Ipari Robotok Kinematikája és Dinamikája

V/C. Kinematikai modellezés: Direkt és inverz kinematikai feladat

Összeállította: Dr. Kuti József
Szerkesztette: Dr. Galambos Péter

Óbudai Egyetem
Pro Sciencia et Futuro

2018. szeptember 21.
Budapest

Bejczy Antal iRobottechnikai Központ

A mozgást leíró Cartesian-térbeli és csuklótérbeli mennyiségek kapcsolatának leírása:

diff. kis elmozdulás és elfordulás $\leftrightarrow$ csuklókoordináta változások
sebesség és szögsebesség $\leftrightarrow$ csuklósebességek
gyorsulás és szöggyorsulás $\leftrightarrow$ csuklógyorsulások

DIREKT KINEMATIKAI FELADAT

Csuklókoordináta változások $\rightarrow$ diff. kis elmozdulás és elfordulás
Csuklósebességek $\rightarrow$ sebesség és szögsebesség
Csuklógyorsulások $\rightarrow$ gyorsulás és szöggyorsulás

INVERZ KINEMATIKAI FELADAT

Diff. kis elmozdulás és elfordulás $\rightarrow$ csuklókoordináta változások
Sebesség és szögsebesség $\rightarrow$ csuklósebességek
Gyorsulás és szöggyorsulás $\rightarrow$ csuklógyorsulások

Direkt kinematikai feladat

Behelyettesítés a kinematikai modell összefüggéseibe:

$$\begin{bmatrix} d\mathbf{r}_{TCP} \\ d\varphi \cdot \boldsymbol{t} \end{bmatrix} = \mathbf{J}(q_1, ..., q_m) \cdot d\mathbf{q},$$

$$\begin{bmatrix} \mathbf{v}_{TCP} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(q_1, ..., q_m) \cdot \dot{\mathbf{q}},$$

$$\begin{bmatrix} \mathbf{a}_{TCP} \\ \boldsymbol{\epsilon} \end{bmatrix} = \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}(q_1, ..., q_m) \cdot \ddot{\mathbf{q}}(t)$$

Ha a $\mathbf{J}(...)$ mátrix mindig invertálható lenne, az egyenletek átrendezésével megoldható lenne az inverz kinematikai feladat is - természetesen nem mindig invertálható.

Az inverz kinematikai feladat alkalmazásai: adott elmozdulást/sebességet/gyorsulást megvalósító csuklókoordináta változások/csuklósebességek/csuklógyorsulások meghatározása

Inverz kinematikai feladat

Adott egy 6 csuklós robot Jacobi-mátrixa az aktuális konfigurációban:

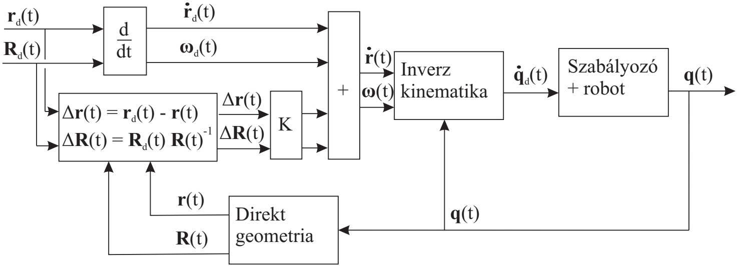
$$\begin{bmatrix} \mathbf{v}_{TCP} \\ \omega \end{bmatrix} = \underbrace{\mathbf{J}(q_1, \ldots, q_m)}_{6 \times 6} \cdot \dot{\mathbf{q}}$$

Ha az nem szinguláris helyzet, a sebességet/szögsebességet eredményező csuklósebességek mátrix invertálással kifejezhetőek:

$$\dot{\mathbf{q}} = \mathbf{J}(q_1, \ldots, q_m)^{-1} \cdot \begin{bmatrix} \mathbf{v}_{TCP} \\ \omega \end{bmatrix}$$

Inverz kinematika alapú trajektóriakövetés

A kiindulási $\mathbf{r}_0$ pozícióból és $\mathbf{R}_0$ orientációból, $\mathbf{r}_d(t)$, $\mathbf{R}_d(t)$ trajektóriát követni / azon eljutni egy célállapotba.



Inverz Kinematika blokk: milyen $\dot{\mathbf{q}}_d$ eredményezi a sebsségek és szögsebességet az aktuális $\mathbf{q}$ konfigurációban

Csak a Jacobi mátrix felírása az aktuális konfigurációban és "invertálásszerű" művelet

Nem invertálható Jacobi-mátrix

Lehetséges okok:

- a mátrix kvadratikus, de szinguláris helyzetben van a robot, így nem teljes rangú
- a mátrix nem kvadratikus, mert a robotkar redundáns
- (a mátrix nem kvadratikus, mert alulhatározott a robot)
- (ezek kombinációi...)

Adott egy $m > 6$ csuklós robot Jacobi-mátrixa az aktuális konfigurációban:

$$\begin{bmatrix} \mathbf{v}_{TCP} \\ \omega \end{bmatrix} = \underbrace{\mathbf{J}(q_1, \ldots, q_m)}_{6 \times m} \cdot \dot{\mathbf{q}}$$

ami nem szinguláris helyzet. $\mathbf{v}_{TCP}$ a kívánt sebesség és ω a kívánt szögsebesség.

Rendszerint végtelen módon eredményezhető a kívánt sebesség szögsebesség. Legyen a cél a megvalósítása minimális csuklósebességekkel ($\dot{\mathbf{q}}^T \dot{\mathbf{q}}$: minimális)

Megoldva az optimumszámítási problémát:

$$\dot{\mathbf{q}} = \underbrace{\mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1}}_{=\mathbf{J}^+} \begin{bmatrix} \mathbf{v}_{TCP} \\ \omega \end{bmatrix}$$

$\mathbf{J}^+$: ún. Moore-Penrose-féle pszeudoinverz

A hagyományos inverzre: $\mathbf{M}\mathbf{M}^{-1} = \mathbf{I}$, $\mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$
(csak kvadratikus, teljes rangú mátrixokra lehetséges)

Egyéb esetekre ún. általánosított vagy pszeudo-inverz ($\mathbf{M}^+$), amire:

1. $\mathbf{M}\mathbf{M}^+\mathbf{M} = \mathbf{M}$
2. $\mathbf{M}^+\mathbf{M}\mathbf{M}^+ = \mathbf{M}$
3. $(\mathbf{M}\mathbf{M}^+)^H = \mathbf{M}\mathbf{M}^+$
4. $(\mathbf{M}^+\mathbf{M})^H = \mathbf{M}^+\mathbf{M}$

ez mindig létezik és egyértelmű.

Az SVD felbontásból, ha

$$\mathbf{M} = \mathbf{U}\begin{bmatrix} \sigma_1 & \dots & 0 & 0 & \dots \\ \vdots & \ddots & & & \\ 0 & & \sigma_n & 0 & \dots \\ 0 & & 0 & 0 & \dots \\ \vdots & & \vdots & \vdots & \ddots \end{bmatrix}\mathbf{V}^T, \mathbf{M}^+ = \mathbf{V}\begin{bmatrix} 1/\sigma_1 & \dots & 0 & 0 & \dots \\ \vdots & \ddots & & & \\ 0 & & 1/\sigma_n & 0 & \dots \\ 0 & & 0 & 0 & \dots \\ \vdots & & \vdots & \vdots & \ddots \end{bmatrix}\mathbf{U}^T$$

Szinguláris helyzet: pszeudoinverz

Adott egy $n < 6$ rangú Jacobi-mátrix az aktuális konfigurációban:

$$\begin{bmatrix} \mathbf{v}_{TCP} \\ \boldsymbol{\omega} \end{bmatrix} = \underbrace{\mathbf{J}(q_1, \ldots, q_m)}_{6 \times m} \cdot \dot{\mathbf{q}}$$

$\mathbf{v}_{TCP}$ a kívánt sebesség és $\boldsymbol{\omega}$ a kívánt szögsebesség.

Nem valósítható meg, csak hibával: $\begin{bmatrix} \mathbf{v}_d \\ \boldsymbol{\omega}_d \end{bmatrix} \approx \mathbf{J}(q_1, \ldots, q_m) \cdot \dot{\mathbf{q}}$,

legyen a cél a hiba $\left(\begin{bmatrix} \mathbf{v}_d \\ \boldsymbol{\omega}_d \end{bmatrix} - \mathbf{J}(q_1, \ldots, q_m) \cdot \dot{\mathbf{q}} \right)^2$ minimálizálása

Megoldva az optimumszámítási problémát: $\dot{\mathbf{q}} = \mathbf{J}^+ \begin{bmatrix} \mathbf{v}_{TCP} \\ \boldsymbol{\omega} \end{bmatrix}$

$\mathbf{J}^+$: az ún. Moore-Penrose pszeudoinverz

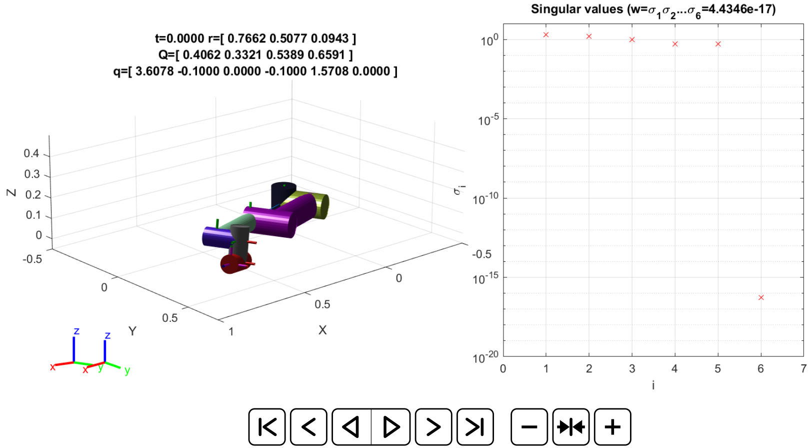
Pszeudoinverz: összegzés

A pszeudoinverz alkalmazásával:

- A többféle módon is megvalósítható a sebességet (azaz ha van nulltér), úgy valósítja meg, hogy a csuklósebesség vektor kicsi (minimális) legyen
- A nem megvalósítható sebességet kicsi (minimális) hibával valósítja meg

Általános számítása: pl. az SVD-ből

Példa: lineáris (max $1[m/s]$ sebességű) interpoláció szinguláris helyzetből



Ha a mozgás irányának van megvalósítható komponense, a robotkar lassan, kis kitéréssel, de kimozdul a szinguláris helyzetből

Szingularitás közeli helyzetek

(Matematikai) szinguláris állapot: a rang lecsökkenése, ekkor bizonyos irányba nem lehetséges a mozgás

Ha nem pont ilyen az irányba mozgatnánk a robotot és az kimozdul a szing. helyzetből, az élet megy tovább...

A szingularitás közelében, ahol a manipulálhatóság csökken:

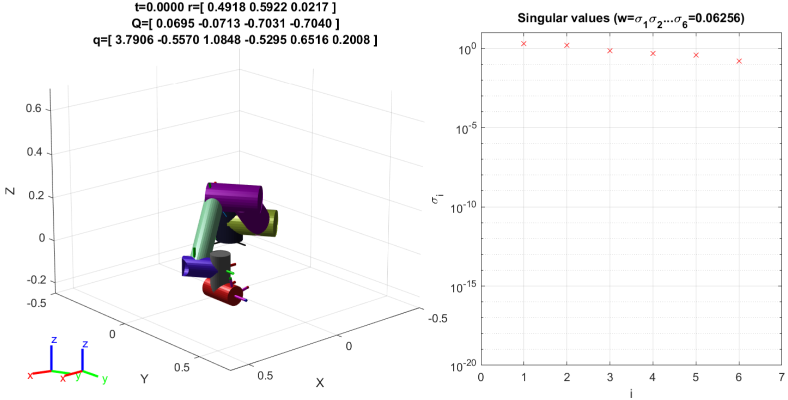
bizonyos irányokba a szokásos nagyságú sebességek/szögsebesség megvalósítása nagy (nem megvalósítható) csuklósebességeket igényel

Ekkor a korlátozásoknak megfelelően a vezérlő lelassítja a mozgást - mint az előző animáción is lassabb a mozgás a szinguláris helyzet közelében

DE kialakulhatnak olyan numerikus körülmények, amelyek miatt a mozgás szinte teljesen leállna, és a robot kis oszcilláló/remegő mozgást végezne - ezért a gyakorlatban a vezérlő a szinguláris helyzetben rendszerint letiltja a robot mozgását

Emlékeztető: a UR5 egyik szinguláris helyzete, amikor 4 csuklótengelye párhuzamos

Írjunk elő lineáris interpolációt ilyen konfiguráción keresztül:



A szingularitás közelében a korlátozott csuklósebességek miatt a mozgás belassul

A szinguláris helyzetben megáll, hogy minimális x/z korrekcióért körbe forog

Más robotstruktúrák esetén több szinguláris konfiguráció körül is látványosan megjelenik a probléma, lásd:

https://www.youtube.com/watch?v=zlGCurgsqg8

Csillapított pszeudoinverz

A szinguláris helyzet körüli "bóklászás" elkerülése:
a csillapított pszeudoinverz $(\mathbf{J}^T\mathbf{J} + \rho^2\mathbf{I})^{-1}\mathbf{J}^T$ használata

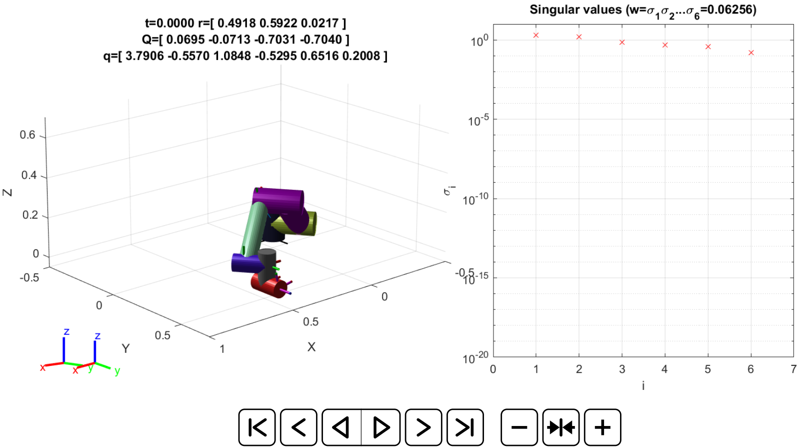
Ennek szinguláris értékei $\sigma_i/(\rho^2 + \sigma_i^2)$ az eddigi $1/\sigma_i$ helyett

Így ha $\sigma_i \to 0$, az érték nem végtelenbe, hanem $1/\rho^2$-hoz tart, tehát határolva van felülről

Ezzel a szingularitás közelében nem lesz annyira érzékeny, hanem ha követési hibával is, de fennmarad a mozgás

Matemaikailag az eddigi $\left(\begin{bmatrix} \mathbf{v}_{TCP} \\ \boldsymbol{\omega} \end{bmatrix} - \mathbf{J}(q_1, \ldots, q_m) \cdot \dot{\mathbf{q}}\right)$
sebesség-hiba költségfüggvényt kiegészíti a nagy csuklósebességeket büntető $\rho^2\dot{\mathbf{q}}^T\dot{\mathbf{q}}$ taggal

Kevésbé érzékeny a szinguláris helyzetre csillapított $\rho = 0.3$ pszeudoinverzzel:

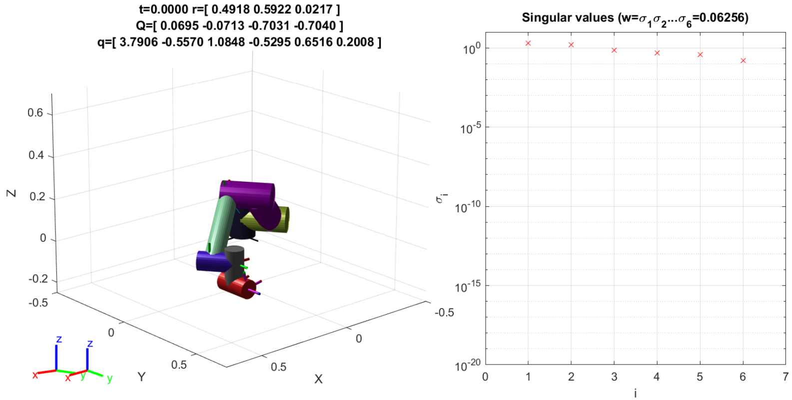


Inverz/pszeudoinverz helyett a transzponált alkalmazása:

$$\dot{\mathbf{q}} = \mathbf{J}^T \begin{bmatrix} \mathbf{v}_{TCP} \\ \boldsymbol{\omega} \end{bmatrix}$$

- Számításigény szempontjából olcsóbb
- Nem alkalmas trajektóriakövetésre: nem garantált hogy kis hibával követné
- De nem okoz problémát a szingularitás

Alkalmazásával az eddigieknél jóval simább mozgás a szinguláris helyzet körül, jóval nagyobb kitérésekkel:



A pálya konvergál a cél pozícióhoz, orientációhoz (részletek majd a dinamikai modellezés részben)

Gyorsulásalgoritmus

A robot mozgása tovább finomítható azzal, hogy a sebesség helyett a gyorsulást ($\mathbf{a}_{TCP}$) és szöggyorsulást írjuk elő (ϵ)

A csuklógyursulások meghatározásához a probléma itt is a Jacobi-mátrix invertálása:

$$\ddot{\mathbf{q}}(t) = \mathbf{J}(q_1, ..., q_m)^+ \left(\begin{bmatrix} \mathbf{a}_{TCP} \\ \epsilon \end{bmatrix} - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \right)$$

Az inverz feladat megoldására ugyanazon stratégiák:

- Jacobi-mátrix pszeudoinverz
- Jacobi-mátrix csillapított pszeudoinverz
- Jacobi-mátrix transzponált

Interpolációs típusok

Jellemző (támogatott/releváns) trajektóriák:

- egyenes
- körív
- spline / egyéb speciális trajektória

Felmerülő kérdések:

- képes-e megvalósítani a trajektóriát? (Megengedett csuklósebességek, önütközés, környezettel ütközés...)
- képes-e tartani az előírt sebességet? (Szingularitás közelében különösen nagy csuklósebességeket igényel..)
- gyorsításhoz engedélyezett-e a letérés a trajektóriáról, vagy leállás? (Technológiai/biztonsági szempontok)

Olyan pálya tervezése a csuklótérben, amely esetén a robot

- nem ütközve önmagával, a környezetével,
- azoktól egy biztonságos távolságot tartva,
- betartva a korlátozásokat a csuklóváltozókon, sebességeken, gyorsulásokon, nyomatékokon,
- minimális idő alatt ér a kívánt pózba (pl. egy munkadarabot megfogni)

(pl.: ROS MoveIt framework-ön keresztül több különböző matematikai hátterű solver elérhető.)

UR5 robot inicializálása:

```
»L1 = Revolute('a',0,'alpha',pi/2,'d',0.08916);
L2 = Revolute('a',-0.425,'alpha',0,'d',0);
L3 = Revolute('a',-0.39226,'alpha',0,'d',0);
L4 = Revolute('a',0,'alpha',pi/2,'d',0.10915);
L5 = Revolute('a',0,'alpha',-pi/2,'d',0.09456);
L6 = Revolute('a',0,'alpha',0,'d',0.0823);
robot = SerialLink([L1 L2 L3 L4 L5 L6], 'name', 'UR5 original');
```

Jacobi-mátrixa $q = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$ csuklóváltozók esetén:

```
J = robot.jacob0(q);
```

Jacobi mátrixának deriváltjának számítása, a fenti q és $dot\_q = [0, 2, 1, 4, 5, 6]$ csuklósebességek esetén:

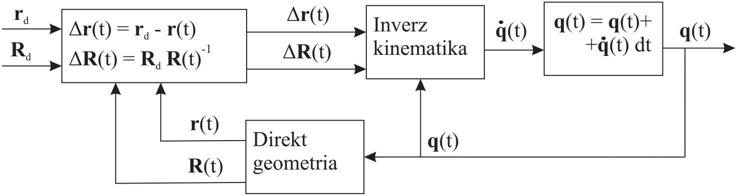
```
dotJ = robot.jacob_dot(q,dot_q);
```

Feladat: jelenlegi csuklókoordinátákból $\mathbf{q}$, eljutni olyan csuklókoordinátákba, amelyekkel a TCP póz: $\mathbf{T}_{desired}$

Szimuláljuk inverz kinematikával az aktuális konfigurációból a robot bemozgatását a cél pózba ($\mathbf{r}_{desired}$ pozíció és $\mathbf{R}_{desired}$ orientáció)

A célállapot konfigurációja az eredmény

A szabályozás egyszerűsíthető, mivel nem lényeges, hogy milyen pályán mozog a robot a célállapotba



Bekonvergál egy megoldásba, de az függ az indítási konfigurációból (probléma: szingularitásbeli deadlock helyzetek kezelése)

ÖSSZEFOGLALÁS

- A direkt kinematikai feladat megoldása behelyettesítés a lineáris összefüggésbe
- Az inverz kinematikai feladat nem megoldható, ha nem megvalósítható irányba (a Jacobi mátrix képterének hiányzó része felé) akarunk elmozdulni
  Közelítő megoldás a pszeudoinverz/csillapított pszeudoinverz vagy transzpontált alkalmazásával
- Redundáns robot esetén/szinguláris helyzetben egyes irányokba több, végtelen sok módon juthatunk el, a nulltérbeli mozgások miatt
- A fenti módszerekre épülnek az iteratív inverz geometriai megoldók

Köszönöm a figyelmet!

Óbudai Egyetem
Pro Sciencia et Futuro

Bejczy Antal iRobottechnikai Központ