Robotok inverz geometriája

1. A gyakorlat célja

Inverz geometriai feladatot megvalósító függvények implementálása. A megvalósított függvénycsomag tesztelése egy kétszabadságfokú kar előírt végberendezés pozíciójának megfelelő csuklószögek meghatározásához.

2. Elméleti bevezető

Inverz geometriai feladat esetén ismert a robot végberendezésének pozíciója/orientációja és keressük a csuklóváltozókat, amelyek az adott pozíciónak megfelelnek. A feladathoz általában egy nemlineáris egyenletet kell megoldani.

2.1 Numerikus módszer az inverz geometriai feladat megoldásához

Jelöljük a robot ismeretlen csuklópozícióját q-val, a végberendezés előírt pozícióját G_D-vel. A robot inverz geometriai feladatát az alábbi nemlineáris egyenletrendszer megoldása adja:

$$G_R(q) = G_D \qquad (1)$$

ahol G_R a robot geometriáját írja le.
Hozzuk az egyenletrendszert az alábbi formába:

$$G(q)=0, \text{ ahol } G(q)=G_R(q)- G_D \qquad (2)$$

Az egyenletrendszer megoldásához fejtsük Taylor sorba a (2) egyenlet jobb oldalát.

$$G(q+dq) = G(q) + \frac{\partial G}{\partial q} dq + O(q) \qquad (3)$$

$O(q)$ magába foglalja a másod és magasabb rendű tagokat.
Ha a sorbafejtésben a $O(q)$-t elhanyagoljuk, akkor megkapjuk azt a dq lépésnagyságot egy kiinduló q-hoz képest, amelyre $G(q+dq)$ egyenlő nullával. A megoldás:

$$dq = -\frac{\partial G_R}{\partial q}^{-1} \left(G_R - G_D\right) \qquad (4)$$

$\frac{\partial G_R}{\partial q}$ a robot Jacobi mátrixát jelöli.

A (4) összefüggés nem adja meg a direkt geometriai feladat pontos megoldását. Ugyanakkor a dq lépéssel közelebb kerülünk a helyes megoldáshoz, mint a kiinduló q

pozícióban. Így az összefüggést alkalmazhatjuk iteratív eljáráshoz. Minden *dq* lépés kiszámítása után új kiindulópontként a *q+dq* értéket alkalmazzuk, majd újraszámítjuk a *dq*-t

Látszik, hogy a megoldáshoz szükség van a robot Jacobi mátrixára.

2.2 A robot Jacobi mátrixa

Amíg a robotok geometriai modellje a robot térbeli pozícióját írja le, a kinematikai modell, a robotok kinematikája a sebességgel kapcsolatos problémákat tárgyalja.

Csuklósebességek:

- rotációs csukló: szögsebesség a forgástengely körül
- transzlációs csukló: lineáris sebesség a mozgástengely mentén

A végberendezés sebessége:

Általában hat komponense van:

- sebességek *x*,*y*,*z* mentén
- szögsebességek *x*,*y*,*z* körül

$$y = \begin{pmatrix} \bar{v} \\ - \\ \omega \end{pmatrix}, \text{ ahol } \quad v = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \qquad \omega = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{5}$$

A robotok kinematikai problémája:

Összefüggés a csuklósebességek és a végberendezés sebessége között:

$$\dot{q} \leftrightarrow \begin{pmatrix} \bar{v} \\ - \\ \omega \end{pmatrix} \tag{6}$$

Az összefüggés meghatározásánál a direkt geometriai feladatból kell kiindulni:

- a végberendezés pozíciója és orientációja:

$$x(\bar{q}) = \begin{bmatrix} p_x(\bar{q}) \\ p_y(\bar{q}) \\ p_z(\bar{q}) \\ \alpha_x(\bar{q}) \\ \alpha_y(\bar{q}) \\ \alpha_z(\bar{q}) \end{bmatrix} \quad \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{a Denavit-Hartenberg } {}^{0}_{n}[T] \text{ utolsó oszlopának 1-3 eleme}$$

$$\dot{x} = \frac{\partial x}{\partial q} \cdot \dot{q}, \quad \text{ahol} \quad \frac{\partial x}{\partial q} = J = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} & \cdots & \frac{\partial x_1}{\partial q_m} \\ \vdots & & \ddots & \vdots \\ \frac{\partial x_n}{\partial q_1} & \frac{\partial x_n}{\partial q_2} & \cdots & \frac{\partial x_n}{\partial q_m} \end{bmatrix} \tag{7}$$

m – a robot szabadságfoka
n – a munkatér dimenziója (≤ 6)
J – a robot *Jacobi* mátrixa

Az inverz kinematikai feladat megoldásához a Jacobi mátrix inverzére van szükség:

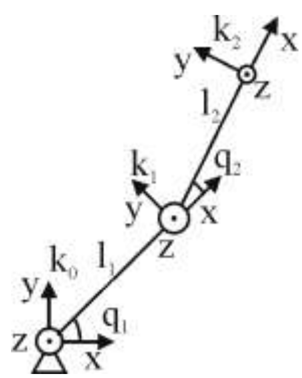
$$\dot{q} = J^{-1}(q) \cdot \dot{x} \tag{8}$$

Csuklógyorsulás és a végberendezés gyorsulása közötti összefüggés:

$$\ddot{x} = J(q) \cdot \ddot{q} + \dot{J}(q) \cdot \dot{q} \tag{9}$$

3. A mérés menete

Legyen az 1. ábrán látható két szabadságfokú robotkar.



1. Ábra: A két szabadságfokú kar

1. Táblázat: A két szabadságfokú kar Denavit Hartenberg paraméterei

i	θ	d	a	α
1	q_1	0	a_1	0
2	q_2	0	a_2	0

Feladat: valósítsunk meg egy függvényt, ami megoldja a kar inverz geometriai feladatát
A kar inverz geometriájának megoldását az alábbi nemlineáris egyenletrendszer adja.

$$\begin{cases} a_1 \cos q_1 + a_2 \cos(q_1 + q_2) = x \\ a_1 \sin q_1 + a_2 \sin(q_1 + q_2) = y \end{cases} \tag{10}$$

A robot Jacobi mátrixa:

$$J = \begin{pmatrix} -a_1 \sin(q_1) - a_2 \sin(q_1 + q_2) & -a_2 \sin(q_1 + q_2) \\ a_1 \cos(q_1) + a_2 \cos(q_1 + q_2) & a_2 \cos(q_1 + q_2) \end{pmatrix} \tag{11}$$

A *RobotGeometryProj* terv *RobotGeometry.cpp* állományában. A megvalósítandó függvények:

int RobotGeometry::DefineGeometry(ColumnVector q, ColumnVector &G)

A függvény bemenete a *q* csuklóváltozók vektora, kimenete a *G* vektor, ami a (10) egyenletrendszer bal oldala. a_1 és a_2 értékeket válasszuk 1-nek.

int RobotGeometry::DefineJacobian(ColumnVector q, Matrix &J)

A függvény bemenete a *q* csuklóváltozók vektora, kimenete a *J* mátrix, amit a (11) mátrix ad. a_1 és a_2 értékeket válasszuk 1-nek.

int RobotGeometry::InverseGeometry(ColumnVector G_Desired, ColumnVector Q_0, int Max_Iteration, Real Error_Tolerance, ColumnVector &Q, Real &Precision)

A függvény bemenetei:
G_Desired – az előírt pozíció, kételemű vektor, amely tartalmazza az *x* és *y* koordinátákat, amire keressök a feladat megoldását.
Q_0 - a kiinduló csuklókoordináták, kételemű vektor, amely tartalmazza a nulladik lépésben a q_1 és q_2 értékeket.
Max_Iteration – egész változó, amelyben megadjuk, hogy maximum hány lépésig keressök az eredményt.
Error_Tolerance – a megengedett eltérés az előírt és a valós megoldás között (számítási pontosság)
&Q – kételemű vektor, amiben az eredményt tároljuk
&Precision – a kapott számítási pontosság

A megvalósítandó algoritmus:

Q=Q0
Számláló = 0
Csináld
    Geometria meghatározása
    Jacobi mátrix meghatározása
    A dq lépés kiszámítása a (4) összefüggés alapján

A lépés elvégzése (Q = Q+dq)
A pontosság kiszámítása
amíg a pontosság az elvártnál vagy a számláló kisebb mint Max_Iteration

A pontosság meghatározásához a G-G_D vektor végtelen normáját számítjuk ki. (A végtelen norma egy vektor abszolút legnagyobb jelöli. Ehhez a Newmat *NormInfinity()* függvényét alkalmazhatjuk. A Jacobi mátrix inverzének kiszámításához a Newmat *i()* függvényét alkalmazhatjuk.
A program teszteléséhez a paramétereket válasszuk:
G_Desired = (0 0)
Q_0 = (0.1 0.1)
Max_Iteration = 100
Error_Tolerance = 1E-8

Vizsgáljuk az eredmény helyességét. Teszteljük a programot más *G_Desired* értékekre is. Teszteljük az algoritmust más *Max_Iteration* és *Error_Tolerance* értékekre.

4. Kérdések és feladatok

1. Keressen és implementáljon más numerikus algoritmusokat, amelyekkel az inverz geometriai feladat megoldható.
2. Tesztelje az algoritmust különböző q_0 kezdeti értékekre és írassa az iterációk számát, ami alatt az eredményt adott pontossággal megkapja. Milyen hatással van a q_0 kezdőérték megválasztása az algoritmus gyorsaságára?
3. Terjessze ki a módszert három *z* tengely körüli rotációs csuklót tartalmazó robot inverz geometriai feladatának megoldásához.