

Robotska manipulacija objektov glede na zunanjo referenčno točko

Tomaž KORITNIK, Darko KORITNIK

Izvleček: Prispevek na kratko opisuje tri matematične pristope k programiranju robotskih mehanizmov; gibanje v notranjih koordinatah, gibanje v zunanjih koordinatah z mobilnim orodjem in gibanje v zunanjih koordinatah glede na fiksno orodje – zunanjo referenčno točko. Za tretji način je podana matematična poenostavitev, ki se izogne invertiranju homogenih matrik dimenzije 4 v realnem času. Na koncu je podan še primer implementacije v programskem jeziku EPSON RC+.

Ključne besede: robotsko vodenje, notranje koordinate, zunanje koordinate, zunanja referenčna točka

1 Uvod

Robotski krmilniki imajo običajno vgrajenih več predprogramiranih funkcij za različne načine gibanja robota. Najosnovnejši način je gibanje po posameznih sklepih oz. gibanje v notranjih koordinatah. V tem primeru posamezne spremenljivke predstavljajo kote rotacijskih sklepov ali pomik translacijskih sklepov. Pri programiranju robota tak način gibanja omogoča zgolj enostavne premike po programiranih točkah, kjer nas vmesno gibanje robota ne zanima. Prednost preračunavanja gibov po notranjih koordinatah je sicer hitrost, saj je to nekoordinirano gibanje, ki ne zahteva sprotnega izračunavanja inverzne kinematike.

Drugi način je gibanje v zunanjih koordinatah oz. pomiki glede na referenčni koordinatni sistem robota. Pomiki v zunanjih koordinatah zahtevajo koordinirano gibanje posameznih osi in izračunavanje inverzne kinematike v vsakem trenutku, zato je robot v tem načinu običajno znatno počasnejši. Programski vmesniki največkrat omogočajo linearno in

cirkularno gibanje ter gibanje po zaporedju poljubnih točk v prostoru z ohranjanjem tangencialne ali normalne orientacije glede na trajektorijo. Gibanje v zunanjih koordinatah lahko poteka glede na referenčni koordinatni sistem robota ali fiksni dislocirani koordinatni sistem, kar je z matematičnega vidika ekvivalentno.

2 Zapis gibanja z mobilnim orodjem

Robotska roka ima na vrhu zadnjega sklepa nameščeno neko prijemalo ali orodje in praviloma nas zanima jo zunanje koordinate gibanja neke določene točke orodja, npr. konice za varjenje, ne pa koordinate samega vrha robota (prirobnice), kamor je orodje pritrjeno. Zato imajo programski vmesniki opcijo za definicijo referenčne točke glede na vrh robota, gibanje robota v zunanjih koordinatah pa se izvaja glede na to točko. Z matematičnega vidika je ta dodatna zahteva ekvivalentna obravnavi robotskega mehanizma z enim dodatnim segmentom, katerega lega je fiksna glede na prejšnji segment v kinematični verigi in ne predstavlja posebnih zapletov. Izračun želene lege vrha orodja je prikazan na *sliki 1*. Robot ima na vrhu nameščeno koničasto orodje, ki naj se pomika po robovih kvadra znanih

dimenzij v znani legi glede na referenčni koordinatni sistem.

Medsebojne lege zapišemo v obliki homogenih transformacij [1] s kvadratnimi matrikami dimenzije 4:

$${}^0H_v {}^vH_T = {}^0H_K {}^KH_T \quad (1)$$

$$H = \begin{bmatrix} R & T \\ P & 1 \end{bmatrix}; R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

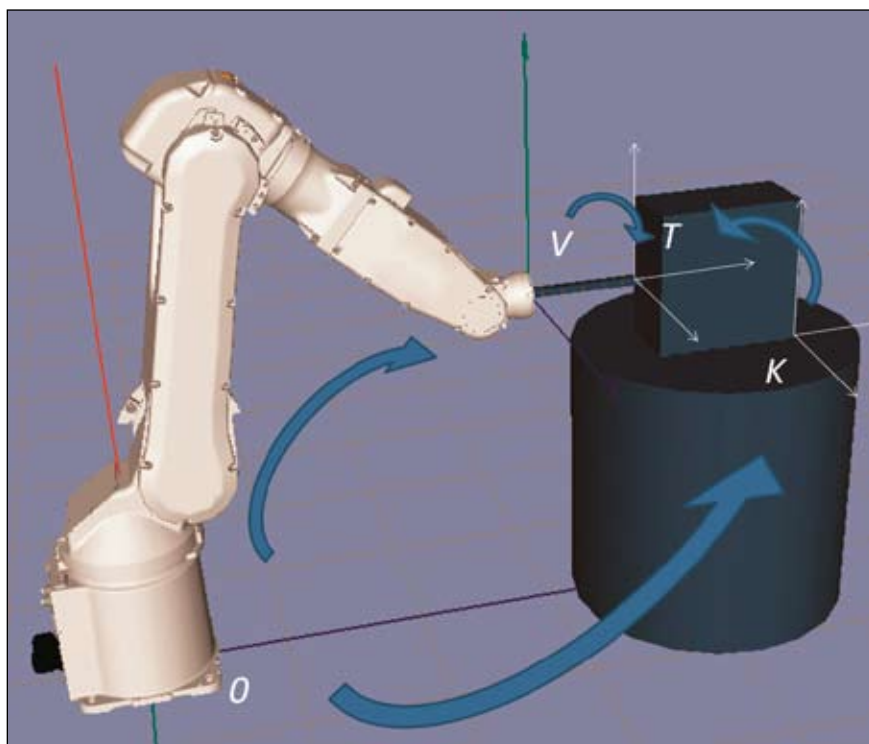
$$T = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, P = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad (2)$$

pri čemer je R rotacijska matrika, T translacijski vektor, P pa je vektor perspektive oz. projekcije. Stolpci rotacijske matrike so smerni vektorji rotacije koordinatnega sistema (R_x , R_y in R_z), translacijski vektor pa vsebuje koordinate translacije izhodišča koordinatnega sistema.

Lega vrha robota je znana iz krmilnika in sledi iz izračunavanja direktne kinematike v realnem času. Ravno tako poznamo lego orodja glede na vrh robota, ki je konstantna, isto pa velja tudi za lego kvadra v referenčnem koordinatnem sistemu. Legu konice orodja glede na kvadrotorej izrazimo kot:

$${}^KH_T = {}^0H_K^{-1} {}^0H_v {}^vH_T \quad (3)$$

Dr. Tomaž Koritnik, univ. dipl. inž.,
Darko Koritnik, univ. dipl. inž.,
DAX Electronic Systems,
d. o. o., Trbovlje



Slika 1. Obdelovalno orodje na robotu, obdelovanec je fiksni

■ 3 Gibanje glede na fiksno orodje

Tretji način vodenja predstavlja situacija, ko orodja ni moč namestiti na vrh robota, pač pa mora biti nameščeno na fiksni točki v delovnem območju robota, robot pa v prijemalu drži obdelovanec in ga premika koordinirano glede na to fiksno referenčno točko. Situacijo prikazuje slika 2. Robot drži kvader znanih dimenzij, ki naj ga po robovih pomika glede na konico orodja, pritrjenega na fiksno podlago.

Glede na sliko 2 zapišemo:

$${}^O H_V {}^V H_K = {}^O H_T {}^T H_K \quad (4)$$

Vidimo, da imata lega kvadra K in lega orodja T v tem primeru obrnjeno vlogo. V enačbi (4) nastopa znana lega vrha robota glede na referenčni koordinatni sistem O, znana je tudi lega sistema kvadra glede na vrh robota in je, podobno kot v prejšnjem primeru lega orodja glede na vrh, nespremenljiva, poznamo pa tudi lego orodja, ki je nespremenljiva v referenčnem sistemu. Izračunamo lahko torej zahtevano lego kvadra glede na orodje:

$${}^T H_K = {}^O H_T^{-1} {}^O H_V {}^V H_K \quad (5)$$

Običajno želimo pri obdelavi ali manipulaciji referirati lego orodja glede na obdelovanec in ne obratno, zato je izražanje lege na tak način nekoliko neintuitivno. Programiranje robota na tak način že pri enostavnejših

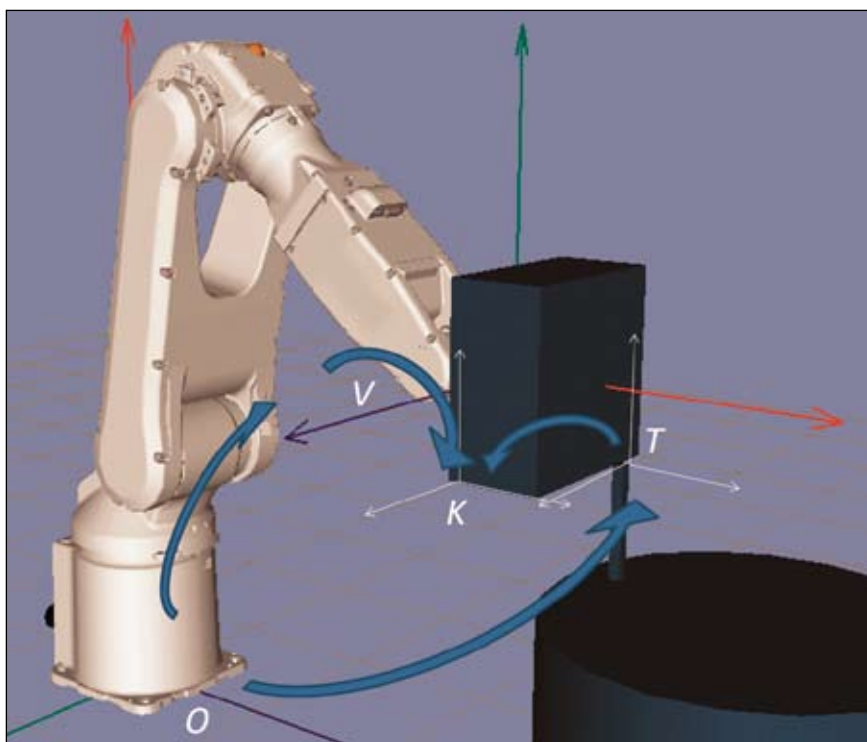
objektih zahteva dobro prostorsko predstavo, pri bolj kompleksnih oblikah v prostoru pa prej ali slej odpove. Zapišimo torej enačbo (5) na način, da bo odražala lego orodja glede na kvader, pri čemer upoštevamo: ${}^T H_K = {}^K H_T^{-1}$

$${}^K H_T = {}^V H_K^{-1} {}^O H_V^{-1} {}^O H_T \quad (6)$$

V enačbi (6) nastopata dve operaciji invertiranja matrike, kar je v splošnem dokaj zahtevno za procesiranje v realnem času. Lega kvadra glede na vrh robota je sicer konstantna in je dovolj, če jo izračunamo enkrat, lega vrha robota glede na referenčni koordinatni sistem pa se mora izračunavati v vsakem programskem koraku. Izračunavanje se poenostavi, če upoštevamo, da ima homogena transformacija lastnost ortogonalnosti oz. da je rotacijska podmatrika ortonormalna. Determinanta ortonormalne matrike je vedno enaka 1, inverz ortonormalne matrike pa je enak transponirani matriki:

$$R^{-1} = R^T \quad (7)$$

Matrika je obenem antisimetrična, kar pomeni, da dobimo inverz rotacijske matrike enostavno z negiranjem nediagonalnih elementov. Če privzamemo, da homogena trans-



Slika 2. Robot premika obdelovanec, fiksno obdelovalno orodje

formacija ne vsebuje projekcije oz. perspektive in skaliranja (to vedno drži v primeru opisa medsebojnih leg ortonormalnih koordinatnih sistemov v homogenih merskih enotah in pomeni, da je vektor v spodnji vrstici matrike vedno enak $[0 \ 0 \ 0 \ 1]$), je nezahteven tudi izračun inverza homogene transformacije dimenzije 4:

$$H^{-1} = \begin{bmatrix} r_1 & -r_2 & -r_3 & -(T \cdot R_X) \\ -r_2 & r_2 & -r_3 & -(T \cdot R_Y) \\ -r_3 & -r_3 & r_3 & -(T \cdot R_Z) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

■ 4 Primer implementacije v razvojnem okolju

Nekateri robotski krmilniki imajo vodenje z zunanjo referenčno točko že vprogramirano. Implementacijo si bomo pogledali na primeru okolja EPSON RC+, namenjenega vodenju vseh industrijskih robotov EPSON. V tem programskem okolju se ta opcija imenuje ECP motion – External Control Point oz. gibanje z zunanjo kontrolno točko. Funkcionalnost ECP je vprogramirana kot dodaten parameter vsem obstoječim ukazom za različne načine gibanja in ne kot samostojen ukaz. Na ta način je možna hitra predelava že obstoječih programov s preprostim dodajanjem parametra. Programsko okolje omogoča definiranje do 16 zunanjih kontrolnih točk. Ni odveč poudariti, da je funkcionalnost ECP smiselna samo za tiste ukaze, ki določajo gi-

banje v zunanjih koordinatah, v našem primeru je to gibanje po ravni črti, krožnem loku v ravnini ali prostoru in t. i. načinu »spline curve« (gibanje po poljubni prostorski krivulji, definirani z vmesnimi točkami in interpolacijo).

Spodnji primer prikazuje programiranje gibanja v ravni črti od točke P1 do P2 in po krožnem loku nazaj do P1 skozi P3, pri čemer je v prvih dveh vrsticah definiran ECP 1 z vrednostmi koordinat x, y, z = 200 mm, vsi koti rotacije pa so enaki 0:

ECPSet 1, XY(200, 200, 200, 0, 0, 0)
ECP 1

Go P1
Move P2 ECP
Arc P3, P1 ECP

Funkcionalnost ECP pride prav tudi med programiranjem točk in gibanjem robota v učnem načinu, še preden je gibanje sprogramirano in avtomatizirano. V oknu za ročno premikanje robota in učenje lahko izberemo ECP kot eno od opcij za izbiro načina gibanja robota (slika 3 levo), pri čemer je treba izbrati ustrezno številko ECP (slika 3 desno). Prikazani postopek s primerom lahko poenostavi programiranje robotske manipulacije, kadar imamo opravka z zahtevo po stacionarnem obdelovalnem orodju, obdelovanec pa ni trivialne oblike.

■ 5 Zaključek

Pokazali smo, da je izračun koordinat za vodenje robota z zunanjo referenčno točko razmeroma enostaven. Gornja izpeljava drži tudi splošno, torej v primerih, ko lega orodja ni konstantna glede na vrh robota oz. referenčni sistem ali ko lega obdelovanca ni konstantna glede na vrh robota oz. referenčni sistem, v vsaki računski iteraciji pa je potrebno izračunavati tudi te transformacije. Zaradi lastnosti homogenega transformacije oz. rotacijskih podmatrik je izračunavanje inverza trivialno in ga lahko izvajamo na vsakem robotskem krmilniku, ki omogoča vsaj osnovno programiranje in računske operacije seštevanja ter množenja.

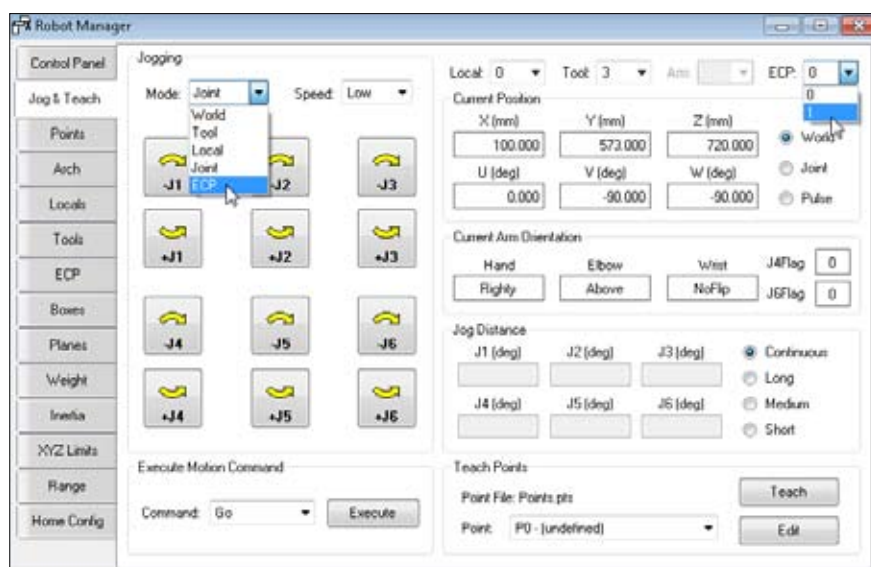
Literatura

- [1] Lenarčič, J., Bajd, T.: Robotski mehanizmi, Univerza v Ljubljani – Založba FE in FRI, 2003, pog. 1.

Robotic manipulation of objects by an external reference point

Abstract: The article briefly presents three mathematical approaches to programming of robotic mechanisms; motion in internal coordinates – joint space, motion in external coordinates – cartesian space, and motion by fixed tool – external reference point. For the third approach, a mathematical simplification is given to avoid the necessity of real-time 4-by-4 homogeneous matrix inverse computing. Finally, the ECP implementation in EPSON RC+ control environment is presented.

Keywords: robot control, internal coordinate space, external coordinate space, external reference point



Slika 3. Okno za ročno premikanje robota in učenje točk v programskem okolju RC