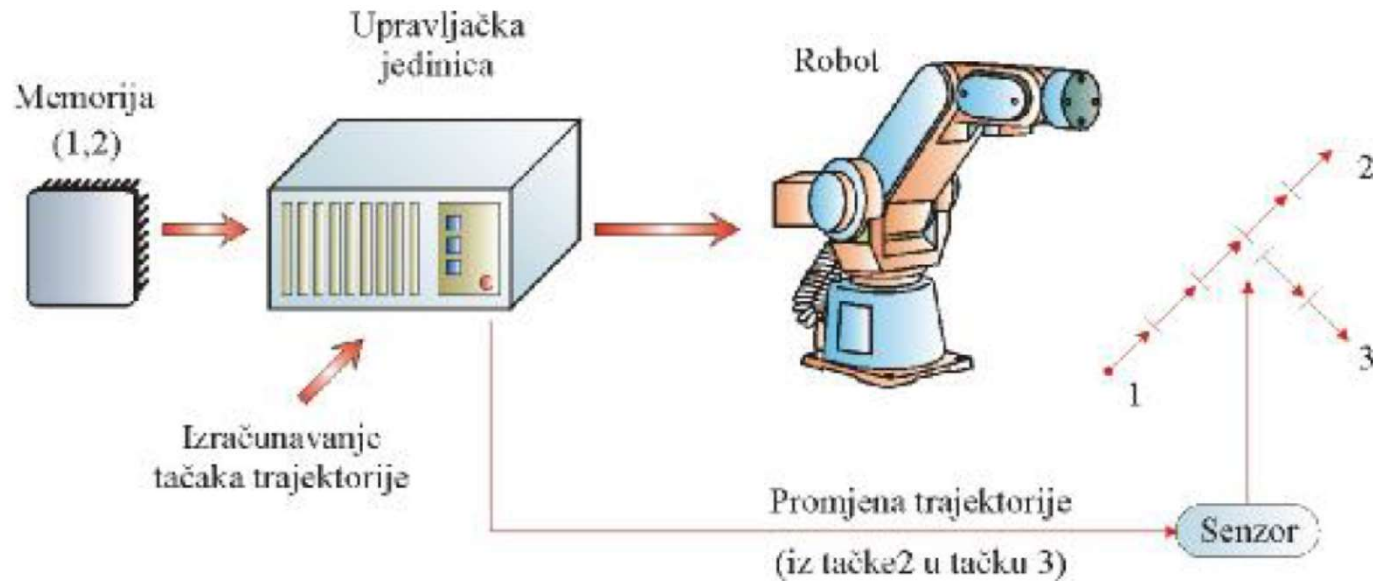
Upravo zbog toga, roboti sa ovakvim načinom rada ne mogu reagovati na promjenu u okolini – za takvo što potrebni su senzori. Rad u off-line ne zahtjeva složenu sensoriku kod robota. Roboti su sa "ograničenom inteligencijom". Kako su sva izračunavanja obavljena i unijeta u upravljački algoritam, robot ne može djelovati ni na kakvu promjenu u okolini, već će nastaviti sa realizacijom unešenog upravljačkog algoritma kao da se promjena nije ni desila. Od robota se ne zahtijeva bilo kakva komunikacija sa okolinom.

Off-line programsko upravljanje ne podrazumijeva brze kompleksne računare, pa su roboti sa ovakvim načinom rada manje skupi od onih sa radom u realnom vremenu. Na sl. 8.1. prikazan je off-line način rada robota kod kojeg se hvataljka kreće po pravoj liniji, definisanoj tačkama 1-6.



Slika 8.1. Off – line upravljanje robotom

Međutim, ukoliko se sva potrebna izračunavanja obavljaju u toku rada robota, za robot se kaže da radi u realnom vremenu ili u on-line režimu. Tokom djelovanja robota, njegova upravljačka jedinica mora, za nastavak kretanja, izračunavati upravljane koordinate slijedeće tačke duž željene trajektorije. Informacije dobijene od senzora o promjenama nastalim u okruženju moraju se obraditi tokom kretanja robota u realnom vremenu.



Slika 8.2. On – line upravljanje robotom

Na sl. 8.2. pokazan je on-line način rada robota, gdje je kretanje u predviđenu tačku 2, zamijenjeno kretanjem u tačku 3. Upravljačka jedinica čita iz memorije koordinate tačaka 1 i 2, zatim izračunava koordinate potrebnih međutačaka koje formiraju pravolinijsku trajektoriju. Ukoliko, međutim, upravljačka jedinica primi sa senzora instrukciju o promjeni ciljne tačke 2 u tačku 3, ona može izvršiti izračunavanje koordinata novih međutačaka, kako bi završni element nastavio kretanje po pravolinijskoj trajektoriji u novu ciljnu tačku 3.

Rad robota u on-line režimu, sa tehničke strane je mnogo zahtjevniji, podrazumijeva vrlo brzu, sofisticiranu upravljačku jedinicu, jer pretpostavlja da se sva izračunavanja obavljaju tokom rada robota, tako da robot u svakom trenutku raspolaže sa potrebnim informacijama za izvođenje planiranog zadatka, npr. raspolaže sa iznosima upravljanih koordinata, kako bi se hvataljka locirala u željenu tačku manipulacionog prostora.

Ukoliko ma koja informacija nije na raspolaganju robotu u pravom trenutku, funkcionisanje robota biće dovedeno u pitanje, što može izazvati ozbiljne posledice. Međutim i pored potrebe za složenijom upravljačkom jedinicom i složenijim upravljačkim algoritmom rad robota u realnom vremenu ima prednosti koje se posebno ogledaju u znatno većoj fleksibilnosti, odnosno sposobnosti robota da mijenja trajektoriju tokom izvođenja radnog zadatka. Ovo posebno dolazi do izražaja pri radu u „neorganizovanim“ sredinama, kada robot na osnovu informacija prikupljenih sa senzora ima mogućnost donošenja logičkih odluka.

Pojam i nivoi upravljanja

Tokom priče o pogonskim sistemima definisali smo upravljačke promenljive pojedinih vrsta pogona. U slučaju elektromotora jednosmerne struje u pitanju je napon na motoru, a u slučaju elektrohidrauličnog pogona, struja servorazvodnika.

Sada zadatak upravljanja možemo definisati na sledeći način: Obezbijediti takvu promjenu upravljačkih veličina koja će proizvesti zadato kretanje u zglobovima robota. Dakle, zadatak se svodi na zadato pokretanje zglobova.

Ovakva definicija, međutim, često nas ne može zadovoljiti. Radi se o tome da robot treba da vrši takozvano funkcionalno kretanje. Kako se funkcionalno kretanje, po pravilu, vezuje za završni uređaj robota, to zadatak upravljanja treba unekoliko preformulisati: Potrebno je obezbijediti takvu promjenu upravljačkih promenljivih koja će proizvesti traženi funkcionalni pokret, tj. traženo kretanje završnog uređaja u prostoru. Svakako, ovako formulisan upravljački problem uključuje i prethodnu definiciju. Naime, funkcionalni pokret treba raspodijeliti na zglobove, a zatim zglobove pokrenuti.

Problem upravljanja robotom možemo postaviti još opštije. Umjesto da zadatak formulišemo preko određenog kretanja, možemo napraviti kvalitativni skok i zadatak formulisati u vidu zahtjeva da robot izvrši neku složenu praktičnu operaciju. Na primjer, zadatak može biti: zavrnuti zavrtnj u predviđeni otvor.

Ovakav zadatak sadrži niz funkcionalnih pokreta. Prvi pokret -hvatanja zavrtnja, drugi prinošenje zavrtnja otvoru, treći zavrtnje i četvrti povratak u polazni položaj. Zadatak bi mogao biti još složeniji ako je, na primjer, potrebno sastaviti neki sklop od više djelova. Tada bi i broj elementarnih funkcionalnih pokreta bio znatno veći. Konačno, veoma čest problem ovog tipa je sakupljanje predmeta rasutih po podlozi.

Opisani način postavljanja zadatka predstavlja kvalitativni skok u odnosu na ranije zahtjeve.

Zadatak više nije kinematički orijentisan (zadato kretanje) već problemski orijentisan (izvršenje određene radnje). Problem koji treba riješiti robot prvo raščlanjuje na elementarne zahvate, tj. na niz elementarnih funkcionalnih pokreta, a zatim ih izvršava. Očigledno, ovo raščlanjavanje problema i utvrđivanje redosleda elementarnih radnji zahtijeva određenu "inteligenciju" kao i određene informacije tj. odgovarajuća čula. Sada zadatak upravljanja može da se formuliše na sledeći način: analizirati, a zatim izvršiti traženu radnu operaciju.

U prethodnoj diskusiji podrazumijevali smo rad u potpuno poznatim uslovima. Tu mislimo na precizno definisan radni prostor i radne operacije. Sledeće uopštenje predstavlja uvođenje veće doze neizvjesnosti. Radi se, često, o pojavi prepreka u radnom prostoru. Izvršenje zadatka tada podrazumijeva i stalno ispitivanje prostora oko robota i odlučivanje o tome kako u konkretnom slučaju postupiti da bi se izvršio zadatak. Formulacija zadatka upravljanja, ipak, ostaje ista kao maloprije.

Na kraju, možemo zamisliti i naredno uopštenje: ne mora biti zadata čak ni radna operacija koju treba izvršiti. Postavimo, na primer, ovakav zadatak: Ispitati dati uređaj (ili sklop), naći kvar i otkloniti ga.

U ovom slučaju robot će tek nakon analize zaključiti koje radne operacije treba izvršiti (npr. zamijeniti neki dio).

Ovo uopštenje bi predstavljalo novi kvalitativni skok jer zadatak postaje orijentisan ka cilju i problem upravljanja formulišemo na odgovarajući način: izvršiti operacije potrebne da bi se postigao traženi cilj.

Nivoi upravljanja

- Zadato pokretanje zglobova
- Funkcionalno kretanje (pokret) završnog uređaja u prostoru
- Izvođenje radne operacije koju čini niz funkcionalnih pokreta
- Analizirati, a zatim izvršiti traženu radnu operaciju

Prethodni nivoi odnose se na rad u potpuno poznatim uslovima u radnom prostoru (precizno definisan radni prostor i radni zadatak)

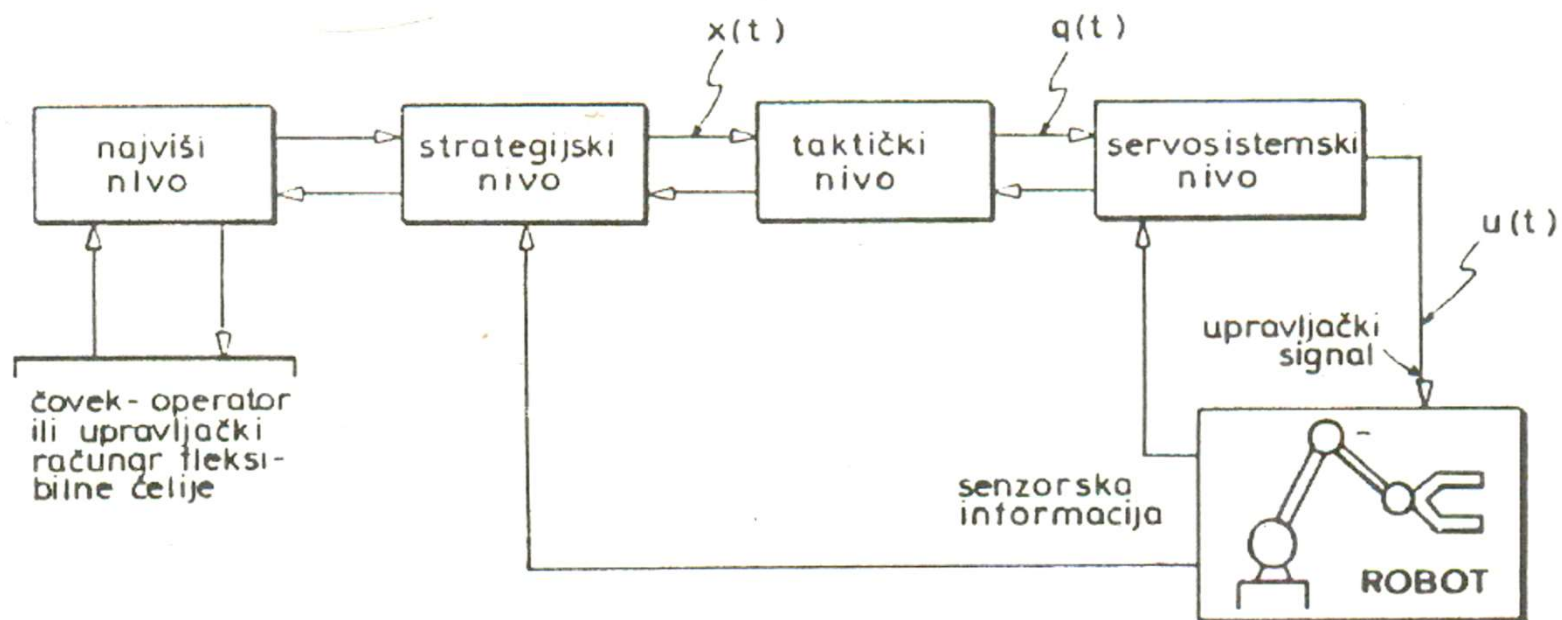
- kretanje robota u radnom prostoru uz ograničenja (izvršiti operacije potrebne da bi se dostigao željeni cilj)

Sledeći opisanu logiku uopštavanja zadatka koji se robotu postavlja dolazimo do upravljanja u više nivoa (sl.8.1.) pri čemu svaki viši nivo priprema zadatak i upravlja radom nižeg nivoa. Takođe, svaki nivo će, u zavisnosti od potrebe, raspolagati određenim senzorskim informacijama.

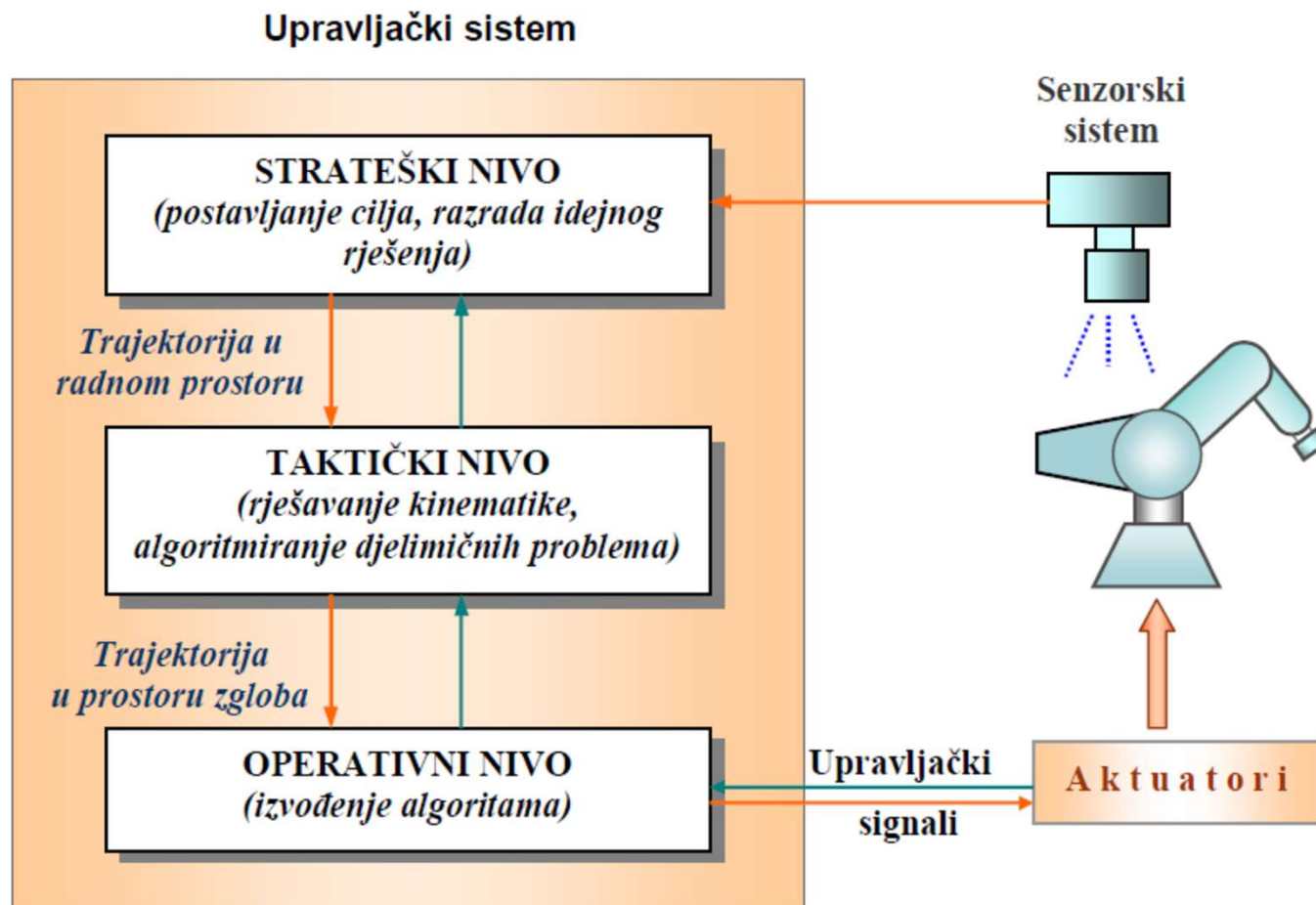
Servosistemski nivo predstavlja najniži nivo upravljanja i on neposredno izvršava kretanje. Zato se, često, i naziva izvršni nivo. Zadatak mora biti u obliku zahtjeva za određenim kretanjem zglobova: $q(t)$, gde je q vektor unurašnjih koordinata. Servosistemi u svakom zglobu obezbjeđuju izvršavanje traženog kretanja.

Ovaj upravljački nivo prima zadatak od višeg nivoa ili pak, neposredno od čovjeka - operatora ako viši nivo ne postoji.

Od mogućih senzorskih informacija servosistemski nivo koristi podatke o položaju i brzini pomjeranja zglobova.



Sl. 8.1. Nivoi upravljačkog sistema



Taktički nivo je prvi viši nivo upravljanja i on vrši raspodjelu kretanja na podsisteme zglobova. Zadatak se prima od višeg nivoa ili čovjeka - operatora neposredno i to u obliku zahtjeva za izvršenje određenog funkcionalnog pokreta $X(t)$, gde je X vektor spoljašnjih koordinata. Na ovom nivou rješava se inverzni zadatak kinematike čime se nalaze kretanje zglobova $q(t)$. Taktički nivo u principu ne zahtijeva dopunske senzorske informacije.

Strategijski nivo. Na ovom upravljačkom nivou, problemski orijentisan zadatak (formulisan u obliku zahtjeva za izvršenja određene radne operacije) raščlanjuje se na elementarne funkcionalne pokrete $X(t)$. Pri tome je neophodno izvršiti i planiranje kretanja koje nekada uključuje i različite vrste optimizacije koja omogućava da se raščlanjavanje izvrši na jednoznačan način. Na primjer: treba po nekom kriterijumu optimizirati redosled sakupljanja rasutih predmeta. Strategijski nivo često uključuje vizuelne sisteme, daljinare i sl. Problem obilaženja prepreka u radnom prostoru može se rešavati na strategijskom nivou ili ga prepustiti sledećem višem nivou.

Najviši nivo upravljanja prima zadatak orijentisan ka cilju, analizira ga, i formuliše radne operacije potrebne za njegovo postizanje.

Na kraju ove diskusije treba naglasiti da se većina današnjih robota zadržava na taktičkom nivou, mada se intenzivno razvijaju metode vještačke inteligencije koje omogućavaju realizaciju strategijskog nivoa.

Ovakvi inteligentni sistemi, mada postoje, još su srazmjerno rijetki u odnosu na prostije robotske sisteme.

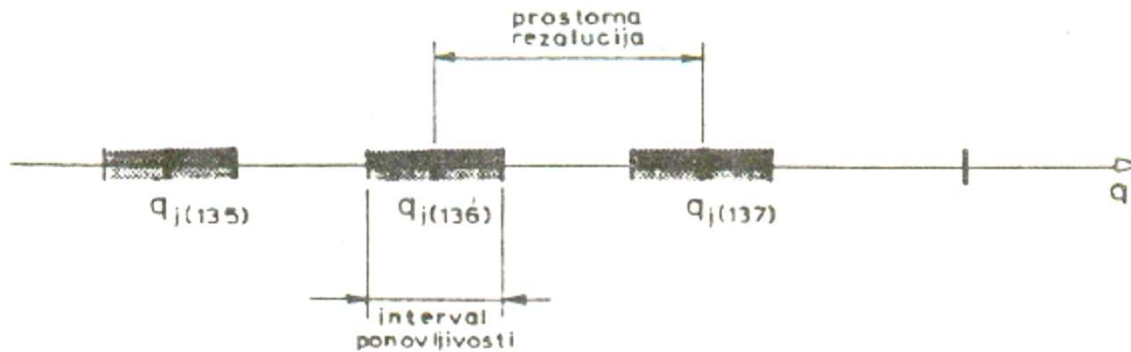
U nastavku ćemo obrađivati prvenstveno najniži (servosistemski) nivo upravljanja. Naime, inverzni problem kinematike (traktički nivo) razmatran je ranije, a Viši nivoi se zasnivaju na metodama vještačke inteligencije, koje prevazilaze obim ovoga kursa.

Preciznost kretanja

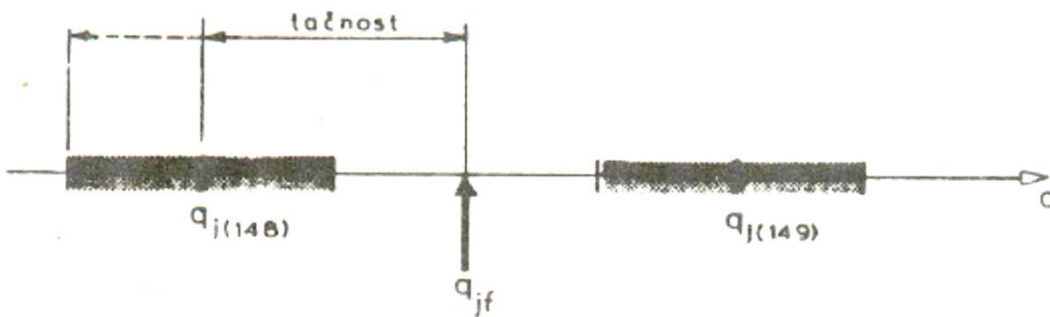
Preciznost kretanja robota je jedan od ključnih faktor uspješnog rada. Da bismo ovaj pojam bolje objasnili razmotrićemo tri veličine kojima se on u robotici opisuje: prostornu rezoluciju, tačnost pozicioniranja i ponovljivost. Sve ove veličine definisaćemo u odnosu na zadatak postizanja zadatog položaja robota u prostoru.

Prostorna rezolucija je najmanji pomjeraj koji robot može izvršiti po nalogu upravljačkog sistema. To nam ukazuje da robot nema mogućnost kontinualnog pozicioniranja, već postoji skup diskretnih položaja u radnom prostoru u koje robot možemo dovesti. Ova diskretizacija posledica je digitalizacije podataka o položaju u memoriji upravljačkog sistema. Na primjer, ako se jedna unutrašnja koordinata pamti u obliku 8 - bitne informacije, to znači da će ta koordinata imati 2^8 - 256 diskretnih položaja. Na slici 8.2a predstavljen je jedan deo diskretizovane koordinate

Preciznost kretanja



a)



b)

Rezolucija je karakteristika određena konstrukcijom upravljačkog sistema i u osnovi zavisi od mjernog sistema puta u povratnoj sprezi.

Razlikujemo dvije vrste rezolucije i to upravljačku i programsku:

Sl. 8.2. Prostorna rezolucija, ponovljivost i tačnost

Upravljačka rezolucija je najmanji inkrement, odnosno veličina kretanja, koja se može zadati programom. Naziva se još i osnovna jedinica rezolucije (BRU - Basic Resolution Unit).

Obično je 0.1° za obrtne ose ili 0.25 mm (0.01 Inch) za translatorne ose;

Upravljačka rezolucija je najmanja promjena, koju mjerni sistem puta u povratnoj sprezi može registrovati.

U poslednje vrijeme, upravljačke i programske rezolucije su jednake, i tada se nazivaju rezolucijom sistema.

Tačnost robota zavisi od tačnosti mehaničkog dijela, upravljačkih algoritama i rezolucije sistema:

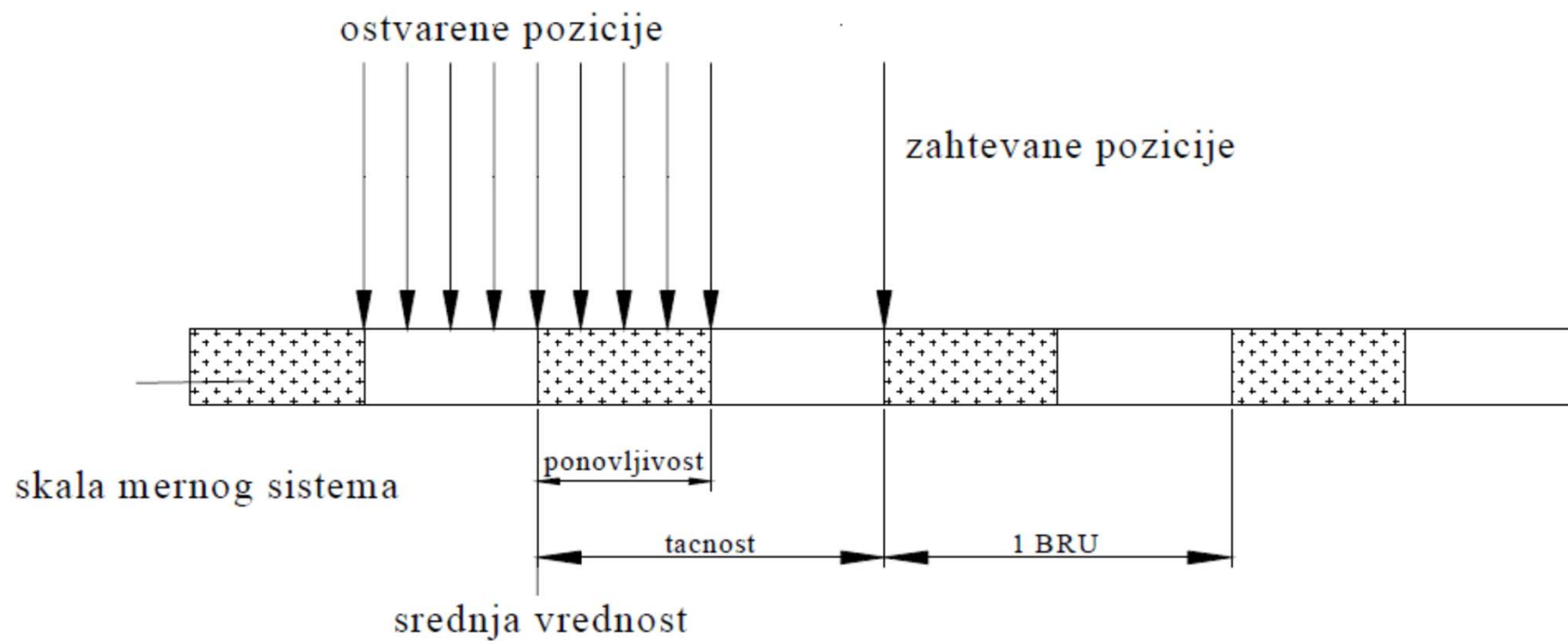
- tačnost mehaničkog dijela je uzrokovana netačnošću izrade djelova, zazorima, deformacijama, itd.;
- upravljački algoritmi uglavnom obuhvataju zaokruživanja pri sračunavanjima, koja se mogu uzeti u obzir;
- sistemska netačnost je uzrokovana rezolucijom sistema i obično iznosi $0.5 \cdot \text{BRU}$.

Tačnost robota se može izraziti na sledeći način:

Tačnost robota = Tačnost mehaničkog dijela + $0.5 \cdot \text{BRU}$

Idealno bi bilo da tačnost mehaničkog dijela bude manja od $0.5 \cdot \text{BRU}$, što bi dalo tačnost od $1 \cdot \text{BRU}$.

Ponovljivost ili tačnost ponavljanja je statistička veličina vezana za tačnost robota. Ako se robot pod istim uslovima više puta dovodi u zadatu poziciju, uočićemo različite položaje. Ponovljivost je odstupanje (devijacija) od srednje vrijednosti ostvarenih pozicija. Ona je značajno bolja od tačnosti, pa je proizvođači robota uvek daju kao karakteristiku.



Dopunski problem javlja se usled mehaničkih efekata koji doprinose smanjenju mogućnosti tačnog pozicioniranja. Jedan od takvih efekata je zazor u prenosnom sistemu, drugi je elastična deformacija elemenata prenosnog sistema, a postoji još niz faktora koji pojačavaju ove efekte.

Tako dolazimo do situacije da posmatrani zglob q , ne možemo pozicionirati tačno u neku od diskretnih tačaka već će se koordinata naći negdje u šrafiranoj okolini diskretizacionih tačaka (sl. 8.2a). Granice šrafiranog intervala određuju se po metodi najgoreg slučaja uzimajući u obzir zbirni uticaj pomenutih mehaničkih faktora.

Ponovljivost opisuje sposobnost robota da ostvari položaj koji mu odredi upravljački sistem. Jasno je da to može biti samo neka od diskretizacionih tačaka. Kako smo već naglasili, usled mehaničkih nepreciznosti robot će doći u neki od položaja unutar šrafiranog intervala, pri čemu će pri svakom novom dolasku zauzeti drugi položaj iz intervala. Zato šrafirani interval na sl. 8.2a nazivamo interval ponovljivosti. Eksperimentalno je moguće utvrditi raspodjelu vjerovatnoće zauzimanja pojedinih tačaka u intervalu ponovljivosti.

Tačnost pozicioniranja.

Do sada smo razmatrali mogućnost robota da zauzme različite položaje u prostoru uzimajući u obzir diskretizaciju radnog prostora. Međutim, u konkretnom zadatku, od robota se zahtijeva da dođe u položaj koji se u opštem slučaju ne poklapa sa nekim od diskretizacionih položaja.

Primjer za koordinatu q_j prikazan je na slici 8.2b. traženi položaj q_{jf} nalazi se između dvije diskretizacione tačke koje upravljački sistem može definisati. Tačnost pozicioniranja predstavlja odstupanje koordinate od q_{jf} . Očigledno da je gornja granica ovog odstupanja jednaka polovini prostorne rezolucije uvećane za poluinterval mehaničke nepreciznosti.

Tipovi upravljanja

- **Upravljanje od tačke do tačke** (engl. point-to-point control) podrazumijeva da se robotu zadaje niz različitih položaja, te on mora redom da dođe u svaki od njih. Pri tome, način kretanja između dva zadata položaja nije bitan. Uprošćeno, zadatak se svodi na to da svaki zglobov robota ("j"), polazeći od proizvoljnog položaja, postigne zadatu poziciju \mathbf{q}_{jf} sa odgovarajućom tačnošću. Ovakvim načinom upravljanja mogu se riješiti mnogi zadaci u industrijskoj primjeni robota, na primjer tačkasto zavarivanje, prenošenje materijala i sl.
- **Upravljanje kontinualnim kretanjem** (engl. continuous path control) podrazumijeva da robot prati zadatu putanju u prostoru uz zadatu promjenu brzine duž putanje. Svedeno na servosistemski nivo, zadatak formulišemo u vidu zahtjeva da svaki zglobov ("j") prati, sa određenom tačnošću, zadatu (nominalnu) promjenu \mathbf{q}_{jnom} . Ovakav način upravljanja neophodan je za izvršavanje zadataka kao što su: šavno zavarivanje, farbanje, pisanje i sl.
- S obzirom na to da je ovaj drugi tip upravljanja opštiji problem, mi ćemo se pretežno koncentrisati na njegovo rješenje. U principu, upravljanje od tačke do tačke možemo posmatrati kao upravljanje kontinualnim kretanjem, pri čemu je zadata nominalna putanja jednaka konstanti tj. $\mathbf{q}_{jnom}(\mathbf{t}) = \mathbf{q}_{jf}$, a početni položaj posmatramo kao početno odstupanje od zadate putanje.

Upravljanje raspregnutim sistemom

Kada govorimo o upravljanju robotom kao raspregnutim sistemom, mislimo na to da se svakim zglobovima robota upravlja kao izolovanim dinamičkim sistemom, dakle zanemarujući dinamički uticaj kretanja jednog zgloba na kretanje drugog. Razlog za ovo leži u složenosti proračuna dinamičkog sprežavanja i težnji ka jednostavnosti upravljanja. Razmotrićemo mogućnosti ovakvog raspregavanja sistema.

Dinamika motora j-tog zgloba $S_{Aj} :$ $\dot{x}_j = C_j x_j + f_j P_{Mj} + d_j u_j$

$$P_{M1} = \frac{H_{11}(\theta)}{N_1^2} \ddot{\theta}_1 + \frac{H_{12}(\theta)}{N_1 N_2} \ddot{\theta}_2 + \dots + \frac{H_{1n}(\theta)}{N_1 N_n} \ddot{\theta}_n + \frac{h_1(\theta, \dot{\theta})}{N_1}$$

Dinamika mehanizma

$$\vdots$$
$$P_{Mn} = \frac{H_{n1}(\theta)}{N_n N_1} \ddot{\theta}_1 + \frac{H_{n2}(\theta)}{N_n N_2} \ddot{\theta}_2 + \dots + \frac{H_{nn}(\theta)}{N_n^2} \ddot{\theta}_n + \frac{h_n(\theta, \dot{\theta})}{N_n}$$

Upravljanje raspregnutim sistemom

Dinamika motora j-tog zgloba

$S_{A_j} :$

$$\dot{x}_j = C_j x_j + f_j P_{M_j} + d_j u_j$$

Sprezanje S_{A_j} podsistema sa mehanizmom kroz moment opterećenja

$$P_{M_j} = \frac{H_{j1}(\theta)}{N_j N_1} \ddot{\theta}_1 + \dots + \frac{H_{jj}(\theta)}{N_j^2} \ddot{\theta}_j + \dots + \frac{H_{jn}(\theta)}{N_j N_n} \ddot{\theta}_n + \frac{h_j(\theta, \dot{\theta})}{N_j}$$

Raspregzanje možemo najjednostavnije izvršiti na sledeći način. Koeficijente H_{ji} , $j \neq i$ zanemarimo tj. proglasiti ih nulama. Koeficijent $H_{jj}(\theta)$ koji predstavlja funkciju svih koordinata θ_i , $i=1,2,\dots,n$ smatraćemo konstantom i usvojiti vrijednost \bar{H}_{jj} koju dobijamo nekom vrstom usrednjavanja ili maksimizacijom. Konačno, sabirak $h_j(\theta, \dot{\theta})$ takođe ćemo smatrati konstantom čija je vrijednost \bar{h}_j . Sada momenat opterećenja postaje

Upravljanje raspregnutim sistemom

$$P_{Mj} = \frac{\bar{H}_{jj}}{N_j^2} \ddot{\theta}_j + \frac{\bar{h}}{N_j}$$

Na ovaj način podsistem \mathbf{S}_{Aj} izolujemo od ostalih podsistema. Ukoliko \bar{H}_{jj}/N_j^2 dodamo inerciji rotora, tada model aktuatora postaje

$$\mathbf{S}_{Aj} : \quad \dot{\mathbf{x}}_j = \mathbf{C}_j \mathbf{x}_j + \mathbf{f}_j P_{Mj} + \mathbf{d}_j u_j$$

$$\mathbf{S}_{Aj} : \quad \dot{\mathbf{x}}_j = \mathbf{C}'_j \mathbf{x}_j + \mathbf{f}_j \frac{h}{N_j} + \mathbf{d}_j u_j$$

pri čemu \mathbf{C}' označava izmijenjenu matricu \mathbf{C} , usled uvećanja momenta inercije.

Upravljanje raspregnutim sistemom

Postavimo sada zahtjev da koordinata θ_j ostvari kretanje koje ćemo zvati nominalnim tj. $\theta_{jnom}(t)$. Takvom kretanju odgovaraće promjena brzine $\dot{\theta}_{jnom}(t)$.

Kako raspolažemo sensorima koji mere položaj θ_j i brzinu $\dot{\theta}_j$ to praćenje nominalnog kretanja realizujemo uvođenjem povratne sprege po koordinati θ_j i brzini $\dot{\theta}_j$. Upravljacki ulaz motora (u_j) sastojće se iz nominalne komponente u_{jnom} izračunate iz modela za zadato nominalno kretanje i komponente povratne sprege Δu_j . Komponenta u_{jnom} se često i izostavlja. Usvojimo da je Δu_j linearna funkcija odstupanja. Tada je:

$$\Delta u_j = -K_{jP}\Delta\theta_j - K_{jD}\Delta\dot{\theta}_j$$

gdje su:

$$\Delta\theta_j = \theta_j - \theta_{jnom}$$

$$\Delta\dot{\theta}_j = \dot{\theta}_j - \dot{\theta}_{jnom}$$

odstupanja položaja i brzine od nominalnih vrednosti, a K_{jP} i K_{jD} su pojačanja povratne sprege.

Upravljanje raspregnutim sistemom

Prvi sabirak u prethodnoj relaciji nazivamo, obično, pozicionom povratnom spregom, a drugi brzinskom ili diferencijalnom spregom. Ukupno, ovaj način upravljanja nazivamo P-D regulatorom.

U modelu aktuatora uočavamo i sabirak $\mathbf{f}_j \bar{\mathbf{h}}_j / \mathbf{N}_j$ koji predstavlja konstantni dio spoljašnjeg opterećenja. U takvom slučaju korisno je uvesti i integralnu povratnu spregu oblika $\mathbf{K}_{jI} \int \Delta \theta_j dt$, zato što ova sprega ostvaruje upravljački signal čak i kada greška $\Delta \theta_j$ padne na nulu. Tako dolazimo do P-D-I regulatora

$$\Delta u_j = -K_{jP} \Delta \theta_j - K_{jD} \Delta \dot{\theta}_j - K_{jI} \int \Delta \theta_j dt$$

Način određivanja pojačanja povratne sprege (\mathbf{K}_{jP} , \mathbf{K}_{jD} , \mathbf{K}_{jI}) koja će obezbijediti zadovoljavajuće praćenje nominalnog kretanja predstavlja zaseban problem. Moguće je koristiti bilo koju od standardnih metoda poznatih iz teorije automatskog upravljanja.

Dvoetajna sinteza upravljanja

- Glavni nedostatak prethodnog postupka je potpuno rasprezanje sistema što u nekim slučajevima može biti neprimjerno realnom sistemu. Naime, nekada je sprezanje podsistema znatno i ne smijee se na opisani način zanemariti. U ovom dijelu pokazaćemo pristup koji u znatno većoj mjeri vodi računa o sprezanju.
- Razdvojićemo sistem upravljanja na dvije etape
 - Etapu nominalnog kretanja (nominalne dinamike) i
 - Etapu poremećenog kretanja

Smatraćemo da je zadatak dat u vidu traženog nominalnog kretanja robota koje je izraženo promjenom unutrašnjih koordinata zglobova: $\mathbf{q}_{j\text{nom}} \dot{\mathbf{q}}_{j\text{nom}} \quad \mathbf{j}=1,\dots,\mathbf{n}$. Kako su pomjeranja motora direktno srazmjerna pomjeranjima zglobova, to pod nominalnim kretanjem podrazumijevamo i $\boldsymbol{\theta}_{j\text{nom}}(\mathbf{t}) \dot{\boldsymbol{\theta}}_{j\text{nom}}(\mathbf{t}), \quad \mathbf{j}=1,\dots,\mathbf{n}$. Ukoliko poznajemo dinamički model cijelog sistema ili kompaktni model, tada možemo izračunati upravljanja $\mathbf{u}_{1\text{nom}}(\mathbf{t}),\dots, \mathbf{u}_{n\text{nom}}(\mathbf{t})$ koja nazivamo nominalnim i koja će voditi sistem željenom putanjom. Ovaj proračun naziva se etapom nominalnog kretanja ili etapom nominalne dinamike.

Ukoliko robot izvršava neki zadatak u kome je kretanje $\boldsymbol{\theta}_{j\text{nom}}(\mathbf{t}), \quad \mathbf{j}=1,\dots,\mathbf{n}$ poznato prije početka izvršavanja, tada je moguće unaprijed riješiti nominalnu dinamiku i izračunato nominalno upravljanje $\mathbf{u}_{j\text{nom}}(\mathbf{t}), \quad \mathbf{j}=1,\dots, \mathbf{n}$ memorisati na nekoj od perifernih jedinica kako bi se moglo čitati i koristiti prilikom izvršenja kretanja. Ovo je, uglavnom, slučaj kod rutinskih industrijskih primjena robota kada se stalno ponavlja unaprijed dato kretanje. U nekom složenijem slučaju $\mathbf{q}_{j\text{nom}}(\mathbf{t})$ pa otuda i $\boldsymbol{\theta}_{j\text{nom}}(\mathbf{t})$ dobija se od taktickog nivoa ali u realnom vremenu, dakle u toku izvršenja. Tada i moramo računati u realnom vremenu, što podrazumijeva da raspolažemo algoritmima za rješavanje dinamike u realnom vremenu.

Dvoetajna sinteza upravljanja

Ako bismo na stvarni sistem primijenili samo nominalno upravljanje, tada bismo dobili sistem sa otvorenom spregom koji ne bi mogao pratiti željeno kretanje. To je otuda što takav sistem nema osobinu samokorekcije. U tom slučaju, nakon bilo kakvog poremećaja koji ga izvede sa željene putanje, sistem će "odlutati". Dopunski razlog je što je i nominalno upravljanje samo približno onome što je realnom sistemu potrebno. To je otuda što dinamički model nikada ne može uzeti u obzir sve karakteristike realnog sistema.

Kretanje realnog sistema nazivamo etapom poremećenog kretanja i u toj etapi uvodimo povratne sprege i određujemo odgovarajuće komponente upravljanja: $\Delta u_j, j=1, \dots, n$. Ukupno, upravljanje će biti oblika:

$$u_j = u_{jnom} + \Delta u_j, \quad j = 1, \dots, n,$$

Dvoetajna sinteza upravljanja

ili vektorski:

$$\mathbf{u} = \mathbf{u}_{\text{nom}} + \Delta \mathbf{u}$$

gdje se pojavljuju vektori upravljanja (kolona matrice dimenzije n).

Komponenta upravljanja usled povratne sprege ($\Delta \mathbf{u}$), u principu, je funkcija odstupanja $\Delta \theta$ i $\Delta \dot{\theta}$, gde je vektor koordinata položaja motora ($\theta = [\theta_1, \dots, \theta_n]^T$).

$\Delta \mathbf{u}$ može biti još i funkcija vremena.

Sada ćemo podsjetiti da je nominalno upravljanje sračunato na osnovu kompletnog modela dinamike, pa uzima u obzir sprezanje podsistema. Pri sintezi povratne sprege, na etapi poremećenog kretanja, možemo sistem raspregnuti, ili se, pak, i dalje držati spregnutog modela. Prvi pristup nazivaćemo decentralizovano upravljanje, a drugi pristup pokazaćemo na primjeru sinteze linearnog optimalnog regulatora.

Decentralizovano upravljanje

Ovaj pristup unekoliko slijedi logiku raspredanja izloženu ranije, međutim, ovde se radi o upravljanju koje na nivou nominala vodi računa o sprezanju, a kretanje oko nominala posmatra se raspregnuto.

Posmatraćemo model kompletne dinarnike kao skup podsistema motora koji su spregnuti posredstvom modela mehanizma. Kako nominalnu dinamiku smatramo riješenom, to nominalno kretanje i nominalno upravljanje smatramo poznatim: $\theta_{jnom}(t)$, $\dot{\theta}_{jnom}(t)$, $u_{jnom}(t)$. U tom slučaju modele možemo napisati u formi odstupanja od nominala. Tako za podsistem "j" dobijamo:

$$\Delta \dot{x}_j = C_j \Delta x_j + f_j \Delta P_{M_j} + d_j \Delta u_j$$

gdje je $\Delta x_j = x_j - x_{jnom}(t)$, $\Delta P_{M_j} = P_{M_j} - P_{M_jnom}$, $\Delta u_j = u_j - u_{jnom}$

a u slučaju modela drugog reda još i $\Delta x_j = [\Delta \theta_j, \Delta \dot{\theta}_j]^T$.

Sprezanje ovog podsistema sa drugim podsistemima, na nivou poremećaja, vrši se posredstvom momenta ΔP_{Mj} . Kada model napišemo u formi odstupanja, dobija se:

$$\Delta P_{Mj} = \sum_{k=1}^n \frac{H_{jk}^*(t, \Delta \theta)}{N_j N_k} \Delta \ddot{\theta}_k + \frac{h_j^*(t, \Delta \theta, \Delta \dot{\theta})}{N_j},$$

gdje je

$$H_{jk}^*(t, \Delta \theta) = H_{jk}(\theta)$$

$$h_j^*(t, \Delta \theta, \Delta \dot{\theta}) = (H_{jk}(\theta) - H_{jk}(\theta_{nom}(t)))\ddot{\theta}_{nom}(t) + h_j(\theta, \dot{\theta}) - h_j(\theta_{nom}(t), \dot{\theta}_{nom}(t)).$$

a θ označava vektor svih koordinata θ_j .

Uočimo da kod pisanja u formi odstupanja svaka funkcija položaja θ (npr. $F(\theta)$) postaje funkcija vremena t i odstupanja $\Delta\theta$ (npr. $F(t, \Delta\theta)$) zbog toga što položaj posmatramo u obliku $\theta = \theta_{nom}(t) + \Delta\theta$.

Prethodne jednačine za $j = 1, \dots, n$ određuju poremećajni model spregnutog sistema. Ukoliko želimo decentralizovano upravljanje, ovaj model je potrebno raspregnuti.

Prvo ćemo uočiti da se sprezanje podsistema "j" sa ostalim podsistemima izražava kroz momenat ΔP_{Mj} . U cilju rasrezanja, koeficijente H^*_{jk} , $j \neq k$ zanemarujemo. Zadržavamo samo H^*_{jj} i smatramo ga konstantom čija se vrijednost određuje usrednjavanjem ili maksimizacijom. Označimo tu vrijednost sa \bar{H}_{jj} .

Sabirak h^* možemo zanemariti ili smatrati linearnom funkcijom odstupanja $\Delta\theta_j$ što se može procijeniti prethodnom simulacijom. Ukoliko uvedemo \bar{H}_{jj} i zanemarimo h^*_j , dobijamo:

$$\Delta P_{Mj} = \frac{\bar{H}_{jj}}{N_j^2} \Delta \ddot{\theta}_j$$

čime smo raspregnuli sistem.

Podsistem aktuatora sada postaje:

$$\Delta \dot{x}_j = C'_j x_j + d_j \Delta u_j,$$

C' označava izmijenjenu matricu C , usled toga što je moment inercije rotora uvećan za \bar{H}_{jj}/N_j^2

Sinteza upravljanja Δu , za sistem dat gornjom relacijom sada se može izvršiti korišćenjem bilo koje od standardnih metoda (npr. metode postavljanja polova).
Određuju se pojačanja K_{jP} i K_{jD} i formira povratna sprega:

$$\Delta u_j = -K_{jP} \Delta \theta_j - K_{jD} \Delta \dot{\theta}_j$$

Ukupno upravljanje za podsistem "j" je: $u_j = u_{jnom} + \Delta u_j$

Očigledno je da se opisani način upravljanja razlikuje od pristupa izloženog u u prethodnom dijelu prvenstveno po tome što je sada nominalno upravljanje izračunato iz spregnutog modela, dok je u prethodnom odeljku nominal računat iz raspregnutog modela, ili čak nije ni uziman u obzir.

Decentralizovano upravljanje

Bez obzira na to što opisani postupak uzima u obzir sprezanje na nivou nominala, sintetizovano upravljanje \mathbf{u}_j ipak ne garantuje stabilnost realnog sistema u svim slučajevima. Kako je poremećeno kretanje posmatrano kao raspregnuto, to u slučajevima jakog sprezanja upravljanje može biti neodgovarajuće i čak dovesti do nestabilnosti. Možemo reci da su ovakvi slučajevi jakog sprezanja prilično rijetki, no ipak, nakon sinteze upravljanja, neophodno je proveriti stabilnost analizom nelinearnog poremećenog modela.

Upravljanje $\Delta \mathbf{u}$, sintetizovano na osnovu raspregnutog modela nazivamo lokalnom povratnom spregom. Ukoliko se na ovaj način stabilnost ne može postići, neophodno je uvesti dopunsku komponentu $\Delta \mathbf{u}_{jG}$ koju nazivamo globalnom povratnom spregom i koja vodi računa o uticaju sprezanja.

Linearni optimalni regulator

Ovaj pristup suštinski se razlikuje od prethodnog po tome što se i u etapi poremećenog kretanja posmatra spregnuti sistem.

Pošto sistem nećemo rasprezati, to ćemo se pri izvođenju regulatora koristiti modelom kompletne dinamike napisanim u kompaktnoj formi

$$\dot{\mathbf{x}} = \hat{\mathbf{c}}(\mathbf{x}) + \hat{\mathbf{D}}(\mathbf{x})\mathbf{u},$$

gde je \mathbf{c} vektor stanja cijelog sistema (dimenzija N), a \mathbf{u} vektor upravljanja (dimenzija n).

Ako kretanje razdvojimo na $\mathbf{X}_{\text{nominal}}(\mathbf{t})$ i odstupanje $\Delta\mathbf{x}$, tada model možemo napisati u formi:

$$\Delta\dot{\mathbf{x}} = \mathbf{C}^*(\mathbf{t}, \Delta\mathbf{x}) + \mathbf{D}^*(\mathbf{t}, \Delta\mathbf{x})\Delta\mathbf{u},$$

gdje je

$$\Delta \dot{\mathbf{x}} = \mathbf{x} - \mathbf{x}_{nom}(t), \quad \Delta \mathbf{u} = \mathbf{u} - \mathbf{u}_{nom}(t);$$

$$C^*(t, \Delta \mathbf{x}) = \hat{C}(\mathbf{x}) - \hat{C}(\mathbf{x}_{nom}(t)) + (\hat{D}(\mathbf{x}) - D(\mathbf{x}_{nom}(t)))\mathbf{u}_{nom}(t);$$

$$D^*(t, \Delta \mathbf{x}) = \hat{D}(\mathbf{x})$$

Početno odstupanje $\Delta \mathbf{x}(0)$ smatra se poznatim.

Izvođenje linearnog optimalnog regulatora zahteva niz složenih koraka (npr. linearizacija, rješavanje Rikatijske jednačine, i sl.). Kako to prevazilazi obim ovog kursa, to ćemo se na nekim mjestima ograničiti samo na postavku problema i interpretaciju rezultata.

Linearni optimalni regulator

U cilju sinteze regulatora potrebno je linearizovati poremećajni model.

$$\Delta \dot{\mathbf{x}} = \mathbf{C}^*(t, \Delta \mathbf{x}) + \mathbf{D}^*(t, \Delta \mathbf{x}) \Delta u,$$

Linearizacija se vrši oko nominalnog kretanja $\mathbf{x}_{\text{nom}}(\mathbf{t})$ tj. oko nule poremećaja $\Delta \mathbf{x}$.

Linearizacija se može izvršiti na različite načine: numeričkim metodama, simboličkim metodama ili, pak, metodama identifikacije. U postupke linearizacije ovde se nećemo upuštati. Naglasicćemo, međutim, da se u svakom slučaju linearizovani model dobija u obliku:

$$\Delta \dot{\mathbf{x}} = \bar{\mathbf{C}}(t) \Delta \mathbf{x} + \bar{\mathbf{D}}(t) \Delta u,$$

gde se matrice $\bar{\mathbf{C}}(\mathbf{t})$ i $\bar{\mathbf{D}}(\mathbf{t})$ dobijaju linearizacijom i funkcije su vremena. Dakle, dobijeni linearni model je nestacionaran. Ako bismo sintetizovali optimalni regulator za takav sistem, došli bismo do vremenski promjenljivih pojačanja povratne sprege.

Kako je to u principu nepoželjno, izvršićemo uprošćenje modela tako što ćemo ga usrednjiti tokom vremena, odnosno naći srednje vrijednosti matrica $\bar{C}(t)$ i $\bar{D}(t)$. Ako su srednje vrijednosti \bar{C} i \bar{D} , tada model postaje stacionarni :

$$\Delta \dot{x} = \bar{C} \Delta x + \bar{D} \Delta u.$$

Pretpostavimo sada da se sve koordinate stanja mjere što omogućava da se povratna sprega uvede po svim odstupanjima Δx . Tada sintezu regulatora vršimo na osnovu minimizacije kvadratnog kriterijuma.

$$J(\Delta u) = \int_0^{\infty} e^{2\gamma t} (\Delta x^T Q \Delta x + \Delta u^T R \Delta u) dt,$$

gde su Q i R pozitivno definitne težinske matrice odgovarajućih dimenzija (Q(NxN); R(n x n)), koje se, kao i stepen stabilnosti γ , biraju tako da se obezbijedi praktična stabilnost sistema.

Usvojeni postupak vodiće rješavnju Rikatijske jednačine. Ukoliko upravljanje pretpostavimo u obliku:

$$\Delta u = -R^{-1} \bar{D}^T K \Delta x,$$

tada je K matrica dimenzija $N \times N$ i predstavlja rešenje Rikatijske jednačine:

$$K(\bar{C} + \alpha I) + (\bar{C}^T + \alpha I)K - K\bar{D}R^{-1}\bar{D}^T K + Q = 0,$$

gde je I jedinična matrica odgovarajućih dimenzija.

Ukupno upravljanje sada je:

$$u = u_{nom} + \Delta u.$$

Prilikom sinteze linearnog optimalnog regulatora uveli smo niz uprošćenja (linearizacija, usrednjavanje). Zbog toga rešenje ipak ne garantuje stabilnost realnog sistema. Zato bi trebalo izvršiti analizu stabilnosti nelinearnog modela poremećaja.

Dopunska nepogodnost leži u pretpostavci o merljivosti svih koordinata stanja i uvođenju povratne sprege po svakoj od njih. Naime, u praksi se, po pravilu, uvode povratne sprege samo po položaju i brzini motora. Na primjer, ne uvodi se sprega po struji rotora iako je to jedna od koordinata stanja elektromotora (ako je opisan modelom trećeg reda).

DIGITALNA SHEMA UPRAVLJANJA

U prethodnom dijelu izveli smo postupak upravljanja zasnovan na rješenju nominalnog upravljanja i povratne sprege po položaju i brzini zglobova (q, \dot{q}). Preciznije rečeno, povratnu spregu smo uveli po položaju i brzini motora koji pokreću zglobove .

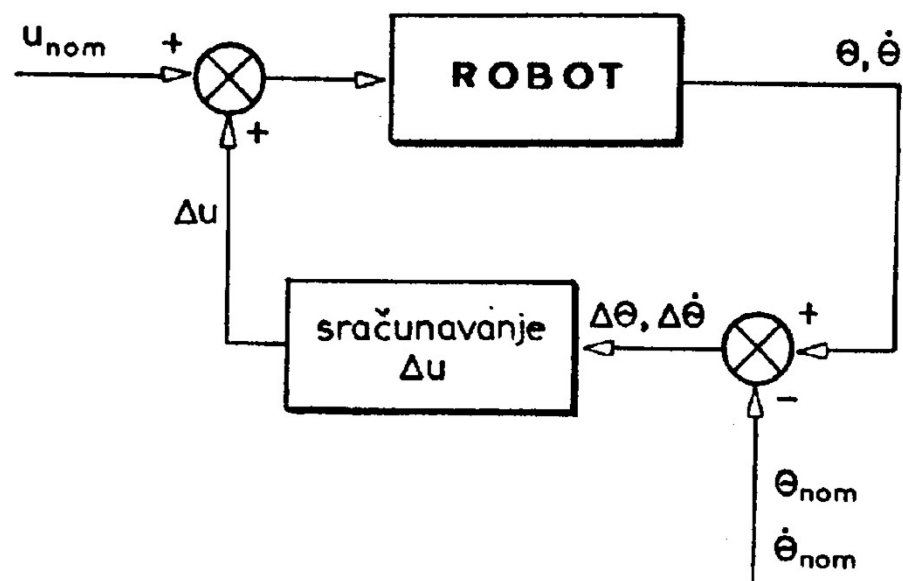
S obzirom na jednoznačnu i prostu vezu između koordinata motora i koordinata zglobova q , u principu je svejedno koja od ovih veličina će učestvovati u formiranju povratne sprege. Kako se u praktičnim realizacijama najčešće mjeri položaj i brzina motora, to smo u prethodnim odeljcima uveli odstupanje, pa i povratnu spregu po ovim veličinama ($\theta, \dot{\theta}$). Svakako, postoji i mogućnost mjerenja koordinate q umesto θ (npr. ako se koristi potencijometar).

Prethodno razmatranje ipak nije u potpunosti tačno. Radi se o tome da u nekim slučajevima moramo uzeti u obzir elastične deformacije u sistemu za prenos pogona, a tada koordinate θ i q postaju nezavisne. U takvom slučaju više nije svejedno da li se mjeri jedna ili druga veličina. Štaviše, u principu bi bilo poželjno mjeriti obje, međutim, to bi nas vodilo specijalnim tipovima regulatora o čemu ovdje nećemo govoriti.

Razmotrimo sada mogućnost praktične realizacije upravljačke sheme sa povratnom spregom prikazane na sl.8.3.

Blok za izracunavanje Δu na osnovu odstupanja $\Delta\theta$ i $\Delta\dot{\theta}$ može se izvesti na različite natine, o čemu je govoreno u prethodnim odeljcima. Mi ćemo usvojiti da je u pitanju linearna forma tj.

$$\Delta u = -K_P \Delta \theta - K_D \Delta \dot{\theta} - K_I \int \Delta \theta dt.$$

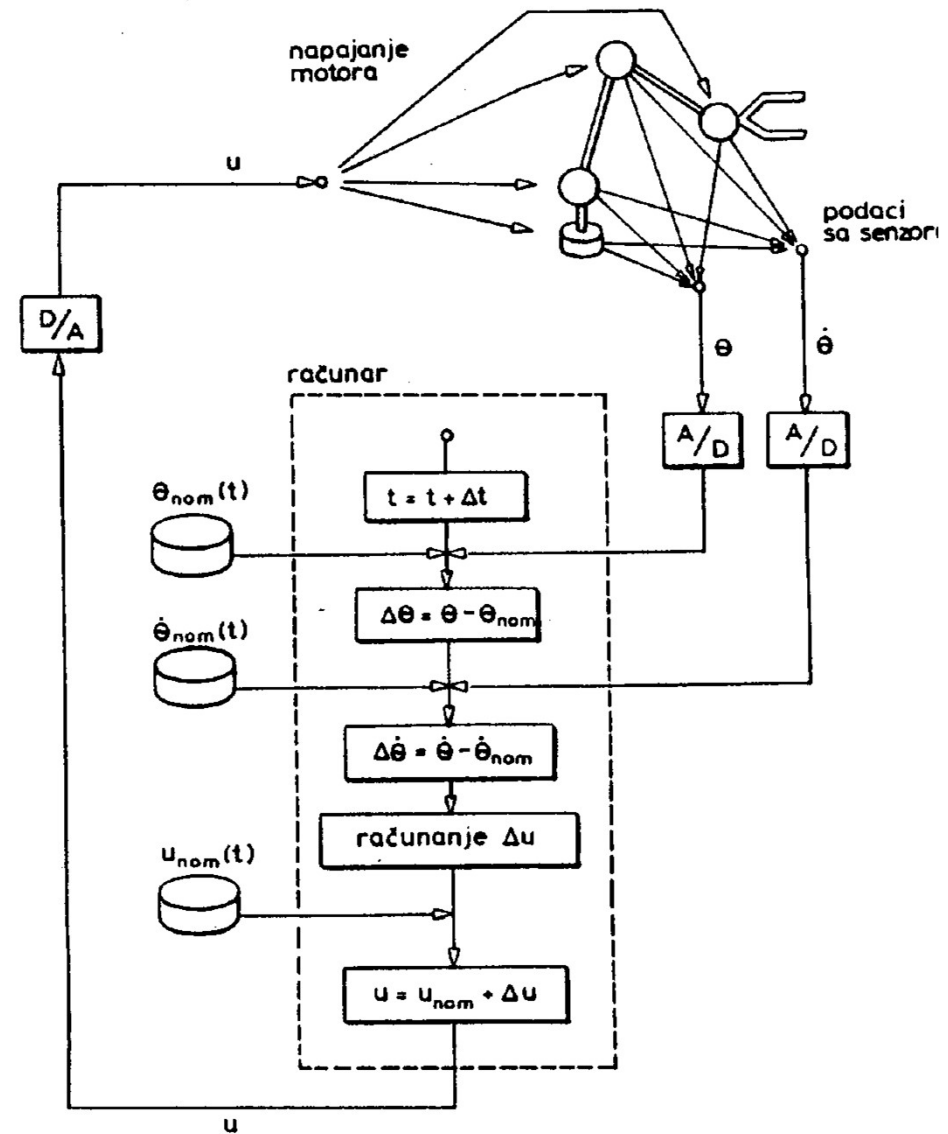


Sl. 8.3. Upravljanje sa povratnom spregom

Ukoliko ovu šemu realizujemo digitalno, dolazimo do detaljnije sheme koja je prikazana na Sl. 8.4. Pri formiranju ove sheme smatrano je da se nominal izračunava unaprijed, a u toku izvršenja zadatka čita se sa diska.

U nekom trenutku (t) senzori položaja mjere koordinate θ i $\dot{\theta}$. Nakon A/D konverzije podaci ulaze u upravljački računar.

Sada se sa diska čita vrijednost $\theta_{nom}(t)$ i $\dot{\theta}_{nom}(t)$ i izračunavaju greške $\Delta\theta$ i $\Delta\dot{\theta}$. Na osnovu tih greški formira se signal povratne sprege Δu . Nakon sabiranja sa nominalnim upravljanjem ($u = u_{nom} + \Delta u$) i D/A konverzije, dobijeni napon primjenjujemo na motore.



Sl. 8.4. Digitalna shema upravljanja

Od trenutka očitavanja senzora, pa do trenutka primjene upravljanja prošlo je izvesno vrijeme Δt . Dakle, upravljanje realizujemo ponavljajući opisani ciklus sa korakom Δt koji nazivamo vrijeme razdvajanja. Dok se ne izračuna nova vrijednost upravljanja primjenjuje se vrijednost izračunata u prethodnom koraku. Slijedi da upravljanje ima stalnu vrednost tokom Δt , a zatim se skokovito mijenja na novu izračunatu vrijednost. Naglasimo još da u robotskim sistemima interval razdvajanja ne bi trebalo da bude duži od 20ms.

Pri primjeni opisane sheme upravljanja uočavamo neminovno kašnjenje. Naime, upravljenje (u) koje odgovara stanju θ i $\dot{\theta}$ tj. trenutku t , biće izračunato i primijenije se na motore tek u trenutku $t + \Delta t$ i vodiće motore sve dok ne stigne nova vrijednost, a to je trenutak $t+2\Delta t$. Ovo kašnjenje pogotovo u slučaju suviše velikog Δt , može bitno uticati na kretanje robota. Ukoliko se pokaže potreba, ovo kašnjenje možemo korigovati postupkom predikcije (predviđanja).

Razmotrimo šta bi bilo neophodno uraditi u intervalu $(t, t + \Delta t)$ kako bi se korigovalo kašnjenje. Ako imamo u vidu da će upravljanje (u), izračunato u ovom intervalu, biti primijenjeno tek od trenutka $t + \Delta t$, onda zaključujemo da bi ono trebalo da odgovara tom trenutku, dakle $u(t+\Delta t)$. Kako je nominalna komponenta smještena na disku, to njenu vrednost $u_{nom}(t + \Delta t)$ možemo pročitati tokom intervala $(t, t + \Delta t)$. Međutim, povratna sprega $\Delta u(t + \Delta t)$ zahtijeva poznavanje greški $\Delta \theta(t + \Delta t)$, $\Delta \dot{\theta}(t + \Delta t)$, a to ne možemo izračunati budući da tokom intervala $(t, t + \Delta t)$ ne znamo položaj i brzinu koje će motori imati na kraju intervala (tj. $\theta(t + \Delta t)$ i $\dot{\theta}(t + \Delta t)$).

Da bismo opisanu ideju ipak sproveli izvršićemo predikciju i predvidjeti tražene vrednosti u trenutku $t + \Delta t$ na osnovu vrijednosti u trenutku t (tj. na osnovu $\theta(t)$, $\dot{\theta}(t)$ i niza prethodnih vrijednosti). Tako dobijamo prediktovane vrednosti $\theta_p(t)$, $\dot{\theta}_p(t)$ koje koristimo za izračunavanje greški:

$$\Delta \theta(t + \Delta t) = \theta_p - \theta_{nom}(t + \Delta t) \text{ i } \Delta \dot{\theta}(t + \Delta t) = \dot{\theta}_p - \dot{\theta}_{nom}(t + \Delta t).$$

Sam postupak predikcije ovde nećemo opisivati

Na slikama 8.3 i 8.4 uveli smo blok koji izračunava povratnu spregu Δu u zavisnosti od greški $\Delta\theta$ i $\Delta\dot{\theta}$. Pri tome smo rekli da se obično radi o linearnoj formi tj. P-D ili P-D-I regulatoru.

Sa stanovišta realizacije najpovoljnije je da pojačanja povratne sprege (K_p , K_D , K_I) budu konstantna. Međutim, u nekim slučajevima to nije moguće. Posmatrajmo, na primjer, zadatak u kome robot prenosi neke predmete sa jednog mjesta na drugo. U fazi nošenja robot (njegovi motori) je znatno više opterećen nego u fazi kada se prazan vraća. Dakle, mijenjaju se dinamički parametri, pa će jednoj fazi odgovarati jedne vrijednosti pojačanja, a drugoj fazi druge vrijednosti. Tako dolazimo do promjenljivih pojačanja koja mijenjaju vrijednost u određenim trenucima, ali su to poznate i zadate vrijednosti izračunate unaprijed na osnovu poznatih tereta koji će biti prenošeni.

Složeniji problem nastaje ako robot prenosi terete koji nijesu unaprijed poznati. Dakle, radi se o promjeni dinamičkih parametara u obimu koji nije zadat. U takvim slučajevima primjenjuju se postupci estimacije kojima se vrši procjena vrijednosti opterećenja, pa se na bazi toga određuju odgovarajuće vrijednosti pojačanja povratne sprege. Kako se u ovom slučaju robot sam prilagođava novom teretu tj. izmijenjenim dinamičkim parametrima, to govorimo o adaptaciji i adaptivnom upravljanju.

Međutim, ako promjena dinamičkih parametara nije velika, tada nije neophodno mijenjati vrijednosti pojačanja. Naime, pojačanja možemo već na početku odrediti tako da mogu odgovarati određenom intervalu promjene nekog dinamičkog parametra. Tada kažemo da je upravljanje robusno u odnosu na promjenu tog parametra.

Nemodelirana fleksibilnost

Jedna od napravljenih aproksimacija kod izvedenih upravljačkih struktura jeste zanemarivanje fleksibilnosti zupčastih prenosnika, ležajeva, osovina i članaka robota.

Ovi elementi su ograničene krutosti i, ukoliko se pri modeliranju uzme u obzir njihova fleksibilnost to će doprinijeti povećavanju reda sistema.

Osnovni razlog zanemarivanja fleksibilnosti pri modeliranju dinamike robotskog sistema jeste taj, da ako je sistem zadovoljavajuće krutosti, prirodne frekvencije nemodelirane rezonancije su vrlo visoke i mogu biti zanemarene u upoređenju sa uticajem dominantnih polova drugog reda.

Zanemarivanjem fleksibilnosti dinamički model robota se znatno pojednostavljuje, ali se mora voditi računa o pobuđenim frekvencijama. Kako i struktura ima svoju frekvenciju (strukturna frekvencija) pri kojoj nastaje njeno rezonantno oscilovanje, to se ne smije dozvoliti da prirodna frekvencija bude jednaka strukturnoj. U protivnom, nastupilo bi rezonantno oscilovanje cjelokupne strukture. Naime, ako je najniža strukturna frekvencija ω_{res} , tada se prirodna frekvencija zatvorene petlje mora ograničiti na

$$\omega_n \leq \frac{1}{2} \omega_{res} .$$

Ovo je moguće odabirom parametara regulatora, jer se sa povećavanjem faktora pojačanja smanjuju vrijeme odziva sistema i greška ustaljenog stanja, a s druge strane iznos faktora pojačanja određen je nemodeliranom strukturnom rezonancijom.

Servosistemi trebaju uvijek biti kritično ili nadkritično prigušeni. Najčešće industrijski roboti imaju nemodeliranu strukturnu rezonanciju u iznosu od 5 Hz do 25 Hz. Uz ispunjen gornji uslov, strukturna frekvencija ne može uzrokovati neželjeno oscilovanje.

Ukoliko strukturna fleksibilnost sistema može biti ustanovljena moguća je približna analiza rezultirajućih vibracija aproksimacijom robotskih segmenata jednostavnim sistemom “masa-opruga” gdje je prirodna frekvencija data jednačinom_:

$$\omega_n = \sqrt{\frac{k}{m}} ,$$

gdje su k krutost fleksibilnog segmenta, a m njegova ekvivalentna masa.

Uključivanje strukturne fleksibilnosti sistema u upravljački zakon potrebno je želi li se dostići zatvorena petlja propusn opseg veći od vrijednosti date izrazom

$$\omega_n \leq \frac{1}{2} \omega_{res} .$$

Rezultirajući model sistema biće višeg reda, a upravljačke tehnike takvih sistema mnogo sofisticiranije i predmetom su brojnih istraživanja.

Suština svih upravljačkih tehnika je da prate grešku između postignute i željene trajektorije, bez obzira da li se to odvija u povratnoj petlji ili direktnoj grani, jer dinamički model robota, ma koliko bio kompleksan, ipak ne odgovara realnoj dinamici. Ovo je posljedica, uglavnom neuključivanja svih efekata u dinamički model, kao što su Coulombovo trenje, labavost prenosnika, tolerancija dimenzija, kao i nedovoljno tačne pretpostavke, npr. potpuno kruti članci manipulatora i dr.

Tipovi programiranja

Postoji više načina programiranja rada robota, a ostvaruje se unosom programskih instrukcija u upravljačku jedinicu robotskog sistema. Važno je napomenuti da neki od načina programiranja zahtijevaju vrlo malo ili nikakvo znanje od korisnika što je bitno u industrijskoj praksi. U zavisnosti o načinu programiranja rada roboti se dijele na:

1. Robote sa graničnim prekidačima,
2. "Playback" robote,
3. Robote sa fazom učenja i off-line programiranjem
4. Robote sa strukturnim programiranjem i
5. Robote sa programiranjem zadatka.

Roboti sa graničnim prekidačima

Ovo su roboti prve generacije, sa upravljanjem tipa “tačka po tačka”. Kod njih se “programiranje” ostvaruje postavljanjem graničnika (električnih, pneumatskih prekidača i sl.) kojima se definiše kretanje po pojedinima osama.

Ne postoje servo sistemi, niti povratne petlje po poziciji i brzini. Senzorski sistem je izuzetno ograničen i sveden samo na nekoliko senzora unutrašnjeg stanja. Samim tim, roboti ovog tipa ne mogu donositi nikakve odluke.

Odlikuju se izuzetnom nefleksibilnošću, jer prelazak na izvršavanje drugog zadatka zahtijeva fizičku intervenciju operatera u smislu postavljanja novih graničnika za definisanje novog željenog kretanja. Najčešće su roboti ovog tipa tzv. pick and place roboti i to su uglavnom pneumatske izvedbe.

“Playback” roboti

“Playback” roboti su opremljeni digitalnom upravljačkom jedinicom u čiju se memoriju snimaju robotski programi. Oni ostaju sačuvani u memoriji dok se god program izvodi. Za robote ovog tipa svojstvena je faza učenja tokom koje robot pamti potrebna kretanja koje potom može potreban broj puta ponavljati, pa zato i naziv “playback” roboti. Za njih je svojstveno on-line programiranje. Faza učenja robota može se izvoditi:

- direktno i
- indirektno.

Kod direktnog učenja vješt operater - ekspert vodi završni uređaj robota kroz manipulacijski prostor duž željene trajektorije, odnosno vrši radnje kao kada bi i sam robot obavljao dati zadatak. Robot, koji je u modu za učenje, vrši periodično pamćenje tačaka (određeno periodom odabiranja) duž željene trajektorije, te stanja završnog uređaja (otvoreno / zatvoreno), kao i periodično generisanje programa.

Podaci se skladište u memoriju upravljačke jedinice i nakon što je završena faza učenje, robot se prebacuje u izvršni mod. Tada je robot u stanju da ponovi prethodno naučenu kretanju obavljajući pri tome i planirani zadatak. Brzina kojom se završni element kreće tokom faze učenja ne mora biti jednaka brzini rada robota.

U slučaju da robot ima velike gabarite i da je operateru teško voditi njegovu hvataljku po željenoj trajektoriji, robot se oprema manjim, zamjenskim robotom – pilotom, koji je iste strukture kao i veliki, a služi samo za izvođenje faze učenja. Operater, naime, vodi hvataljku manjeg robota kroz željene sekvence kretanja, svi podaci se periodično čuvaju u zajedničku upravljačku jedinicu, tako da je pravi robot u stanju reprodukovati željeno kretanje.

Ovdje ne postoji gotovo nikakva, ili je u pitanju vrlo mala potreba za intervencijom senzora u realnom vremenu tokom faze učenja.

Moguće je CP programiranje, pa se ovi roboti najčešće koriste za zavarivanje, farbanje i sl. Ne zahtijeva se od korisnika gotovo nikakvo znanje o robotima, niti kompleksno matematičko opisivanje trajektorije. Ovaj, stariji način “programiranja” robota ima niz nedostataka:

- Metod je neprimjeniv kada robot treba djelovati u okruženju opasnom po čovjeka.
- Brzina “programiranja” zavisi o vještini, osjećanjima, brzini rada operatera.
- Zbog prisustva čovjeka nekada je nemoguće sinhronizirati kretanja svih komponenti u radnom prostoru (npr. kada je robot dio fleksibilne proizvodne ćelije).
- Pristup je neefikasan sa aspekta korišćenja raspoložive memorije. Naime, iste sekvence kretanja nemoguće je ponavljati izvršavanjem već postojećeg dijela programa, već se moraju ponovo isprogramirati, što doprinosi stvaranju vrlo dugih programa.

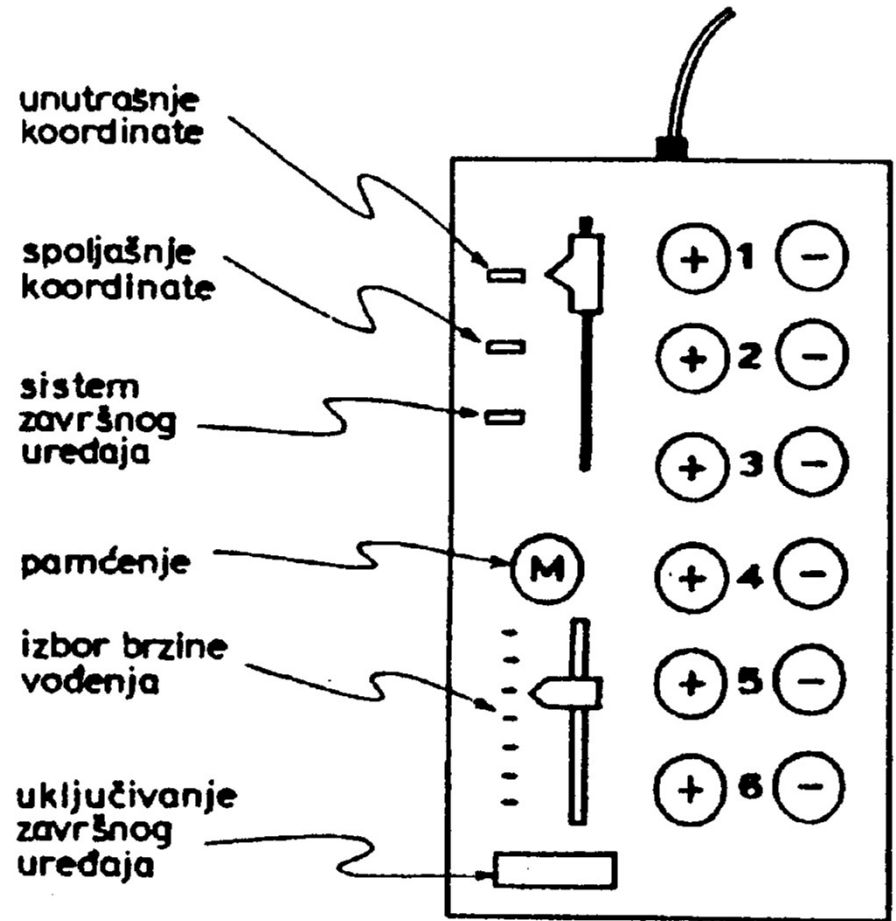
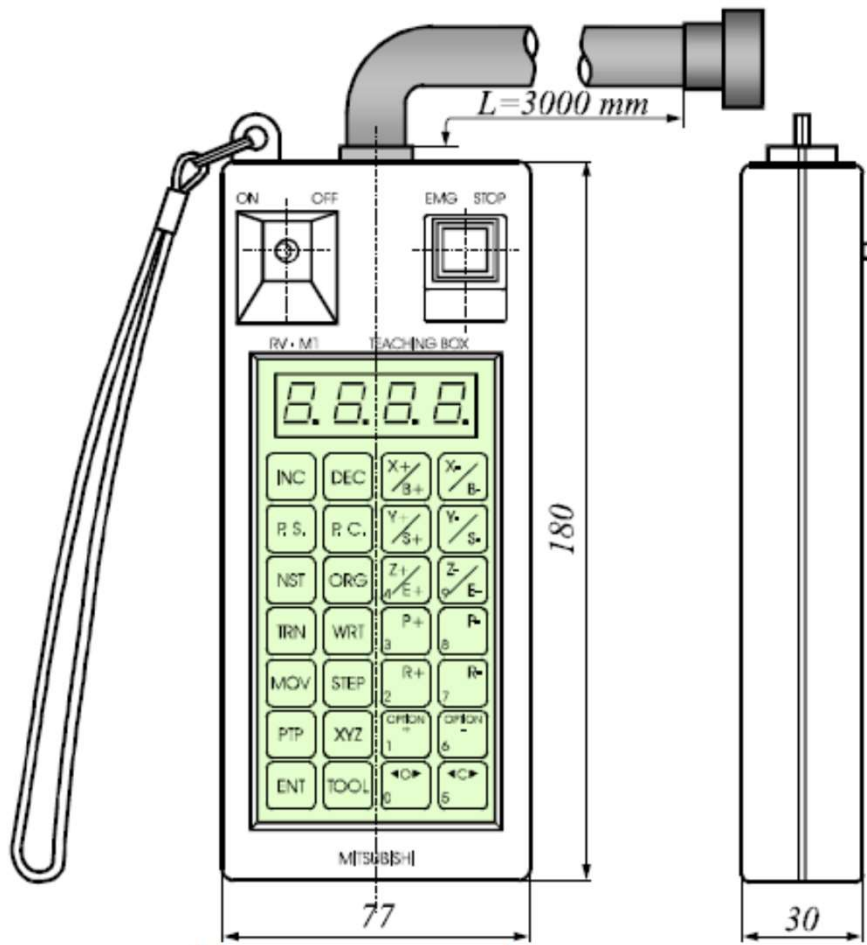
Kod indirektnog izvođenja faze učenja koristi se daljinski upravljač za učenje sa komandnom palicom (engl. teach pendant). Naime, kod ove faze učenja robot se prebacuje u mod za učenje, a operater daljinskim upravljanjem vodi hvataljku duž potrebne trajektorije.

Po dostizanju željene tačke, operater pomoću elementa za učenje, snima podatke sa unutrašnjih senzora pozicije zglobova u memoriju upravljačke jedinice. Snimljenim koordinatama se pridružuje određeni naziv, najčešće broj, tako da je tačka definisana brojem u memoriji upravljačke jedinice robotskog sistema. Pomoću elementa za učenje unose se i programske naredbe.

Za razliku od predhodnog pristupa, ovdje je moguće izvođenje faze učenja robota i u opasnim sredinama jer se učenje odvija daljinski, a direktno je primjenjivo kod teških i robusnih robota. Također, bolje je iskorišćenje raspoložive memorije jer, zavisno o tipu upravljačke jedinice programski može biti izveden uslovni ili bezuslovni skok na pojedine sekvence programa, što omogućuje izbjegavanje programiranja istog kretanja više puta. Nedostatak ove metode je odvratanje pažnje operatera od procesa, radi odabira odgovarajuće tipke na upravljaču. Jedan od nedostataka, u oba slučaja, kod "playback" robota je neproduktivnost robota tokom faze učenja, dok je, s druge strane, način definisanja radnog zadatka vrlo jednostavan.

Roboti sa fazom učenja i off-line programiranjem

Kod robota sa off-line programiranjem definisanje koordinata tačaka planirane trajektorije se izvodi pomoću priveska za učenje (engl. teaching box), dok programe pišu programeri koristeći se robotskim programskim jezicima. Ovdje se radi o upravljanju tipa “tačka po tačka”. Na privesku za učenje nalazi se displej za prikazivanje broja tačke manipulcijskog prostora tokom faze učenja, dok će tokom faze izvođenja programa biti prikazan broj programske linije koja se izvodi. Mogu biti prikazani i neki statusni pokazatelji. Na privesku za učenje nalaze se tipke za upravljanje stanjem hvataljke, kretanjima u zglobovima ili spoljašnjim koordinatama, tipka za resetovanje rada robota, tipka koja omogućuje izvođenje programa “korak po korak” i sl.

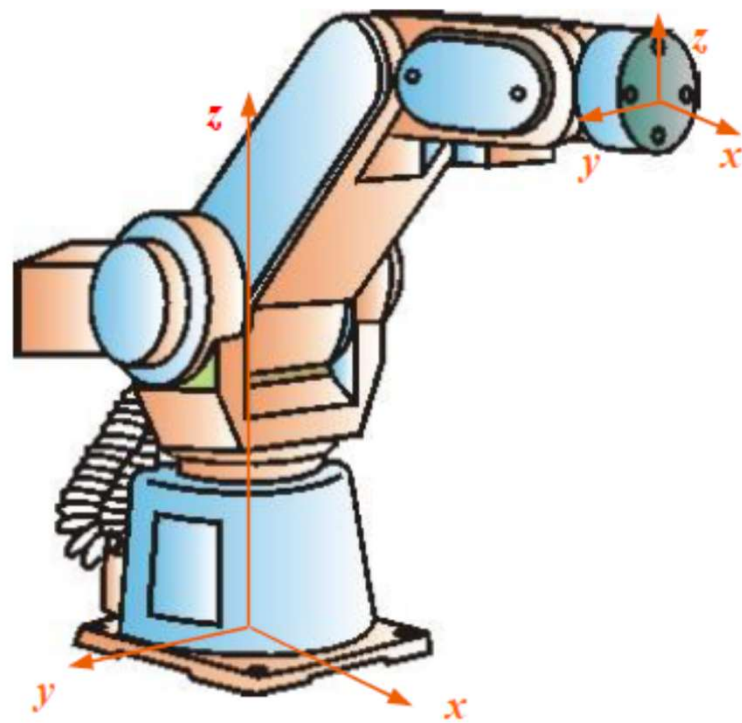


Sl 8.6. Uredaj za obuku

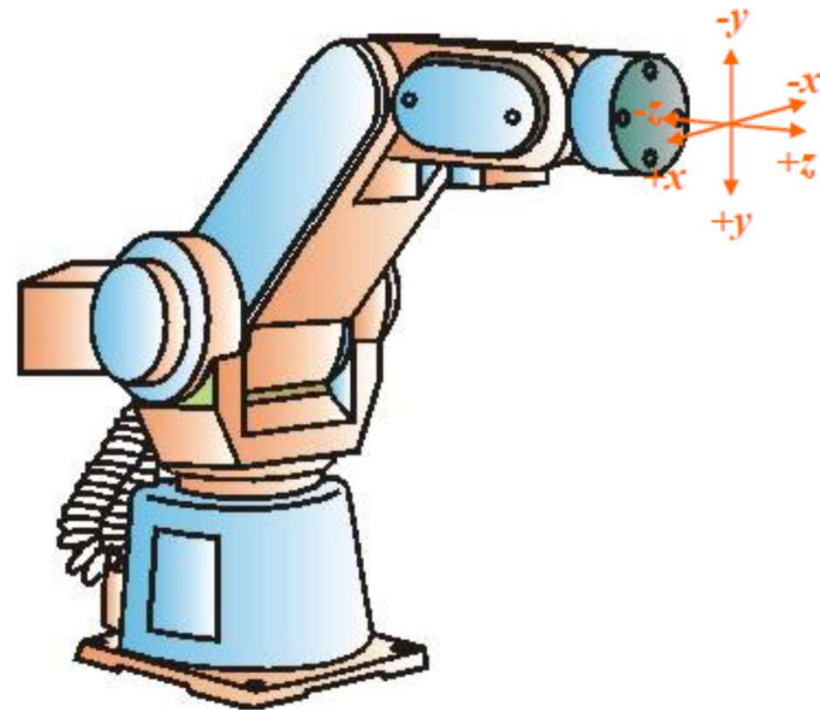
Tokom faze učenja, hvataljka se, postepenim pomjeranjima pomoću tipki uređaja za učenje, dovodi u željenu tačku. Tačka se pod određenim brojem skladišti u memoriju upravljačke jedinice, kao i kod “playback“ robota sa indirektnom fazom učenja. Ukupan broj sačuvanih tačaka kojima se definiše trajektorija ograničen je raspoloživom memorijom. Kod ovih robota mogu se pojaviti poteškoće želi li se hvataljka pomijerati po pravoj liniji posebno kod robota sa revolutnim zglobovima. Naime, kretanje po pravoj liniji prirodno je kod robota sa kartezijskom strukturom, dok se kod robota sa revolutnim zglobovima mora koristiti linearna interpolacija.

Problem se može riješiti primjenom dvije metode upravljanja kretanjem i to:

1. primjenom vanjskog koordinatnog sistema (engl. world coordinate system) i
2. primjenom koordinatnog sistema alata (engl. tool coordinate system).



a)



b)

Slika 9.4. Različiti načini upravljanja kretanjima osa

U prvom slučaju, prikazanom na sl. 9.4.a, definiše se vanjski koordinatni sistem pričvršćen za osnovu robota, a u središtu se hvataljke definiše novi koordinatni sistem sa istom orijentacijom.

Na sl. 9.4.b prikazan je koordinatni sistem alata čiji se položaj i orijentacija definišu u odnosu na površinu za koju je pričvršćena hvataljka.

Ova dva sistema upravljanja kretanjem svrsishodna su samo kada robot ima sposobnost pomijeranja duž prave linije. U oba slučaja robot se, primjenom uređaja za učenje, prebacuje u odgovarajući mod rada, koji omogućuje da se kretanje hvataljke ostvaruje duž osa navedenih koordinatnih sistema.

U upravljačku jedinicu robota unosi se program preko kompjutera, pojedinačno instrukcija po instrukcija, ili automatski uz postojanje namjenskih softwarea sa navedenom mogućnosti.

U tom slučaju, pisanje programa može biti izvršeno i na nekom drugom mjestu, a program će putem magnetnog medija ili mrežno biti proslijeđen kompjuteru povezanom za upravljačku jedinicu robota. Program se oslanja na sačuvne tačke u memoriji upravljačke jedinice.

Roboti sa off – line programiranjem i fazom učenja predstavljaju produžetak “playback” robota sa indirektnim učenjem, uz otklanjanje mnoštva nedostataka u pogledu povećane tačnosti, sinhronizovanosti kretanja sa drugim komponentama, uvođenju povratnih petlji i sl. Osnovne prednosti robota sa off-line programiranjem u odnosu na prethodne su:

- programiranje je vrlo jednostavno, jer uglavnom sintaksa instrukcije sa engleskom kraticom upućuju na radnju koju treba obaviti,
- modularno programiranje omogućuje korišćenje rutina iz biblioteka programa,
- moguća je sinhronizacija svih pokreta u radnom prostoru što posebno dolazi do izražaja kada je robot sastavni dio fleksibilnih proizvodnih sistema,
- informacije sa senzora ili CAD/CAM informacije mogu biti ugrađene u program što dopušta ugradnju logike u program, odnosno mogućnosti odlučivanja,
- program se može napisati bez pristupa robotu,
- postupcima simulacije trajektorije se mogu prekontrolisati, optimizovati, pa se izbjegavaju eventualna oštećenja robotskog sistema kod izvođenja programa bez prethodne kontrole,
- moguće je uslovno djelovanje robota, na bazi ugrađenih povratnih petlji.

Roboti sa strukturnim programiranjem

Kod svih do sada opisanih načina programiranja težište je bilo na opisivanju trajektorije koju treba da slijedi hvataljka, tako da će izvođenjem kretanja robot izvršiti planirani zadatak. Drugim riječima, korišćeni su programski jezici orijentisani programiranju kretanja.

Kod robota sa strukturnim programiranjem postoji mogućnost definisanja položaja predmeta primjenom vizuelnih sistema, a omogućuju definisanje kontinualne putanje.

Programiranje se nastoji okrenuti zadatku. Programi upotrijebljeni za robote ovog tipa nastali su razvojem kompjuterskih programskih jezika i imaju filozofiju strukturnog programiranja. Npr. robotski program AL temelji se na sintaksi kompjuterskog programskog jezika ALGOL, HELP je produžetak PASCALa. Nedostatak im je što od korisnika zahtijevaju određeni nivo znanja o robotskim jezicima.

Roboti sa programiranjem zadatka

Za razliku od svih dosadašnjih tipova robota čije je korišćenje bilo uslovljeno postojanjem dobro organizovane sredine, roboti sa programiranjem zadatka imaju mogućnost djelovanja u neorganizovanim sredinama. Opremljeni su vrlo snažnim senzorskim sistemom. U prvom redu to su vizuelni sistemi koji prikupljaju informacije o rasporedu predmeta u manipulacijskom prostoru robota. S obzirom da je postupak obrade informacija sa vizuelnih senzora najčešće izuzetno složen, a s druge strane, da bi te informacije bile raspoložive u realnom vremenu, robotski sistem mora biti opremljen vrlo brzom sofisticiranom upravljačkom jedinicom sa razvijenim strateškim modelom upravljanja.

Roboti ovog tipa moraju imati intenzivnu komunikaciju sa okolinom, imaju mogućnost donošenja odluka i u determinističkim i nedeterminističkim sredinama, pa su roboti multivarjabilni vođeni sistemi. Za programiranje rada ovih robota koriste se programski jezici čije naredbe označavaju radni zadatak kojeg treba obaviti.

Npr. robotski jezik AUTOPASS sadrži instrukciju koja od robota zahtijeva da prenese komad sa jednog na drugo mjesto, ne određujući pri tome kako će se prikupiti informacija o položaju radnog komada i sl. Danas se još uvijek roboti ovog tipa šire ne koriste i predmetom su brojnih istraživačkih radova

Robotski jezici

Danas postoji mnoštvo programskih jezika namijenjenih programiranju rada robota. Ne postoji apsolutni standard za robotske jezike. Neki robotski jezici nastali su od programskih jezika namijenjenih programiranju NC alatnih mašina, dok su drugi razvijeni iz kompjuterskih programskih jezika.

U prvom slučaju programske jezike su razvili uglavnom proizvođači opreme, dok su u drugom slučaju oni nastali na institutima za robotiku kroz istraživački rad. Oni su uglavnom napisani za određene robotske familije

Tabela 9.1. Robotski jezici

Robotski jezik	Proizvođač	Karakteristika
Funky	IBM	Omogućuje komunikaciju sa dodatnom opremom, a izvođenje faze učenja i unos instrukcija mogući su preko tastature.
VAL (VicArm Language)	Stanford University	Razvijen 1973. godine. Temelji se na robotskom jeziku WAVE. Omogućuje definiranje kontinuirane putanje.
AUTOPASS	IBM	Temelji se na programskom jeziku PL/I. Omogućuje višeprosorski rad, obradu informacija sa senzora, kao i rad više robota.
AL (Arm language)	Stanford Artificial Intelligence Lab.	Temelji se na programskom jeziku ALGOL. Omogućuje vođenje prihvatnice po putanji, brzini, interpolaciju putanje, izuzetno je složen i koristi se u istraživačke svrhe.
LM (Langage de Manipulatio)	IMAG robotics laboratory at the University of Grenoble	Razvijen je 1979. godine. Prihvata glavne koncepte jezika AL (paralelno programiranje), ali se temelji na mikrokompjuteru.

Pod robotskim programom naziva se skup instrukcija čijim će izvršavanjem upravljačka jedinica robotskog sistema omogućiti da kroz programirane kretnje robot obavi planirani zadatak.

Programiranje tekstualnim programskim jezicima ima prednosti u odnosu na programiranje npr. "playback" robota, a ogledaju se u slijedećem:

- bolje su iskorišćene senzorske sposobnosti,
- poboljšana je mogućnost upravljanja eksternom opremom,
- unaprijeđena je primjena robota u fleksibilnim proizvodnim ćelijama, gdje je funkcionisanje robota uslovljeno stanjima mašina i drugih komponenti fleksibilne ćelije,
- omogućena je komunikacija sa drugim kompjuterskim sistemima,