

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

Ulveczki Balázs

Robotkarok mozgatása akadályok között

BSc Szakdolgozat

Témavezető: Szeghy Dávid

Geometria Tanszék



Budapest, 2017

Tartalomjegyzék

1. Bevezetés	3
1.1. A robotokról	3
1.2. Felhasznált ismeretek	3
1.3. Matematikai modell	4
2. Az alapprobléma - Denavit-Hartenberg paraméterek	7
2.1. Direkt kinematika	7
2.2. Inverz kinematikai feladat	11
2.3. Direkt sebességkinematika	12
2.4. Inverz sebességkinematika	14
2.5. Szingularitások	15
2.6. Inverz kinematikai feladat - ismét	15
3. Akadályok elkerülése - a konfigurációs tér	17
3.1. Bevezetés	17
3.2. Direkt módszer	18
3.3. Diszkrét konfigurációs tér felépítése	21
4. Útkeresés konfigurációs téren	26
4.1. Diszkrét eset	26
4.2. Folytonos eset	29
5. Összefoglalás	34
6. Felhasznált irodalom	35
7. Köszönetnyilvánítás	36

1. fejezet

Bevezetés

1.1. A robotokról

Mai modern világunkban már senki számára nem ismeretlenek a robotok, hiszen a háztartásunk és iparunk is nehezen képzelhető el robotok alkalmazás nélkül. A robotok részei a popkultúrának is, megannyi könyv, film, dal főszereplői emberszerű intelligenciával, humanoid, vagy éppen pusztításra termett testtel rendelkező robotok.

Bár az emberiség évezredek óta álmodik robotokról, melyek helyettünk dolgoznak, a robotika igen fiatal tudományág. Az első tudományos igényű munkákra egészen az 1940-es évekig kellett várni, mikor először nyílt lehetőség elektronikusan vezérelt robotok tervezésére. Azonban a számítástechnika robbanászerű fejlődésével a robotika is előretört, ma már humanoid, közel emberi mozgással rendelkező robotokat is létre tudunk hozni, melyeket mesterséges intelligencia vezérel.

Jelen dolgozat tárgyát a robotkarok képzik, melyekkel első sorban az iparban találkozhatunk, azonban az orvosi felhasználásuk is egyre gyakoribb. Az alapvető elvárás ezekkel az eszközökkel szemben, hogy munkájuk ne igényeljen emberi beavatkozást, a felmerülő feladatokat, mint tárgyak mozgatása, eszközök és szerszámok megfelelő helyre juttatása, előre megírt program vezérelje.

Célunk, hogy ezen robotkarok mozgásával, mozgatásával megismerkedjünk, programozásukhoz matematikai háttérrel biztosítsunk.

1.2. Felhasznált ismeretek

A szakdolgozat jelentős részben támaszkodik a Spong könyvre [2], mely a robotikában széles támogatottságot élvez. A tematikus egyezésem felül általában a jelölések is illeszkednek, amennyiben ez lehetséges volt, valamint több ábra is eredetileg a könyvben szerepelt.

Ezen felül ismertnek tekintjük a következőket

1. Elemi és projektív geometria alapjai, projektív leképezések, affin leképezések,

melyek megtalálhatóak a Hajós jegyzetben [1].

2. A többváltozós analízis vektormezőkre vonatkozó része, mely a Laczkovich- T. Sós [4][5] jegyzetekbenben fellelhető.
3. A vektorterek és normák alapfogalmainak ismerete. Ez megtalálható a Kolmogorov könyvben [6].

1.3. Matematikai modell

Matematikai eszközök segítségével szeretnénk robotkarokkal dolgozni, így először is definiálnunk kell, hogy mi is az a robot.

Definíció. *(A Robot Institute of America szerint.) A robot egy újraprogramozható, többfunkciós manipulátor anyagok, eszközök, részegységek vagy speciális műszerek változókkal programozott mozdulatsor segítségével való mozgatására tervezve, különféle feladatok elvégzésére.*

Mint már említettük, a programozhatóság alapvető feltétel, így a manipulátort minél általánosabban, ugyanakkor könnyen kezelhetőnek kell vennünk.

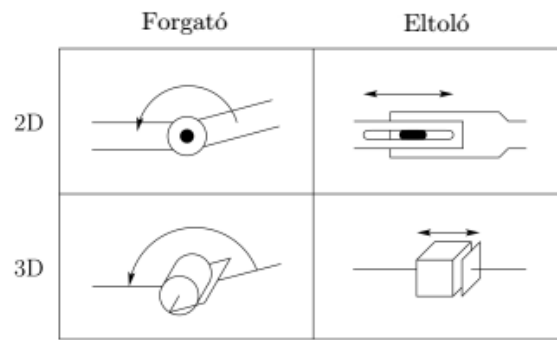
Az általunk vizsgált robotkarok három egységből állnak össze: merev szegmensek, az őket összekapcsoló ízületek(csuklók), illetve a kéz, mely a robotkar végén foglal helyet, és alkalmas az eszközök és tárgyak megfogására, vagy éppen maga a munkaeszköz.



Jelen dolgozatban a kéz felépítésével nem foglalkozunk részletesen, azt a robotkar pontszerű végének tekintjük, melyet mozgatni szeretnénk a tér meghatározott pontjaiba.

Definíció. *Munkatérnek nevezzük azon pontok halmazát, ahova a robotkar valamely pontja eljuttatható.*

A csuklóknak két elemi típusával foglalkozunk; a forgató, másnével rotációs csuklóval (R); valamint az eltoló, vagy prizmatikus csuklóval (P). Ezen elemi, egyetlen szabadsági fokkal rendelkező, így egy paraméterrel leírható ízületek segítségével előállíthatóak a bonyolult, több szabadságfokkal rendelkező ízületek, például egy gömbcsukló két forgató ízülettel modellezhető, melyeket egy 0 hosszú szegmens köt össze.



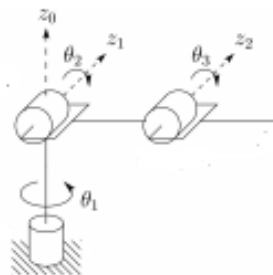
Definíció. Csukló hatástengelyének forgató csukló esetén a forgatás tengelyét nevezzük; eltoló esetben pedig az eltolásnak megfelelő irányvektorú, a csuklón átmenő egyenest értjük.

Definíció. Konfigurációnak nevezzük a robotkar egy állapotát, melyet a $q(q_1, q_2 \dots q_n)$ vektorral jellemezhetünk, ahol a q_i az i . csukló paraméterét jelenti.

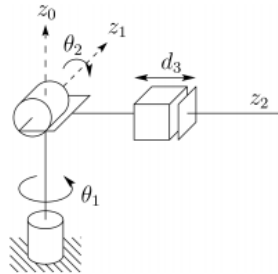
1. Megjegyzés. Forgató csukló esetén q_i a szegmensek, mint félegyenesek szöge radiánban; eltoló csukló esetén pedig a teljesen visszahúzott állapothoz képest vett eltolás vektorának hossza.

A robotkarokat típusokba sorolhatjuk a csuklók fajtája szerint. Néhány példa:

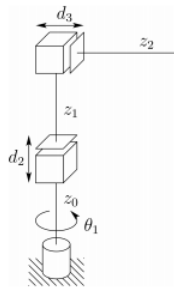
1. Az antropomorf robotkar (RRR), mely 3 forgató csuklóval rendelkezik, melyek közül az első kettő hatástengelye merőleges. Ez írja le megközelítően az emberi kar mozgását.



2. Gömbi robotkar (RRP), mely nevét a gömbi koordináta-rendszerrel való hasonlósága miatt kapta, ugyanis a paraméterek megfeleltethetőek a 3 koordinátának.



3. Hengeres robotkar (RPP), mely a gömbi robotkarhoz hasonlóan a hengerkoordináta-rendszerrel mutat hasonlóságot.



2. fejezet

Az alapprobléma - Denavit-Hartenberg paraméterek

Ebben a fejezetben megismerkedünk a robotkarok mozgásának alapjaival, hogy hogyan függnek össze a csuklók paraméterei, illetve a kéz helyzete és sebessége a munkatérben.

4 fő kérdés merül föl, melyekkel foglalkozni fogunk: a paraméterek ismeretében határozzuk meg a kéz helyzetét; határozzuk meg a paramétereket, hogy a kéz egy adott helyen legyen; a paraméterek változtatása esetén milyen sebességvektorral mozog a kéz; illetve adott sebességvektorhoz milyen függvény szerint változtassuk a paramétereket?

2.1. Direkt kinematika

Az első és egyben legegyszerűbb probléma a kéz helyzetének meghatározása a paraméterek függvényében, melyre általában direkt kinematikai feladatként utalunk.

A megoldáshoz lépésenként fogunk visszafelé végighaladni a robotkar szegmensein, egyszerre mindig csak a szomszédos részek egymáshoz viszonyított helyzetét vizsgálva, majd az alaphoz eljutva az addigi lépéseket összesítve megkaphatjuk a kéz helyzetét a robotkar alapjához képest.

Ehhez irányított tengelyű, jobbsodrású, ortonormált koordináta-rendszereket illesztünk minden szegmenshez úgy, hogy a szegmens pontjai állandóak legyenek a hozzá illesztett koordináta-rendszerben. Az i . szegmenshez illesztett koordináta-rendszer legyen $(o_i; x_i; y_i; z_i)$. Amennyiben ezt véletlenszerűen tesszük meg, úgy a koordináta-rendszerek közötti transzformáció leírásához 6 paraméterre van szükségünk (3 az origó eltolása, 3 egy $SO(3)$ -beli forgatás). Azonban a Denavit-Hartenberg konvenció lehetőséget kínál arra, hogy a koordináta-rendszer megválasztásánál két szabály alkalmazásával csak 4 paraméterre legyen szükségünk a transzformáció leírásához. A szabályok a következők:

1. A z_i tengely egybeesik az $i+1$. izület hatástengelyével.
2. Az x_i tengely merőleges a z_{i-1} tengelyre, valamint metszi is azt.

Tétel. *Mindig lehetséges így felvenni a koordináta-rendszereket.*

Bizonyítás. Indukcióval bizonyítunk. A 0., azaz bázis rendszerben csupán z_0 kötött, az első csukló tengelyén felvehetünk tetszőleges pontot origónak és z_0 -t jobbsodrású rendszerre kiegészítő x_0, y_0 vektorokat.

Tegyük föl, hogy az első $i-1$ koordináta-rendszer már adott megfelelő feltételekkel. Ekkor a z_i tengelyt kijelöli számunkra az i . csukló hatástengelye az első pont szerint, valamint z_{i-1} már ismert. Amennyiben ezek a tengelyek párhuzamosak, x_i tengely tetszőleges, velük egy síkban lévő, rájuk párhuzamos egyenes lehet. Ekkor az x_i, z_i metszéspontja kijelöli o_i pontot, valamint y_i rájuk merőleges, origóból induló egyens, mely irányítását a jobbsodrás megadja. Ha z_i, z_{i-1} nem párhuzamosak, úgy x_i tengely a két egyenes normáltranszverzálisa, vagy a metszés esetén a metszéspontban a síkjukra állított merőleges. Ekkor ismét egyértelműen adott az origó, mint a metszéspont, illetve az jobbsodrású ortogonálissá kiegészítő y_i . \square

Tudjuk tehát, hogy fel tudjuk venni így a koordináta-rendszereket. Most pedig nézzük a 4 paramétert, ami leírja az $i-1$. rendszer helyzetét az i . rendszerben. Ehhez vegyünk föl egy köztés (o, x, y, z) koordináta-rendszert az alábbi tulajdonságokkal:

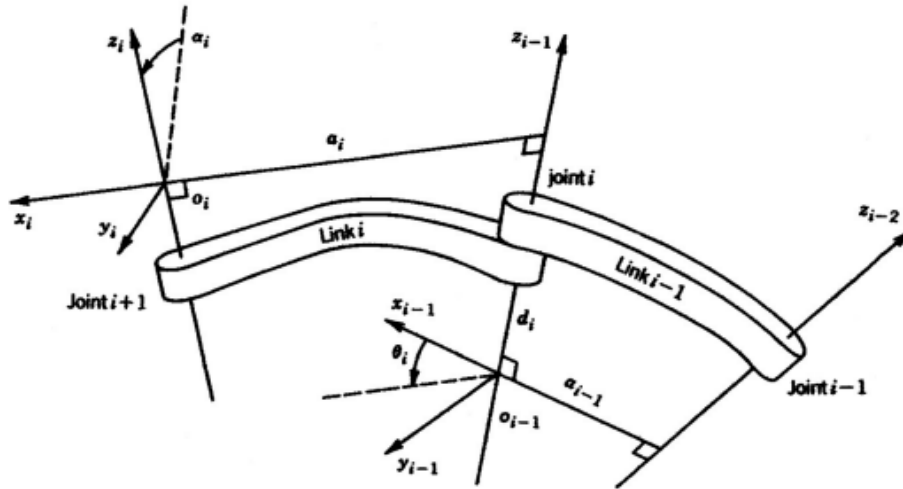
1. $o = x_i \cap z_{i-1}$.
2. $x = x_i$, és azonos irányítású is vele.
3. $z = z_{i-1}$, és azonos irányítású vele.

Ekkor y adott az ortogonalitás és jobbsodrású rendszer miatt.

Legyen $l_i = d(o_{i-1}; o)$, $a_i = d(o, o_i)$, ahol d z_{i-1} -nek megfelelően irányított távolság, valamint legyen α_i z és z_i irányított szöge (melynél figyelembe vesszük, hogy a tengelyek irányítását is tartja), θ_i pedig x_{i-1} és x irányított szöge (hasonlóan a tengelyek irányítástartásával). Ekkor $(l_i; a_i; \alpha_i; \theta_i)$ egyértelműen meghatározza a két koordináta-rendszer egymáshoz viszonyított helyzetét, azaz valóban elég 4 adat a szokásos 6 helyett.

Vegyük továbbá észre, hogy z_{i-1}, z_i egymáshoz viszonyított helyzete nem függ az i . csukló állásától, így α_i, a_i csak a szegemens alakjától függő konstansok. Továbbá amennyiben a csukló prizmatikus, úgy annak állása (mely egy eltolás) nem befolyásolja a két x tengely szögét, azaz θ_i csak a forgó csukló állásától függ, az l_i távolság pedig éppen forgató csukló esetén konstans, és az eltoló csukló paraméterétől függhet.

Azaz a 4 paraméter közül, mely meghatározza az áttérési transzformációt, 3 minden esetben a robotkar geometriájából következő konstans, 1 pedig a csukló paraméterének függvénye.



Megvannak tehát az áttérést meghatározó paramétereink, már csak azt kell megadnunk, hogy ezek ismeretében hogyan számolható ki az egyik rendszerben adott pont koordinátái a másik rendszerben. Ehhez 4 dimenziós homogén koordinátákat fogunk használni, mivel ezek segítségével bármely lineáris transzformáció felírható egy 4x4-es mátrixszal való szorzásként.

Ahhoz, hogy az i . rendszerről áttérjünk az $i-1$. rendszerre, az előzőekben látottak alapján 4 lépést kell végrehajtanunk:

1. Eltolunk az x_i tengely mentén a_i hosszan, ennek mátrixsza

$$\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Elforgatunk az x_i tengely körül α_i szöggel. Ennek mátrixsza

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Eltolunk a z_i tengely mentén l_i hosszan, aminek mátrixsza

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Elforgatunk a z_i tengely mentén θ_i szöggel. A mátrix

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Az áttérést ezen transzformációk egymás utáni végrehajtásával kapjuk, melynek az adott mátrixok szorzása felel meg, jobbról balra a fenti sorrendben. Tehát az $i-1$. koordináta-rendszerrel az i -re való áttérés mátrixa homogén koordináták esetén

$$M_i = M(l_i, \theta_i, a_i, \alpha_i) = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i * \sin\theta_i & \sin\theta_i * \sin\alpha_i & a_i * \cos\theta_i \\ \sin\theta_i & \cos\theta_i * \cos\alpha_i & -\cos\theta_i * \sin\alpha_i & a_i * \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & l_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mivel az n szegmensből álló robotkarunk keze az (o_n, x_n, y_n, z_n) koordináta-rendszerben állandó (hiszen mereven hozzá van rögzítve), legyenek a homogén koordinátái $\begin{pmatrix} u_n \\ v_n \\ w_n \\ 1 \end{pmatrix}$.

Ekkor a kéz homogén koordinátái az alaprendszerben

$$\begin{pmatrix} u_0 \\ v_0 \\ w_0 \\ 1 \end{pmatrix} = M_1 * M_2 \dots M_n * \begin{pmatrix} u_n \\ v_n \\ w_n \\ 1 \end{pmatrix}$$

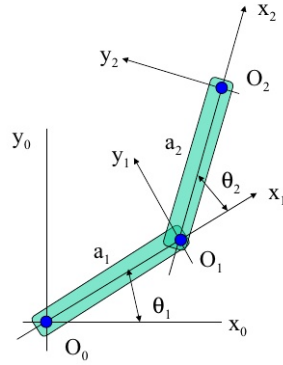
Mivel a teljes jobb oldal csak a csuklók állásától és a robotkar geometriájától függ, a fentebb ismertetett módon, így megoldottuk a direkt kinematikai feladatot.

Egy egyszerű példa

Tekintsünk egy síkbeli, RR robotkart, melynek szegmensei a_1, a_2 hosszú szakaszok, és határozzuk meg a Denavit-Hartenberg paramétereiket!

Ekkor mindkét csukló hatástengelye merőleges a síkra, amiben dolgozunk, így a z_0, z_1 tengelyek is azok. Ekkor x_i tengelyeket a robotkar síkjában lévőnek választjuk az egyszerűség kedvéért, így pedig y_i szintén síkbeli, és már adott a jobbsodrás miatt. Így a robotkartól függő paraméterek értékét már tudjuk is: $l_0 = a_1, l_1 = a_2, \alpha_0 = \alpha_1 = 0$. Ebből azt is megállapíthatjuk, hogy az áttérési mátrixaink 3. sora és oszlopa mindig a 0 vektor lesz, így ezeket el is hagyhatjuk, 4x4 helyett 3x3 mátrixokkal számolva így.

A θ_0 és θ_1 paraméterek az ábrán látható módon olvashatóak le, mely definíció szerint az x tengelyek elfordulása.



2. Megjegyzés. Ezek a θ szögek nem a csukló paraméterei, hanem a D-H paraméterek. A csukló állásából a $\theta_i = q_i - \pi$ módon kaphatjuk meg.

Ekkor az áttérési mátrixok:

$$M_0(l_0; \theta_0) = \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & l_0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 & l_0 * \cos\theta_0 \\ \sin\theta_0 & \cos\theta_0 & l_0 * \sin\theta_0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M_1(l_1; \theta_1) = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & l_1 * \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & l_1 * \sin\theta_1 \\ 0 & 0 & 1 \end{pmatrix}$$

Az ábrán látható, hogy ezúttal a kéz homogén koordinátái a 2 indexű koordinátarendszerben a $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, így a korábban megismert összefüggés szerint:

$$\begin{pmatrix} v_0 \\ w_0 \\ 1 \end{pmatrix} = M_0(l_0; \theta_0) * M_1(l_1; \theta_1) * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} =$$

$$\begin{pmatrix} l_0 * \cos\theta_0 + l_1 * \cos\theta_0 * \cos\theta_1 - l_1 * \sin\theta_0 * \sin\theta_1 \\ l_0 * \sin\theta_0 + l_1 * \cos\theta_0 * \sin\theta_1 + l_1 * \cos\theta_1 * \sin\theta_0 \\ 1 \end{pmatrix} =$$

$$\begin{pmatrix} l_0 * \cos\theta_0 + l_1 * \cos(\theta_0 + \theta_1) \\ l_0 * \sin\theta_0 + l_1 * \sin(\theta_0 + \theta_1) \\ 1 \end{pmatrix}$$

2.2. Inverz kinematikai feladat

A robotkarok mozgatásának megismerésében a következő logikus lépés az inverz kinematikai feladat, mely annyiban tér el a direkt kinematikaitól, hogy most a csuklók

állása helyett a kéz pozíciója adott az alaprendszerben, és ehhez keresünk megfelelő paramétereket a csuklókhoz.

Azonban ennek a feladatnak az általános megoldása az előbb látott egyenlet megoldása az n darab csukló paraméterre, mint változóra. Ez pedig egy 16 egyenletből álló, n változós, nemlineáris egyenletrendszer, melyet nemhogy általánosan, de konkrét esetekben is nehezen tudunk megoldani.

Van azonban egy másik módszer, melyhez előbb meg kell ismerkednünk a robotkarok sebességkinematikai feladataival.

2.3. Direkt sebességkinematika

Direkt sebességkinematikai feladatnak nevezzük azt a problémát, ha adottak a csuklók paraméterei az idő függvényében, és kíváncsiak vagyunk, hogy ez mekkora sebességet jelent a kéz esetében a munkatéren.

Legyenek adottak $\forall i \in \{0 \dots n\}$ esetén $l_i(t), \theta_i(t), a_i(t), \alpha_i(t)$ a Denavit-Hartenberg paraméterek idő szerinti, folytonosan deriválható függvényei.

3. Megjegyzés. *Bár tudjuk, hogy az $a_i(t), \alpha_i(t)$ paraméterek mindig konstansok, valamint minden $l_i(t), \theta_i(t)$ párok közül az egyik, ennek ellenére függvényként kezeljük őket a konzisztensebb jelölések érdekében.*

Ekkor a direkt kinematikai feladat megoldását felhasználva a kéz koordinátái az alaprendszerben

$$\begin{pmatrix} u_0(t) \\ v_0(t) \\ w_0(t) \\ 1 \end{pmatrix} = M_0(t) * M_1(t) \dots M_n(t) * \begin{pmatrix} u_n \\ v_n \\ w_n \\ 1 \end{pmatrix}$$

4. Megjegyzés. *A $\begin{pmatrix} u_n \\ v_n \\ w_n \\ 1 \end{pmatrix}$ vektor nem függ az időtől, hiszen ez a kézhez mereven rögzített koordináta-rendszer.*

A robotkarunk tulajdonságainak ismeretében a jobb oldalon a szorzás elvégezhető, és a következő alakú eredményre vezet:

$$\begin{pmatrix} u_0(t) \\ v_0(t) \\ w_0(t) \\ 1 \end{pmatrix} = \begin{pmatrix} f_1(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, 0_n(t), \theta_n(t), a_n(t), \alpha_n(t)) \\ f_2(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, 0_n(t), \theta_n(t), a_n(t), \alpha_n(t)) \\ f_3(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, 0_n(t), \theta_n(t), a_n(t), \alpha_n(t)) \\ 1 \end{pmatrix}$$

5. Megjegyzés. Az f függvényeket csak egyszer kell egy adott robotkar esetében meghatározni az M_i mátrixok segítségével, így a valós idejű számításoknál ezek már ismertek és rendelkezésre állnak. Ráadásul mivel a mátrixok is csak a paramétereiktől függenek, így azonos felépítésű (csuklók típusa és száma szerint) robotkarok esetén az f függvények is azonosak.

A sebességvektort a helyvektor idő szerinti deriváltjaként kapjuk. Így a kéz sebességvektora a t . időpillanatban :

$$\begin{pmatrix} u'_0(t) \\ v'_0(t) \\ w'_0(t) \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{d}{dt} f_1(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, l_n(t), \theta_n(t), a_n(t), \alpha_n(t)) \\ \frac{d}{dt} f_2(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, l_n(t), \theta_n(t), a_n(t), \alpha_n(t)) \\ \frac{d}{dt} f_3(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, l_n(t), \theta_n(t), a_n(t), \alpha_n(t)) \\ 0 \end{pmatrix}$$

Ekkor a többváltozós láncszabályt alkalmazva:

$$\begin{aligned} u'_0(t) &= \frac{\delta}{\delta l_0} f_1(l_0(t), \dots) * l'_0(t) + \frac{\delta}{\delta \theta_0} f_1(\dots \theta_0(t) \dots) * \theta'_0(t) \dots \\ &\quad \dots + \frac{\delta}{\delta \alpha_n} f_1(\dots \alpha_n(t)) * \alpha'_n(t) \\ v'_0(t) &= \frac{\delta}{\delta l_0} f_2(l_0(t), \dots) * l'_0(t) + \frac{\delta}{\delta \theta_0} f_2(\dots \theta_0(t) \dots) * \theta'_0(t) \dots \\ &\quad \dots + \frac{\delta}{\delta \alpha_n} f_2(\dots \alpha_n(t)) * \alpha'_n(t) \\ w'_0(t) &= \frac{\delta}{\delta l_0} f_3(l_0(t), \dots) * l'_0(t) + \frac{\delta}{\delta \theta_0} f_3(\dots \theta_0(t) \dots) * \theta'_0(t) \dots \\ &\quad \dots + \frac{\delta}{\delta \alpha_n} f_3(\dots \alpha_n(t)) * \alpha'_n(t) \end{aligned}$$

Vegyük észre, hogy a jobb oldalon mindhárom esetben a D-H paraméterek valamely lineáris kombinációja áll. Így átírhatjuk a következő alakra:

$$\begin{pmatrix} u'_0(t) \\ v'_0(t) \\ w'_0(t) \end{pmatrix} = J(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, l_n(t), \theta_n(t), a_n(t), \alpha_n(t)) * \begin{pmatrix} l'_0(t) \\ \cdot \\ \cdot \\ \cdot \\ \alpha'_n(t) \end{pmatrix}$$

Ahol J $3 \times 4n$ dimenziójú Jacobi-mátrix, melyre

$$J_{ij} = \frac{\delta}{\delta x_j} f_i(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, l_n(t), \theta_n(t), a_n(t), \alpha_n(t))$$

6. Megjegyzés. Az előző megjegyzéshez hasonlóan J mátrixot is elég egyszer kiszámolni f_1, f_2, f_3 függvények ismeretében.

Ezzel a direkt kinematikai feladatot megoldottuk, hiszen a jobb oldalon a konkrét paraméterek és függvények ismeretében minden számolható, a szorzás elvégezhető.

Példa f és J meghatározására

Lássuk a direkt kinematikai feladatnál látott robotkar esetében a sebességkinematikai feladat megoldását!

Legyen $\theta_0(t)$ és $\theta_1(t)$ a paraméterek időbeli változását leíró függvény, l_0, l_1 konstansok. Ekkor a korábbi eredményeket felhasználva

$$\begin{pmatrix} v_0(t) \\ w_0(t) \\ 1 \end{pmatrix} = \begin{pmatrix} l_0 * \cos\theta_0 + l_1 * \cos(\theta_0(t) + \theta_1(t)) \\ l_0 * \sin\theta_0 + l_1 * \sin(\theta_0(t) + \theta_1(t)) \\ 1 \end{pmatrix}$$

azaz

$$f_1(l_0; \theta_0(t); l_1; \theta_1(t)) = l_0 * \cos\theta_0(t) + l_1 * \cos(\theta_0(t) + \theta_1(t))$$

$$f_2(l_0; \theta_0(t); l_1; \theta_1(t)) = l_0 * \sin\theta_0(t) + l_1 * \sin(\theta_0(t) + \theta_1(t))$$

Ebből a J mátrix a következő:

$$J(l_0; \theta_0(t); l_1; \theta_1(t)) = \begin{pmatrix} c_1 & -l_0 * \sin\theta_0(t) - l_1 * \sin(\theta_0(t) + \theta_1(t)) & c_3 & -l_1 * \sin(\theta_0(t) + \theta_1(t)) \\ c_2 & l_0 * \cos\theta_0(t) + l_1 * \cos(\theta_0(t) + \theta_1(t)) & c_4 & l_1 * \cos(\theta_0(t) + \theta_1(t)) \end{pmatrix}$$

c_i elemeket nem kell kiszámolnunk, mivel a hozzájuk tartozó értékek a változók deriváltját tartalmazó vektorban 0, hiszen konstansokat deriválunk. Így tehát a kéz sebességére a következőt kapjuk:

$$v(t) = \begin{pmatrix} -l_0 * \sin\theta_0(t) - l_1 * \sin(\theta_0(t) + \theta_1(t)) & -l_1 * \sin(\theta_0(t) + \theta_1(t)) \\ l_0 * \cos\theta_0(t) + l_1 * \cos(\theta_0(t) + \theta_1(t)) & l_1 * \cos(\theta_0(t) + \theta_1(t)) \end{pmatrix} * \begin{pmatrix} \theta'_0(t) \\ \theta'_1(t) \end{pmatrix}$$

2.4. Inverz sebességkinematika

Ezúttal ismerjük a csuklók állását, ezáltal a direkt kinematikai feladat megoldhatósága alapján a kéz helyzetét is. Adott egy sebességvektor, amivel a kezét szeretnék elmozdítani, ehhez keressük a D-H paramétereket leíró függvények megfelelő deriváltját.

A direkt sebességkinematikai feladat megoldásául kapott egyenletben tehát a Jacobi-mátrixot továbbra is ismerjük, valamint adott a kívánt sebességvektor a bal oldalon. Figyelembe kell azonban vennünk, hogy a $4n$ változónk közül előre meghatározott $3n$ biztosan 0, hiszen konstans függvény deriváltjaként írhatjuk föl őket. Legyen $q_i, i \in \{0 \dots n\}$ a nemnulla változók, J' pedig a nekik megfelelő oszlopok J mátrixból. Ekkor a $J' * q = v$ egyenletrendszer kell megoldanunk, ahol v a kívánt sebességvektorunk. Amennyiben van megoldás, ezt egy egyszerű Gauss-eliminációval megtehetjük, majd a megoldás q értékeit deriváltnak választva a kéz a kívánt sebességvektorral mozog.

7. Megjegyzés. Egy ilyen feladat numerikus módszerek segítségével megoldható, de ennek tárgyalása nem tartozik ehhez a dolgozathoz.

2.5. Szingularitások

Azonban bizonyos helyzetekben előfordulhat, hogy a $J' * q = v$ egyenletnek az adott v vektorra nincs megoldása. Ez azt jelenti, hogy a robotkart adott helyzetből nem lehet a kívánt sebességvektorral elindítani.

Definíció. Szinguláris konfigurációknak nevezzük a robotkar azon helyzeteit, mikor

$$\varrho(J'(l_0, \theta_0, a_0, \alpha_1, \dots, l_n, \theta_n, a_n, \alpha_n)) < \dim(W)$$

azaz van olyan v vektor, mely irányába nem mozdítható a robotkar.

Az összes szinguláris konfiguráció megkeresése nehéz feladat. Algebrailag a

$$\varrho(J'(l_0(t), \theta_0(t), a_0(t), \alpha_0(t), \dots, l_n(t), \theta_n(t), a_n(t), \alpha_n(t))) < \dim(W)$$

feladatot kell megoldanunk, azonban ez már nemlineáris feladat. Lehetőség van szingularitások geometriai megkeresésére is, azonban ez csupán az adott robotkar felépítésének ismeretében lehetséges, általánosan nem.

A probléma nem tartozik ezen dolgozat fő témái közé, így a megoldás kereséstől ezúttal nehézsége miatt eltekintünk, és a szinguláris konfigurációkat a továbbiakban ismertnek tekintjük.

Példa szingularitásra

Tekintsük ismét a jól bevált robotkarunkat példának. Az előző példában meghatároztuk, hogy

$$J'(\theta_0; \theta_1) = \begin{pmatrix} -l_0 * \sin\theta_0 - l_1 * \sin(\theta_0 + \theta_1) & -l_1 * \sin(\theta_0 + \theta_1) \\ l_0 * \cos\theta_0 + l_1 * \cos(\theta_0 + \theta_1) & l_1 * \cos(\theta_0 + \theta_1) \end{pmatrix}$$

Jelen esetben egy 2x2-es mátrixszal könnyű dolgunk van, hiszen rengeteg karakterizációja van annak, hogy ez nem teljes rangú. Jelen esetben mi az oszlopvektorok lineáris függetlenségét vizsgáljuk, azaz csak akkor szinguláris egy pozíció, ha

$$c * \begin{pmatrix} -l_0 * \sin\theta_0 \\ l_0 * \cos\theta_0 \end{pmatrix} = \begin{pmatrix} -l_1 * \sin(\theta_0 + \theta_1) \\ l_1 * \cos(\theta_0 + \theta_1) \end{pmatrix}$$

Ez pedig csak akkor teljesül, ha $\theta_0 = \theta_0 + \theta_1 + k * \pi \Rightarrow \theta_1 = k * \pi$, azaz ha a második csukló teljesen ki van nyújtva, vagy teljesen visszahajlik. Ezek tehát ennek az egyszerű robotkarnak a szinguláris helyzetei.

2.6. Inverz kinematikai feladat - ismét

Most, hogy már ismerjük a sebességkinematikai feladatok megoldását, visszatérhetünk az inverz kinematikai feladathoz.

Szeretnénk tehát tudni, hogy hogyan határozzuk meg a csuklók állását, hogy a robotunk keze a megfelelő helyre kerüljön. Ehhez tekintsünk egy kezdőpozíciót, melyben a kéz helyzete p_0 , és ismerjük a csuklók paramétereit. A célunk legyen p_1 pont. Ekkor vegyünk egy $\phi : [0; l] \rightarrow W$ differenciálható függvényt, mely teljesíti a $\phi(0) = p_0; \phi(l) = p_1$ feltételt, valamint elkerüli a szinguláris pozíciókat. Amennyiben ezt nem tudjuk megtenni, mert valamely végpont szinguláris, vagy a szinguláris pozíciók elválasztják a két helyzetet egymástól, úgy arra kell figyelni, hogy a görbe megoldható irányból haladjon át a szingularitásokon, hiszen ekkor a kijelölt egyenlet megoldható. Ekkor az inverz sebességkinematikai feladat minden pontra megoldható, azaz a kéz ϕ mentén való végigvezetéséhez egy lineáris differenciálegyenlet-rendszerhez tartozó kezdetiérték-problémát kell megoldanunk a nem állandó D-H paraméterekre. Ezt pedig meg tudjuk tenni, így az inverz kinematikai feladatot is megoldottuk.

8. Megjegyzés. *Ezzel a módszerrel egy lehetséges megoldáshoz jutunk. Különböző útvonalak különböző megoldásokhoz vezethetnek, azonban az összes lehetséges megoldás megtalálása nehéz feladat, annál is inkább, mivel végtelen sok is lehet belőlük; valamint a gyakorlati jelentősége kicsi.*

Ezen felül arra is rögtön megoldást találtunk, hogyan lehet két konfiguráció között útvonalat találni. Mitöbb, arra is tudunk ügyelni, hogy a kéz ne lépjen át bizonyos sebességet, gyorsulást, ami érzékeny tárgyak mozgatásakor elvárás lehet.

3. fejezet

Akadályok elkerülése - a konfigurációs tér

3.1. Bevezetés

Az előző részben megismert módszerek bár gyorsak és könnyen implementálhatóak, komoly hátrányuk, hogy csupán a kéz pozícióját követik a munkatéren. Amennyiben a munkatér nem üres - azaz nem minden konfiguráció megengedett, - úgy problémás eldönteni egy útról, hogy áthalad-e tiltott konfiguráción. Ennek a problémának a megoldására adunk egy hatékony módszert.

Az alapgondolat, mely vezérel minket, az alapprobléma hiányosságait küszöböli ki. Mielőtt elkezdenénk bármilyen útvonalkeresést, eldöntjük minden lehetséges konfigurációról, hogy megengedett-e az adott munkatéri akadályok mellett.

Definíció. *Akadálynak nevezzük a munkatér egy olyan ponthalmazát, mellyel a robotkarnak nem lehet közös pontja.*

9. Megjegyzés. *A továbbiakban akadályok alatt mindig zárt halmazokat értünk, amennyiben külön nem jelöljük. Ez bizonyos esetekben egyszerűsíti a számításokat, valamint összhangban van a gyakorlati alkalmazásokkor felmerülő esetekkel.*

Definíció. *Konfigurációs térnek nevezük a robotkar konfigurációinak halmazát, melyek előállhatnak egy üres (akadálymentes) munkatérben. Jelöljük ezt a halmazt C -vel.*

10. Megjegyzés. *A konfigurációs tér pontjai és a nekik megfelelő robotkar-állás munkatéri ponthalmaza közötti izomorfia okán az egyszerűség kedvéért néha azonos jelölést használunk a két dologra.*

Definíció. *Megengedett konfigurációknak nevezzük a konfigurációs tér azon részhalmazát, mely konfigurációk nem ütköznek akadályba a munkatéren.*

Definíció. *Tiltott konfigurációknak nevezzük a megengedett konfigurációk komplementerét a konfigurációs térre nézve, azaz az olyan konfigurációkat, melyek akadályba ütköznek a munkatéren.*

Tétel. Egy k darab forgató- és n darab eltoló ízületből álló robotkar konfigurációs tere homeomorf egy k dimenziós tórusz és $[0; 1]^n$ Descartes-szorzatával.

Bizonyítás. A csuklók paramétereinek definíciójából triviális. □

11. Megjegyzés. Mivel bármely két zárt intervallum homeomorf egymással (hiszen köztük invertálható lineáris leképezés adható), ezért időnként a $[0; 1]$ intervallum helyett az eltoló ízület tényleges mozgási tartományát fogjuk intervallumnak választani.

Az általánosság megsértése nélkül lehtőségünk van rá, hogy a robotkarok szegmenseit vastagság nélkül, térgörbéként kezeljük. Ugyanis amennyiben a robotkarunk legnagyobb vastagsága d , úgy az akadályokat kibővítjük a tőlük munkatéren legfeljebb d távolságra eső pontokkal. Így a redukált robotkar az új akadályokkal való ütközés nélkül az eredeti karunk sem ütközhet a régi akadályokba.

3.2. Direkt módszer

Ebben a módszerben egy adott felépítésű kar esetében az elemi geometria eszközeivel, illetve bonyolultabb esetben az 1. fejezetben megismert módszer segítségével határozzuk meg a tiltott konfigurációkat.

Általános megállapítások

Definíció. Akadály által generált tiltott konfigurációknak nevezzük a konfigurációs tér azon legbővebb részhalmazát, melynek minden eleme ütközik az adott akadállyal. Jelöljük ezt $G(A)$ -val.

Tétel. Legyen A és B akadály. Ekkor $G(A \cup B) = G(A) \cup G(B)$

Bizonyítás. Kétirányú tartalmazást bizonyítunk:

$x \in G(A \cup B) \Rightarrow \exists p \in A \cup B$, amire x konfiguráció ütközik p ponttal $\Rightarrow p \in A \vee p \in B \Rightarrow x \in G(A) \vee x \in G(B) \Rightarrow x \in G(A) \cup G(B)$

$x \in G(A) \cup G(B) \Rightarrow x \in G(A) \vee x \in G(B) \Rightarrow \exists p \in A \vee \exists q \in B$ amire x konfiguráció ütközik p , vagy q pontokkal $\Rightarrow \exists r \in A \cup B$, amire x ütközik r ponttal (p vagy q létezik és ilyen) $\Rightarrow x \in G(A \cup B)$. □

Tétel. Egy akadály és az akadály határa által generált tiltott konfigurációk azonosak.

12. Megjegyzés. Ez a tétel csak akkor igaz, ha a robot kezdőpontja nem esik bele egyik akadályba sem. Azonban ezt bátran feltehetjük, hiszen ez azt jelentené, hogy a robot minden konfigurációja tiltott.

Bizonyítás. A robotkarok folyonosságából és az akadályok zártságából triviális. □

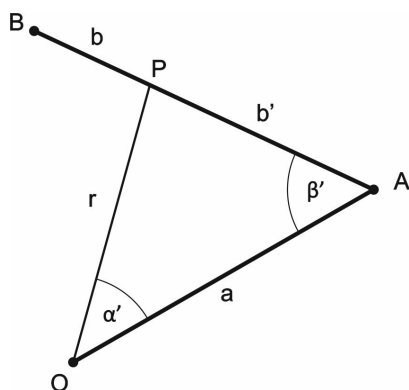
A fentiekből következik, hogy a konfigurációs tér tetszőlegesen pontosan meghatározható csupán szakaszok (síkban) és sokszögek (térben) által generált tiltott konfigurációk meghatározásával, valamint az akadályok poliéderekkel való közelítésével.

Ennél azonban általános esetben nem tudunk többet mondani, ugyanis az adott tiltott halmazok meghatározásának módja nagyban függ a robotkar tulajdonságaitól.

Egy egyszerű robotkar konfigurációs terének meghatározása

Tekintsük az a és b hosszú szakaszokból, mint szegmensekből álló síkbeli RR robotkارت. A két paraméter legyen α és β , az első szegmens x tengellyel bezárt irányított szöge, illetve a 2. szegmens 1. szegmennel, mint félegyenessel bezárt irányított szöge. Ennek a konfigurációs terét fogjuk meghatározni, méghozzá általánosan.

Először vizsgáljuk a munkatér egyetlen, p pontja által generált halmazát! Legyenek adottak p poláris koordinátái az x tengelyre nézve $(r; \gamma)$. Világos, hogy az első szegmens akkor ütközik p pontba, ha $r \leq a$ és $\alpha = \gamma$. Ekkor $\{(\alpha; \beta) | \alpha = \gamma\} \subset G(p)$. Amennyiben a második szegmens ütközik a p ponttal, akkor a célunk a robotkar második paraméterének (β) tiltott helyzetének meghatározása az első paraméter (α) függvényében. Tekintsük a következő ábrát:



Az összes esetet lefaktorizálhatjuk az OP tengelyre való tükrözés szerint, így dolgozhatunk α', β' irányítás nélküli, $[0; \pi]$ tartományba eső szögekkel, melyeket később visszafejtjük a keresett robotkar-paraméterekké.

A koszinusz-tételt alkalmazva adódik a következő: $b' = \sqrt{a^2 + r^2 - 2 * a * r * \cos \alpha'}$. Itt jegyezzük meg, hogy a $b' \leq b$ feltételből adódik $\cos \alpha' \geq \frac{a^2 + r^2 - b^2}{2 * a * r}$. Ismét a koszinusz-tétel segítségével, az előző eredményt felhasználva kapjuk $\cos \beta' = \frac{a - r * \cos \alpha'}{\sqrt{a^2 + r^2 - 2 * a * r * \cos \alpha'}}$ egyenletet.

Minden megtalált $(\alpha'; \beta')$ párhoz két $(\alpha; \beta)$ pont tartozik a konfigurációs téren, $(\gamma + \alpha'; \beta')$ és $(\gamma - \alpha'; 2\pi - \beta')$

Tehát a $p = (r; \gamma)$ pont generált tiltott konfigurációi, ha $r \leq a$,

$$G_1(p) = \{(\gamma; \beta) | \beta \in [0; 2\pi]\} \cup$$

$$\left\{(\gamma + \alpha'; \cos^{-1} \frac{a - r * \cos \alpha'}{\sqrt{a^2 + r^2 - 2 * a * r * \cos \alpha'}}) \mid \alpha' \in [0; \cos^{-1} \frac{a^2 + r^2 - b^2}{2 * a * r}]\right\} \cup$$

$$\left\{(\gamma - \alpha'; 2\pi - \cos^{-1} \frac{a - r * \cos \alpha'}{\sqrt{a^2 + r^2 - 2 * a * r * \cos \alpha'}}) \mid \alpha' \in [0; \cos^{-1} \frac{a^2 + r^2 - b^2}{2 * a * r}]\right\}$$

$r > a$ esetben pedig

$$G_2(p) = \left\{(\gamma + \alpha'; \cos^{-1} \frac{a - r * \cos \alpha'}{\sqrt{a^2 + r^2 - 2 * a * r * \cos \alpha'}}) \mid \alpha' \in [0; \cos^{-1} \frac{a^2 + r^2 - b^2}{2 * a * r}]\right\} \cup$$

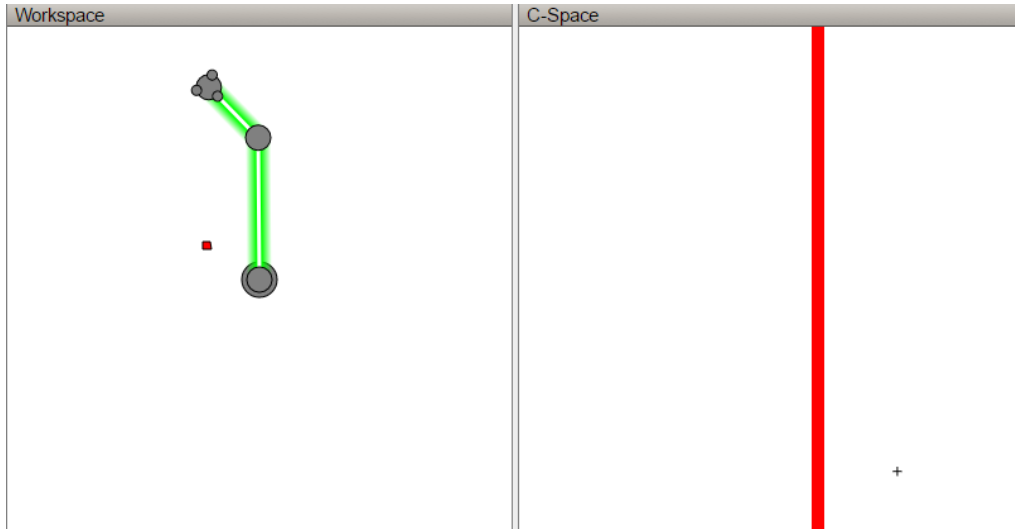
$$\left\{(\gamma - \alpha'; 2\pi - \cos^{-1} \frac{a - r * \cos \alpha'}{\sqrt{a^2 + r^2 - 2 * a * r * \cos \alpha'}}) \mid \alpha' \in [0; \cos^{-1} \frac{a^2 + r^2 - b^2}{2 * a * r}]\right\}$$

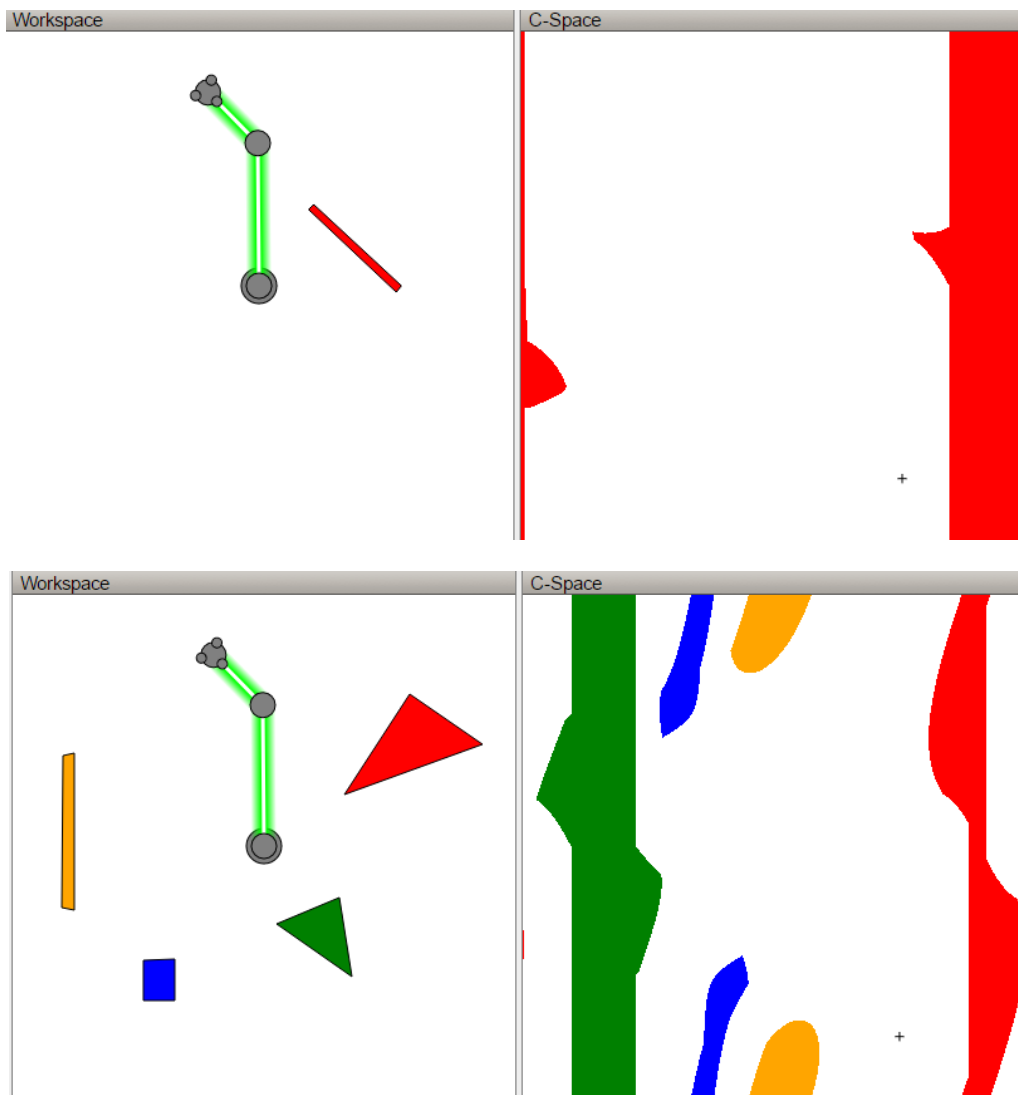
Ezzel tehát a munkatér tetszőleges, poláris koordinátákkal adott pontjához meg tudjuk adni a tiltott konfigurációkat. Így adott A akadály esetén nincs más dolgunk, mint meghatározni azokat a $\phi_{1,2} : [0; 1] \rightarrow W$, poláris koordinátákkal megadott függvényeket, melyekre $\{\phi_1(t) \mid t \in [0; 1]\} = \delta A \cap K_a$ és $\{\phi_2(t) \mid t \in [0; 1]\} = \delta A \cap (W - K_a)$, ahol K_a az a sugarú, origó középpontú zárt körlap. Ekkor az akadály által generált tiltott konfigurációk halmaza

$$G(A) = \{G_1(\phi_1(t)) \cup G_2(\phi_2(t)) \mid t \in [0; 1]\}.$$

13. Megjegyzés. Vegyük észre, hogy amennyiben a robotkar RR helyett RPRP típusú, és az eltoló csuklók tengelye egybeesik az előtte és utána lévő szegmenssel, úgy a fenti metódus érvényes marad. Ekkor ugyanis az a és b paraméter helyett a prizmatikus csuklók paramétere szerinti függvények lesznek, melyeket beilleszthetünk a fenti képletbe, így megkaphatjuk a 4 dimenziós konfigurációs teret.

Néhány példa akadályokra és konfigurációs térre:





3.3. Diszkrét konfigurációs tér felépítése

Néhány esetben a robotkar összetettsége okán nem célszerű az előző fejezetben látott módszer, hiszen minden szegmens újabb dimenziót ad a konfigurációs térhez, valamint a rotációs csuklók számának növekedésével megnőhet a szükséges esetszétválasztások száma. Más esetekben a precizitás elé helyezzük azt, hogy a konfigurációs tér algoritmikusan meghatározható legyen.

Ezekre a problémákra adhat megoldást, ha nem az összes konfigurációt vizsgáljuk, csupán megfelelően kiválasztva véges sokat. Jelölje ezen konfigurációk halmazát C' . Ekkor a direkt módszerben látott, inverz kinematikai feladathoz hasonló számítások helyett a konfigurációs tér kiválasztott pontjain végzünk el egy kiterjesztett direkt kinematikai feladatot, mely könnyebben algoritmizálható, és számítási igényt tekintve is jelentősen kedvezőbb.

A továbbiakhoz szükséges bizonyos fajta távolságokat definiálnunk.

Definíció. *Munkatéri távolságon, illetve robotkarok távolságán a robotkar két konfigurációjának, mint a munkatér halmazainak az alábbi értelemben vett távolságát értjük. $d_W(A; B) = \max(\sup_{p \in A} \inf_{q \in B} d(p, q); \sup_{p \in B} \inf_{q \in A} d(p, q))$, ahol $d(p, q)$ a p és q pontok euklidészi távolsága a munkatéren.*

Ez a távolság azért hasznos, mert ekkor ha egy konfiguráció a munkatéren d távolságra van egy akadálytól, mint két ponthalmaz szokásos euklidészi távolsága, akkor az így értelmezett munkatéri távolság szerint hozzá d -nél közelebb lévő konfigurációk sem ütköznek az adott akadályba.

14. Megjegyzés. *Esetünkben a konfigurációk zárt halmazok a munkatéren, így az infimum és szuprérum felvételük valamely pontokra.*

15. Megjegyzés. *Az euklidészi távolság hasonló tulajdonságaiból világos, hogy ez a távolság nemnegatív és szimmetrikus.*

Tétel. *A d_W távolságra igaz a háromszög-egyenlőtlenség, azaz $d_W(A; B) \leq d_W(A; Z) + d_W(B; Z)$*

Bizonyítás. Tudjuk, hogy pontonként igaz az euklidészi távolságra. A $d_W(A; B)$ távolság vételesen föl a maximum első tagjában, $p_0 \in A, q_0 \in B$ pontokon. $\inf_{w \in Z} d(p_0; w)$ vételesen föl w_0 pontban $\inf_{q \in B} d(w_0; q)$ pedig q_1 pontban. Ekkor tudjuk, hogy $d(p_0; q_1) \geq d(p_0; q_0)$, és a háromszög-egyenlőtlenség miatt $d(p_0; q_1) \leq d(p_0; w_0) + d(w_0; q_1)$. Ekkor a szuprérumok és maximumok miatt:

$$d_W(A; B) = d(p_0; q_0) \leq d(p_0; q_1) \leq d(p_0; w_0) + d(w_0; q_1) \leq d_W(A; Z) + d_W(B; Z)$$

□

Definíció. *Konfigurációs téri távolságon, vagy röviden konfigurációk távolságán a két konfigurációnak megfelelő pontok Euklidészi távolságát értjük a konfigurációs téren. Jele $d_C(A; B)$.*

16. Megjegyzés. *Természetesen a rotációs csuklóknak megfelelő koordináták esetén nem $|A_i - B_i|$ -vel, hanem $\min_{k \in \mathbb{N}} |A_i + k * 2\pi - B_i|$ -vel számolunk, mint egy dimenziós távolság, ahol A_i az i . csuklóparaméter.*

Tétel. *Tetszőleges robotkar esetén $\exists c$ konstans, melyre $\forall A, B \in C$ esetén $d_W(A; B) \leq c * d_C(A; B)$*

Bizonyítás. Legyen a robotkar csuklóinak száma k , a maximális hossza (a prizmatikus csuklók állásának függvényében) R , a prizmatikus csuklókat paraméterezzük a valós mozgástér szerint. Legyen $A^0 \dots A^k$ konfigurációk olyan sorozata, melyre $i > j$ esetén $A_i^j = A_i$; $i \leq j$ esetekben pedig $A_i^j = B_i$, azaz egyesével kicseréljük

a koordinátákat. Ekkor $A = A^0$; $B = A^k$ és a háromszög-egyenlőtlenség miatt $d_W(A; B) \leq \sum_{j=1}^k d_W(A^{j-1}; A^j)$. Most $d_W(A^{j-1}; A^j)$ -re szeretnénk becslést adni, B_j és A_j eltérése alapján.

Amennyiben a j . koordinátának megfelelő csukló eltoló, úgy világos, hogy $d_W(A^{j-1}; A^j) \leq |B_j - A_j|$, hiszen a többi koordináta azonos, ez pedig csak néhány pontot tol el az adott hosszúságú vektorral.

Ha a j . koordináta forgató, akkor $d_W(A^{j-1}; A^j) \leq R * \frac{\min_{k \in \mathbb{N}} |B_j + 2 * \pi - A_j|}{2 * \pi}$, hiszen itt forgatásról van szó, ahol minden pont által megtett út felülről van becsülve a jobb oldalon.

Ekkor $d_W(A; B) \leq k * \max(\frac{R}{2 * \pi}; 1) * \sum_{j=1}^k d(A_j; B_j) = k * \max(\frac{R}{2 * \pi}; 1) * |A - B|_1$. Mivel tudjuk, hogy az egyes és Euklidészi normák ekvivalensek [6], és d_C sima euklidészi norma, így készen vagyunk. \square

Felosztások meghatározása

Természetesen adódik a kérdés, hogy hogyan válasszuk meg C' -t, mely segítségével jó képet kaphatunk a konfigurációs térről.

Hasonlóan természetes válasz a konfigurációs tér egyenletes felosztása. Ekkor valamilyen k természetes számot választva a konfigurációs teret k^d darab, k dimenziós, egybevágó téglára bontjuk, és a csúcspontokat választjuk C' -nek. Ekkor $k \rightarrow \infty$ esetén szomszédos $A, B \in C'$ csúcsokra $d_C(A; B) \rightarrow 0 \Rightarrow d_W(A; B) \rightarrow 0$, azaz a felosztás finomításával a munkatéren értelmezett távolság is konzisztensen csökken.

17. Megjegyzés. *Szomszédos csúcsok alatt olyan konfigurációkat értünk, melyek között van közvetlen éle valamely téglának.*

18. Megjegyzés. *Téglarács helyett természetesen használhatunk szimplerárcsot, vagy egyéb k dimenziós térrárcsot, de jelen dolgozatban ezzel nem foglalkozunk részletesen.*

Az egyenletes felosztás alapelve az volt, hogy így a kiválasztott konfigurációk távolsága a konfigurációs téren egyenletesen csökken a kiszámolt pontok számának növelésével. Azonban nekünk kedvezőbb, ha a munkatéri távolság csökken egyenletesen, így spórolva számítási kapacitást. Ehhez továbbra is egyenletes felosztást fogunk használni, azonban a különböző csuklókhöz tartozó dimenziók szerint különböző finomságú felosztásokat vetünk be.

Legyen a célunk, hogy bármely dimenzióban két szomszédos konfiguráció távolsága a munkatéren legfeljebb ϵ legyen, a lehető legkevesebb kiszámított pont mellett.

Nézzük a prizmatikus csuklókhöz tartozó paramétereiket! Itt világos a dolgunk, hiszen a csukló robotkarban való elhelyezkedésétől függetlenül egy adott nagyságú eltolásról van szó (bár más lehet az érintett pontok halmaza). Így itt az ϵ távolság tartásához $\lceil \frac{r}{\epsilon} \rceil$ darab osztópontra van szükség, ahol r a csukló mozgásának terjedelme.

Rotációs csuklók esetén forgatásról van szó, azaz itt számít a pontok távolsága a forgatás középpontjától. Legyen R a csukló után következő szegmensek hosszának

maximuma (a prizmatikus csuklók függvényében). Ekkor az ϵ elmozdulást biztosan nem lépjük túl $\lceil \frac{2*\pi*R}{\epsilon} \rceil$ osztópont alkalmazásával.

19. Megjegyzés. *Húrok hossza helyett ívhosszal számolva felülről becsüljük a szükséges felosztások számát, hogy egyszerűbb legyen a számítás. Valamint R és ϵ gyakorlatban tapasztalt aránya miatt a két hossz közötti eltérés kicsi.*

Lehetőségünk van arra is, hogy az előző módszerek valamelyikével kapott C' halmazhoz további konfigurációkat adjunk, ahol szükségét érezzük. Miután meghatározzuk C' minden eleméről, hogy tiltott vagy megengedett, vizsgáljuk meg a tégláinkat: lesznek *megengedett* téglák, melyek minden csúcsa megengedett, *tiltott* téglák, melyek minden csúcsa tiltott, illetve *vegyes* téglák, melyeknek nem minden csúcsa azonos. Feltételezvé, hogy a tiltott és megengedett téglák összes belső pontja is adott típusú (ennek vizsgálatát a következő szakaszban végezzük), elég a vegyes téglákon ismételtelen egyenletes felosztással újabb konfigurációkat venni C' -höz, így kevesebb számítással érhetünk el azonos pontosságú diszkrét konfigurációs teret.

Túl kicsi akadályok problémája

A diszkrét konfigurációs tér elkészítése után élünk azzal a feltevéssel, hogy a megengedett téglák minden belső pontja is megengedett, hiszen így tehetjük meg, hogy a diszkrét pontokat összekötve útvonalat találunk két konfiguráció között. Az előző módszerben láttuk, hogy ha tudunk egy akadály létezéséről, akkor annak határának jobb közelítésére van lehetőségünk a teljes konfigurációs tér finomabb felosztása nélkül.

Azonban megtörténhet, hogy az eredetileg meghatározott, ϵ munkatéri finomságú diszkrét konfigurációs téren egy pontszerű akadály által generált tiltott konfigurációk közül egyik sem jelenik meg, így a C' alapján minden konfigurációt megengedettnak ítélnénk.

Ennek elkerülésére van lehetőségünk, ha az ϵ finomság kijelölése után az akadályokat egy ϵ széles zónával kibővítjük. Így biztosan minden akadálynak nyoma lesz C' elemei között, és a további finomítást ezen akadályok mellett végezzük el. Ezek után az eredeti akadályokkal kiszámítjuk az új felosztásra a megengedettséget, majd szükség esetén újabb finomításokat végzünk.

20. Megjegyzés. *Hasonlóan járhatunk el, ha tiltott téglák homogenitásáról akarunk meggyőződni. Ekkor az akadályokat ϵ -al kisebbítjük, és végezzük el az előbb vázolt lépéseket.*

Konfiguráció megengedettségének eldöntése

Miután már meg tudjuk határozni, hogy mely konfigurációkra kell megvizsgálni, hogy akadályba ütköznek-e, ideje magára az eldöntésre is algoritmust adnunk.

Először rögzítsük minden szegmens 3 különböző pontját, hiszen ezek meghatározzák a térbeli helyzetét. Gyakorlati szempontból előnyös a két végpontot és egy tetszőleges

belső pontot választani. Most paraméterezzük a szegmenst ezen három pont helyzetének függvényében. Ezek után minden pontra meghatározzuk azt a robotkart, melyet a pont utáni részek elhagyásával kapunk. Ekkor az első fejezetben megismert Denavit-Hartenberg módszer segítségével tetszőleges konfigurációra meg tudjuk mondani a kijelölt pontok helyzetét, amit a paraméteres alakba helyettesítve tetszőleges konfigurációra megkapjuk a hozzá tartozó összes, munkatéri pontot, melyet a robotkar elfoglal. Ekkor az akadályok halmazával való diszjunktság vizsgálatával eldönthető, hogy a konfiguráció megengedett, vagy sem.

21. Megjegyzés. *Amennyiben egy szegmensek egyenes szakasz, úgy természetesen 3 helyett elég 2 pontot rögzíteni rajta, és az alapján paraméterezni.*

22. Megjegyzés. *Forgató csuklóval elválasztott szegmensek esetében választhatjuk a közös végpontot, mint tartópontot, ezzel számítási időt spórolva.*

4. fejezet

Útkeresés konfigurációs téren

Ebben a fejezetben adottnak tekintjük a robotkar konfigurációs terét, beleértve a megengedett és tiltott konfigurációk halmazának ismeretét. Ennek fényében szeretnénk megengedett útvonalat keresni két konfiguráció között, mely nem ütközik akadályba.

A kezdeti konfiguráció általában adott, azonban célként gyakran a kéz egy adott helyzete van megadva. Ekkor az inverz kinematikai feladat segítségével meghatározzuk egy hozzá tartozó konfigurációt, és ezt tekintjük célnak.

Előfordulhat, hogy a kéz egy pozíciójához több konfiguráció is tartozik (ahogyan a 2. fejezetben láthattuk), és ezek közül nem mind érhető el megengedett úton a kezdőpontból. Ekkor nincs más megoldás, mint másik útvonalat választani az inverz kinematikai feladatban, ezzel próbálva új cél konfigurációhoz jutni.

A továbbiakban feltételezzük, hogy egyértelműen adott a kezdeti- és cél konfiguráció is.

4.1. Diszkrét eset

Amennyiben diszkrét konfigurációs terünk van, úgy az útkeresés 3 részből áll: út keresése a kezdőpont és egy C' beli pont között, út keresése a végpont és egy C' beli pont között, valamint út keresése C' két adott eleme között.

Útkeresés C' -ig

Nem vizsgáljuk külön a kezdő- és végpont problémáját, hiszen irányítás nélküli utakat keresünk, így a két feladat általános megoldása azonos.

Amennyiben a pontunk egy megengedett téglában van, úgy a téglát tetszőleges csúcsába vezető, téglából nem kilépő görbe megengedett, a "Túl kicsi akadályok problémája" című szakasz szerint.

Amennyiben vegyes téglában van, úgy határozzuk meg a konfiguráció és az akadályok távolságát a munkatéren, majd ezt a távolságot a felosztás finomságának választva

finomítjuk a pontunkat tartalmazó téglán C' -t. Így biztosak lehetünk benne, hogy az új, kibővített konfigurációs terünkben már megengedett téglá tartalmazza a kezdőpontot.

Útkeresés C' pontjai között

Gráfot készítünk C' megengedett pontjaiból, mint csúcsokból. Két csúcs között van él, ha mindkettő ugyan annak a megengedett téglának a csúcsai, vagy ha mindkettő megengedett, és közöttük fut egy téglá éle. Ekkor ha találunk egy utat a gráfon, annak egyértelműen megfelel egy görbe a konfigurációs téren, melynek minden pontja megengedett.

Adott tehát egy egyszerű gráfunk, ezen a gráfon szeretnénk utat találni két csúcs között, amire a továbbiakban három algoritmust is adunk.

Szélességi keresés

Ez az algoritmus lehetőséget biztosít egy gráf összefüggő komponensébe tartozó csúcsok szisztematikus bejárására, egyben ad egy élszám szerinti legrövidebb utat egy kijelölt kezdőcsúcsból. Lássuk, hogyan működik!

Kiindulunk a kezdőcsúcsunkból, ez az egyetlen csúcs, mely 0 éllel elérhető, ő lesz a H_0 halmaz egyetlen eleme. Első lépésként meglátogatjuk a kezdőcsúcsunk összes szomszédját, ezek érhetőek el 1 éllel a kezdőcsúcsból, ők alkotják H_1 halmazt. A k . lépés során tekintjük a H_{k-1} halmaz szomszédai közül azokat, melyek nem elemei a $\bigcup_{i=0}^{k-1} H_i$ halmaznak, őket tekintjük H_k halmaznak, valamint mindegyikükhöz feljegyezzük, hogy H_{k-1} mely elemével szomszédosak (ha többel, akkor egy tetszőlegeset közülük).

Az algoritmus véget ér, ha a célul kitűzött csúcs bekerül egy H_n halmazba, vagy ha nincs több eddig be nem járt, szomszédos csúcs. Ekkor első esetben a feljegyzett szomszédos csúcsok mentén haladva kapunk egy n élből álló utat a kezdőcsúcshoz; utóbbi esetben pedig megállapítottuk, hogy nincs út a két kijelölt csúcs között.

23. Megjegyzés. *Az algoritmusunk véges csúcshalmaz esetén véges (és nekünk véges halmazunk van), legfeljebb csúcsszámnyi lépést tesz.*

Dijkstra-algoritmus

Nem elégszünk meg azonban ennyivel, ugyanis a felosztás finomsága eltérő a konfigurációs tér egyes részein, így az élszám szerinti legrövidebb út markánsan eltérhet az euklidészi értelemben vett hossz szerinti legrövidebbtől. Így rendeljük minden élhez a végpontjaiknak megfelelő konfigurációk távolságát a konfigurációs téren. Legyen ez a c függvény.

Dijkstra algoritmus a mohó algoritmus, mely a szélességi keresést alapul véve legrövidebb utat talál két csúcs között, amennyiben az élsúlyok nem negatívak. Az algoritmusához szükségünk van egy S halmazra, melyben azt jegyezzük, mely csúcsokhoz

ismerjük már a minimális utat, illetve S elemeihez tartozó d távolság-függvény, mely a kezdőcsúcstól való távolságot jelöli. Továbbá amikor egy csúcs bekerül az S halmazba, lejegyezzük, hogy melyik az őt megelőző csúcs a bizonyos legrövidebb útban.

Az algoritmus indulásakor S halmaz csupán a kezdőpontot tartalmazza, melynek a távolsága 0, megelőzője pedig nincs.

Egy lépésben tekintjük azokat a csúcsokat, amelyek egy él mentén elérhetőek S -ből, de maguk még nem elemei neki. Legyen ez a H halmaz. Közülük választjuk ki azt, melyre a

$$\min_{x \in H} \min_{y \in S; (x,y) \in E} (d(y) + c(x, y))$$

minimum felvételük. Ez kerül be új csúcsként S halmazba, y (szintén az, amelyen a minimum van) lesz a megelőzője, és a minimum értéke a $d(x)$ távolság.

Az algoritmus megáll, ha H üres, vagy a célcsúcsot választjuk ki. Előbbi esetben nincs út, utóbbi esetben a megelőzőkből álló út lesz a legrövidebb keresett út, $d(x)$ hosszal.

Tétel. *A Dijkstra-algoritmus valóban a legrövidebb utat találja meg a két kijelölt csúcs között.*

Bizonyítás. Használjunk indukciót. Kezdetben üres S , így minden benne lévő csúcshoz legrövidebb utunk van. Tegyük föl, hogy eddig minden csúcs legrövidebb úttal került be. Tegyük föl indirekten, hogy a most következő x az első csúcs, mely bekerül S halmazba, de nem a legrövidebb utat találtuk meg hozzá. Legyen a kezdőcsúcs u , ahonnan a legrövidebb utat keressük. Vizsgáljuk azt az állapotot, amikor majd x -et választjuk S elemei közé, w megelőzővel. Ekkor az x -hez vezető legrövidebb út u -ból biztosan kilép S -ből (lehet, hogy vissza is tér, mi most az u -tól számított első elhagyást vizsgáljuk), legyen az utolsó S -beli pont y , az első S -en kívüli z . Legyen a legrövidebb út z - x szakaszának hossza p .

Az indirekt feltevésünk azt jelenti, hogy $d(y) + c(y, z) + p < d(w) + c(w, x)$, hiszen az indukciós feltevés szerint eddig hibátlan volt az S halmaz, azaz y -ba és w -be vezető legrövidebb utak hosszát helyesen határozza meg d függvény. Viszont az algoritmus beválogató-feltételéből következik, hogy $d(y) + c(y, z) \geq d(w) + c(w, x)$, valamint $c \geq 0$ feltételből következik $p \geq 0$, így $d(y) + c(y, z) + p \geq d(w) + c(w, x)$, ami ellentmondás, így az indukciós lépés igaz, tehát az állítást igazoltuk. \square

Heurisztika

Mint láttuk, a Dijkstra-algoritmus a szélességi keresés egy javított, több információt nyújtó változata. Gráfelméleti értelemben nem javítható tovább, azonban nekünk rendelkezésünkre áll további információ, hiszen a konfigurációs térből származtatott gráfunk speciális. Még hozzá abból a szempontból, hogy ismerjük a célcsúcs *irányát*, és konfigurációs téren való távolságát (ami nem azonos a gráfban található minimális úttal). Ezt az információt felhasználva szeretnénk javítani Dijkstra algoritmusán.

A következőkben bemutatott módszer A^* algoritmus néven vált ismertté.

Bevezetünk egy heurisztikus függvényt a gráfunk csúcsaira, legyen ez $h(v)$. Minden csúcshoz hozzá fogjuk rendelni a célnak megfelelő konfigurációtól vett távolságát a konfigurációs téren, így $h(v) = d_C(v; p_1)$.

A Dijkstra-algoritmus egyetlen ponton módosul, az S halmazba való beválasztási feltételnél, itt ugyanis

$$\min_{x \in H} \min_{y \in S; (x,y) \in E} (d(y) + c(x, y))$$

helyett

$$\min_{x \in H} \min_{y \in S; (x,y) \in E} (d(y) + c(x, y) + h(y))$$

minimumhelyét választjuk be, azonban a kezdőcsúctól való távolság természetesen változatlanul marad rögzítésre. Ezzel pontosan a kívánt hatást érjük el, azaz az algoritmus korábban kísérletezik a cél irányába eső csúcsokon, mint eltérő irányokban.

Definíció. Egy h függvényt elfogadható heurisztikának nevezünk, ha

$$\forall v \in V, h(v) \leq c_{min}(v, p_1),$$

azaz alulról becsüli a gráfban található legrövidebb út hosszát.

Tétel. Amennyiben $h(v)$ elfogadható heurisztika, úgy az A^* algoritmus is a legrövidebb utat találja meg.

24. Megjegyzés. Az előző tétel bizonyítása összetett, így itt eltekintünk az ismeretétől, az megtalálható a Russell, Norvig jegyzetben [3].

4.2. Folytonos eset

Itt két különböző módszert fogunk megismerni, mely segítségével megengedett utat találhatunk a konfigurációs tér két adott pontja között.

Gradiens-módszer

A módszer alapját az az ötlet adja, hogy kezeljük a robotot egy pontszerű részecskeként a konfigurációs téren, melyen felépítünk egy mesterséges potenciált. Ez a potenciál vonzani fogja a robotot a cél felé, és taszítani az akadályoktól, miközben ideális esetben egyetlen minimum helye van, éppen a célban.

A két feltételünk teljesítéséhez adódik, hogy az U potenciál álljon elő $U = U_v + U_t$ alakban, ahol U_v a célponthoz való vonzást, U_t pedig az akadályok taszítását adja.

Ekkor a részecskére ható erőt az $F(q) = -\nabla U(q) = -\nabla U_v(q) - \nabla U_t(q)$ adja.

25. Megjegyzés. Látható, hogy az összegalakban való keresés előnye szorzat, hatvány vagy egyéb forma helyett a deriválás és összeadás felcserélhetősége is.

Vonzó potenciál

Mit várunk el vonzó potenciáltól?

1. Monoton nőjön a q_0 ponttól való távolság függvényében.
2. Mindenhol folytonosan deriválható.
3. A gradiens hossza a cél közelében kisebb, mint a céltól messze (így dominánssá téve a közeledést a kezdeti fázisban)

Ezen feltételeknek tökéletesen megfelel a

$$U_v(q) = \frac{1}{2} * c_v * d_C(q; q_0)^2$$

függvény, ahol c_v skálázásra szolgáló paraméter. Ekkor a részecskére ható erő

$$\nabla U_v(q) = c_v * (q - q_0)$$

Ennek a skálázási paraméternek a szerepe a konfigurációs tér nagyságának, illetve összetettségének a "hányadosáról" van szó, miszerint nagyobb térre nagyobb paraméter ideális a közelítés sebessége szempontjából, összetettebb térre viszont kisebb, az egyenesebb útvonal érdekében lehet szükség.

Taszító potenciál

Itt is van néhány dolog, amit elvárunk a potenciáltól:

1. Ne engedje a robotot akadályba ütközni, bármekkora is a vonzó potenciál.
2. Folytonosan differenciálható legyen.
3. Az akadálytól távolodva minimális, vagy 0 hatása legyen.

Az első és harmadik feltételt kielégíthetjük, ha

$$U_t^A(q) = \frac{1}{2} * c_t * \left(\frac{1}{d_C(q; A)} - \frac{1}{d_0} \right)^2,$$

ha $d_C(q; A) \leq d_0$, egyébként pedig $U_t^A(q) = 0$ alakban keressük, ahol c_t ismételten skálázó konstans, d_0 paraméter, hogy milyen távolságig hasson egy akadály, A tiltott konfigurációk halmaza (akadály), amely generálja ezt a potenciált. Így

$$\nabla U_t^A = c_t * \left(\frac{1}{d_C(q; A)} - \frac{1}{d_0} \right) * \frac{1}{d_C(q; A)^2} * \nabla d_C(q; A),$$

ahol

$$\nabla d_C(q; A) = \frac{q - b}{d_C(q; b)},$$

ha b az A halmaz q -hoz legközelebbi pontja. Ez abból következik, hogy a függvényünk lokálisan a b ponttól mért euklidészi távolság, tehát leggyorsabban a b -től egyenesen távolodva nő. Méghozzá mivel az adott $q - b$ irányú egyenes mentén a függvény lineáris, így a gradiens hossza 1, tehát csak le kell normálnunk a $q-b$ vektort.

Ennek a skálázó paraméternek első sorban az első paramétert kell kiegyenlítenie, hogy a két potenciál hasonló nagyságrendű legyen a kívánt részeken.

A folytonosan differenciálhatósághoz elég, ha az egyes halmazok (akadályok) konvexek, ugyanis ebben az esetben $d_C(q; A)$ folytonosan differenciálható a megengedett pontokra, míg konkáv esetben ez nem biztos. Tehát bontsuk föl a tiltott konfigurációkat konvex halmazokra, ezen halmazok halmaza legyen T . Ekkor

$$U_t(q) = \sum_{A \in T} U_t^A(q)$$

és

$$\nabla U_t(q) = \sum_{A \in T} \nabla U_t^A(q)$$

Az algoritmus

Van tehát egy módszerünk, mely létrehoz egy megfelelő potenciált a konfigurációs téren. Nézzük, hogy ez alapján hogyan jutunk el a célba!

Iterációt fogunk alkalmazni, a következő módon:

1. $i = 0$; p_0 -t a kezdőpontra állítottjuk.
2. Ha $d_C(p_i; q_0) \leq \epsilon$, akkor megállunk. Ha ϵ kisebb, mint a cél akadályoktól vett minimális távolsága, akkor innen egy egyenes szakasszal célba érhetünk.
3. Egyéb esetben $p_i = p_{i-1} + c_i * \frac{F(p_{i-1})}{\|F(p_{i-1})\|}$, ahol c_i a korábbi paraméterek, illetve a céltól való távolságtól függhet.

c_i értékeknek két szempontnak kell megfelelni: ne "ugorjunk" bele akadályba, és ne tartson túl sokáig az iteráció.

Ehhez $c_i = \frac{\min(d_0; d(p_{i-1}; q_0))}{2}$ általánosan megfelelő lehet, bár a gradiens hosszát nem veszi figyelembe.

Az algoritmusunk tehát megvan, csupán egy gond van vele, méghozzá az, hogy a potenciál lokális minimumaihoz be tud ragadni. Sajnos erre nincs jó módszer, hogy a potenciálunk lokális minimumoktól mentes legyen (többszörösen konkáv tiltott halmazok esetén mindenképpen probléma van), így más megoldást kell találnunk.

Véletlen séták

Először is észre kell vennünk, ha lokális minimumba, vagy közelébe értünk: amennyiben több lépés után közel kerülünk egy, már korábban érintett ponthoz, az szükséges és elégséges feltétele is az elakadásnak.

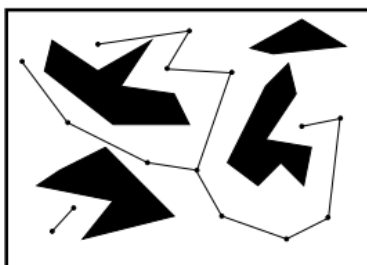
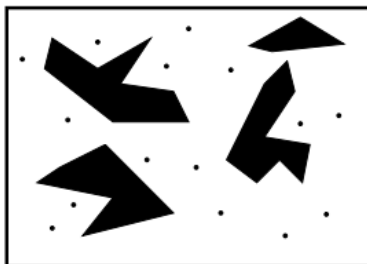
Sajnos ekkor eddig a modern robotikában megoldatlan problémához jutottunk: jelenleg nem ismerünk általánosan érvényes, nagy hatékonysággal működő módszert. A leginkább elterjedt megoldás szerint a minimum hely vonzaskörzetével nagyságrendileg megegyező várható hosszúságú véletlen sétát teszünk, közben figyelve az ütközések elkerülésére, majd a séta végpontjából újraindítjuk az algoritmust.

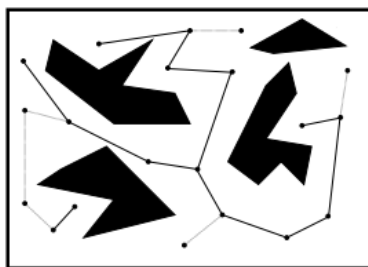
Előre eltárolt utak

Ezt a módszert az ihlette, hogy a robotkarjainknak az adott konfigurációs téren sokszor kell utat keresniük, melyek gyakran közel esnek egymáshoz. Így a diszkrét esthez hasonlóan itt is egy gráfot fogunk használni, mely csúcsai között ismerjük az ütközésmentes utakat. Így csupán a kezdő- és célpontunk és a hozzájuk legközelebb eső gráfcsúcs között kell minden alkalommal egy lokális útkeresést lefolytatni.

Véletlen gráf

A megengedett konfigurációk halmazából egyenletes eloszlás mellett véletlenszerűen kiválasztunk n elemet, ezek lesznek a gráfjaink csúcsai. Ezek után mindenden csúcsból a hozzá legközelebb lévő k csúcshoz egy szakaszt veszünk, és ellenőrizzük, hogy az megengedett-e. Ha igen, bevesszük élnek, ha nem, akkor eldobjuk. Így első lépésként kapunk egy gráfot, amely élei megengedett utak, azonban nem biztos, hogy összefüggő.





Második lépésként két lehetőségünk van: a független gráfrészek között felvehetünk további pontokat, így reménykedve út találásában, vagy gradiens módszerrel lokálisan utat keresni. Ekkor a vonzó potenciált nem egyetlen ponthoz, hanem az elérni kívánt teljes csúcshalmazhoz igazítjuk, hiszen nem számít, melyiket találjuk meg.

Ezt tudjuk addig folytatni, hogy összefüggő gráfot kapjunk, amennyiben a konfigurációs tér is összefüggő. Ezek után a diszkrét esetben látott elv érvényesül útkeresésnél, azaz gradiens módszerrel utat keresünk egy közeli csúcsba a kiinduló- és célkonfigurációból, majd a két csúcs között valamely keresési algoritmussal utat keresünk.

5. fejezet

Összefoglalás

Az előző fejezetek során először megismerkedtünk a robotkarokkal, majd megteremtettük a témakör matematikai tárgyalásához szükséges alapokat.

Megfogalmaztuk és megoldottuk a robotkarok helyzetét érintő alapvető kérdéseket, a direkt- és inverz kinematikai feladatokat. Megismertük a sebességkinematikát, mely lehetővé tette a robotkarok mozgását érintő kérdések megoldását, úgy mint a kéz útja, sebessége a mozgás során. Közben talákoztunk szinguláris pozíciókkal, melyekben az előbbi eredményeink nem mindig bizonyultak igaznak. Miközben megismerkedtünk az elméleti alapokkal, egy egyszerű példán keresztül a gyakorlati alkalmazásokra is rálátásunk nyílt.

Ezek után rátértünk a gyakorlatban jól ismert problémára: gyakran fordul elő, hogy a robotkar munkaterében akadályok vannak, melyekre az eddigi módszereink nem kínáltak kielégítő megoldást. Így bevezettük a konfigurációs tér fogalmát, mely megmutatja, hogy a robotkar mely helyzetei megengedettek és melyek nem az akadályok által. Vizsgáltuk diszkrét és folytonos eseteit, valamint hogy hogyan lehet őket geometriai módszerekkel pontosan meghatározni, illetve numerikus módszerekkel közelíteni.

Végül rátértünk a konfigurációs tereken való útkeresésre, melyhez diszkrét esetben gráfelméleti ismeretekre volt szükségünk, valamint folytonos esetekben a többváltozós analízisből ismert eljárások segítettek munkánkat.

Mindezek eredménye képpen egy robotkart tetszőleges akadályok között tudunk mozgatni ütközés nélkül.

További kutatásra is ad lehetőséget a téma, hiszen vannak területek, melybe itt nem volt lehetőségünk komolyabban belemenni. Ilyen lehet a szingularitások meghatározására irányuló vizsgálódások, az inverz kinematikai feladat differenciálegyenleteinek mélyebb tanulmányozása. Gyakorlati haszna lehet a sokszor előforduló robotkarok konfigurációs terének általános felépítése az egyszerű példánkhoz hasonlóan, vagy akár az analízis területére tévedve a gradiens- és a véletlen séták módszereit finomítani.

6. fejezet

Felhasznált irodalom

1. Hajós György, Bevezetés a geometriába. Budapest: Nemzeti Tankönyvkiadó, 2006. 594 pp. isbn: 963190116.
2. Spong Mark W., Hutchinson Seth, and Vidyasagar M., Robot Modeling and Control. New York: John Wiley and Sons, 2005. 496 pp. isbn: 9780471649908.
3. Stuart Russell, Peter Norvig, Mesterséges Intelligencia. Budapest: Panem, 2005. 1206 pp. isbn: 9789635454112
4. Laczkovich Miklós, T. Sós Vera, Analízis I. Budapes: 2012 Typotex. 591 pp. isbn: 978-963-2797-32-8
5. Laczkovich Miklós, T. Sós Vera, Analízis II. Budapes: 2013 Typotex. 432 pp. isbn: 978-963-2797-33-5
6. Sz.v. Fomin, A.n. Kolmogorov, A függvényelmélet és a funkcionálanalízis elemei. Budapest: 2012 Typotex. 556 pp. isbn: 9789632791197

7. fejezet

Köszönetnyilvánítás

Végezetül szeretnék köszönetet mondani azoknak, akik lehetővé tették számomra ezen sorok megírását.

Külön köszönettel tartozom témavezetőm, Szeghy Dávid felé, aki a szakdolgozatban nyújtott segítsége előtt, saját lelkesedésével a téma iránt megszerettette velem a matematika ezen területét.

Köszönettel tartozom családomnak és barátaimnak, hogy mindvégig támogattak.

Leipold Péter tanácsai és meglátásai nélkül ezen dolgozat nem készülhetett volna el jelen formájában.

Végezetül hálával és köszönettel tartozom Szabó Zsófi felé, aki folyamatos támogatása és ösztönzése mellett tolerálta munkamódszereimet is.