Kereken gördülő mobilis robotok irányítása

Márton Lőrinc

Bevezető

A mobilis robotok helyváltoztatásra képes irányított mechatronikai rendszerek. Amennyiben a mobilis robot robotkarral is el van látva, mobilis manipulátorról beszélünk.

A kerekeket általában villamos motorral hajtjuk meg, amelyeket egy célszámítógép irányít. Megfelelő szenzorokkal felszerelve a mobilis robotok képesek autonóm mozgást megvalósítani. A robot energiaellátását általában akkumulátorokkal valósítjuk meg.

A kerekekkel felszerelt mobilis robotok előnye, hogy viszonylag egyszerűen irányíthatóak (például a lábon járó robotokhoz képest) és energiahatékonyak. Hátrányuk, hogy komplexebb akadályok esetén nem képesek továbbhaladni (például lépcső esetén).

A legelterjedtebb irányítási feladatok közé tartozik a mobilis robot eljuttatása adott célpontba vagy annak biztosítása, hogy a robot egy síkbeli pályán végighaladjon.

Mobilis robotok fontosabb irányítási lehetőségei:

- *Teleoperáció* – távvezérlés – az irányító személy videó információ alapján irányítja a robot mozgását. A robot vagy a környezete képalkotó eszközökkel van felszerelve, amelyet továbbítani képes a felhasználóhoz. A felhasznló mozgással kapcsolatos utasításokat küld vissza a robotnak. Nem beszélhetünk autonóm robotról, hiszen a mozgás irányítását a feladat végrehajtásának minden pillanatában az ember végzi.

- *Előre megadott pálya követése* – A környezetben a robot előre megadott síkbeli pontokon kell áthaladjon. Ez megoldható Ponttól Pontig szabályozással, vagy a pontokra pályát fektethetünk, és pályakövetést valósíthatunk meg. A feladat végrehajtása során a robot ideiglenesen megállhat vagy letérhet a pályáról, amennyiben akadályt észlel, majd az akadály kikerülése vagy megszűnése után visszatér az előírt pályára. Az ilyen feladatoknál a robotot akadálydetektáló érzékelőkkel (például ultrahang vagy lézer alapú akadálydetektálók) szereljük fel az előre nem tervezett akadályok elkerülésére.

- *Autonóm mozgás* – Adott a cél koordinátája de a kiindulási pont és a cél közötti terep ismeretlen, a robot mozgás közben kell meghatározza a pályáját. A kiindulóponttól a célig való eljutás során a robot az érzékelő jelek alapján „feltérképezi” a terepet és határozza meg az utat a célhoz. A környezet érzékelésére és a robot lokalizálására használható érzékelők: inerciális mérőegység, mélységi képet szolgáltató kamerák vagy lézeres érzékelők, videokamera-képfeldolgozás, lézer vagy ultrahang alapú akadálydetektáló, GPS stb.

Mobilis robotok felépítése, modellezése

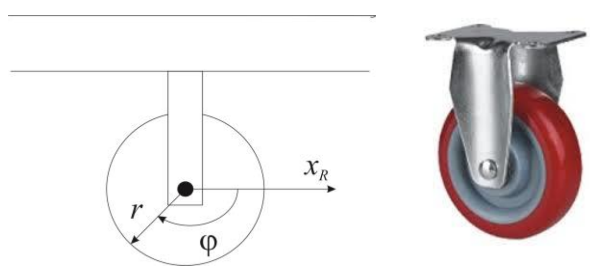
A kerekek

A mobilis robotok meghajtásának a legelterjedtebb módja a kerékmeghajtás. Ezen felül alkalmaznak még lánctalpas megoldást is, főleg nehéz terepen mozgó vagy mentési feladatokat végző robotok esetén.

A meghajtás szempontjából kétféle keréktípus különböztetünk meg: *aktív kerék* – meghajtott kerék, amely kapcsolva van a robot mozgását biztosító motorhoz, illetve *passzív kerék* – szabadon forog, csak a robot stabilitását, a földdel való állandó kontaktusát biztosítja.

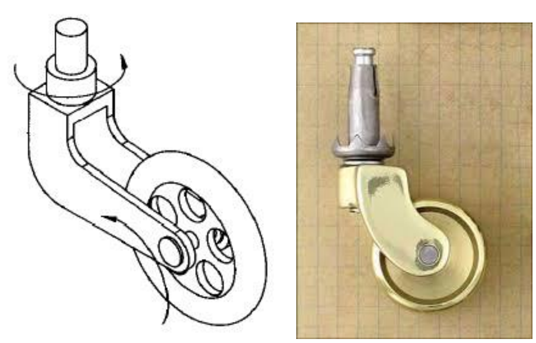
A gyakorlatban három keréktípus terjedt el:

A *fix kerék* esetén a keréktengely rögzítetve van a robot vázához. A kerék forgása a keréktengelyre merőleges elmozdulást biztosít, x_R mentén. Ha r a kerék sugara és a kerék szögsebessége $\dot{\varphi}$, akkor a kerék haladási sebessége $v = r\dot{\varphi}$. Az aktív kerekek általában ilyen típusúak.



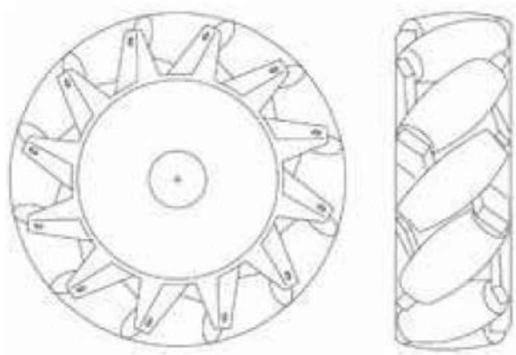
Fix kerék

A *Castor kerék* esetén a kerék tengelye elfordulhat a robot vázához képest. Ugyanakkor a kerék tengelye nem a felfüggesztési pont alatt helyezkedik el. Ezek a tulajdonságok megkönnyítik a mobilis robot irányváltoztatását, elfordulását. Általában passzív kerékként alkalmazzuk.



Castor kerék

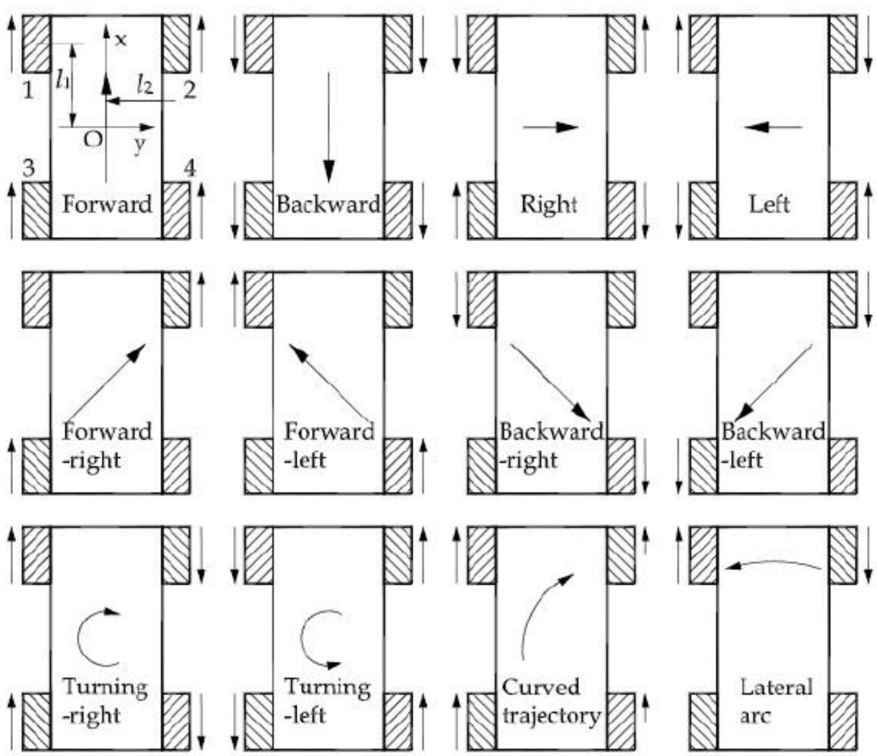
A Mecanum kerék segítségével omnidirekcionális (minden irányban instant módon elmozdulni képes) mobilis robotot valósíthatunk meg. A kerekekre passzív görgők vannak felszerelve általában 45 fokos szöggel elfordítva a haladási iránytól.



Mecanum kerék

Ezzel a kialakítással a kerék haladását biztosító erő egy része nem a fix kerekek haladási irányába, hanem az arra merőleges irányba hat mivel a görgőkön a súrlódási erő iránya nem egyezik meg a keréktengelyre merőleges haladási irányával.

Az omnidirekcionális mozgás megvalósításához a robot négy aktív kerékkel kell rendelkezzen. Az alábbi ábrán látható, hogy hogyan kell a Mecanum kerekeket felszerelni a robotra, valamint a kerékvezérlési stratégiákat a különböző mozgástípusok megvalósításához.

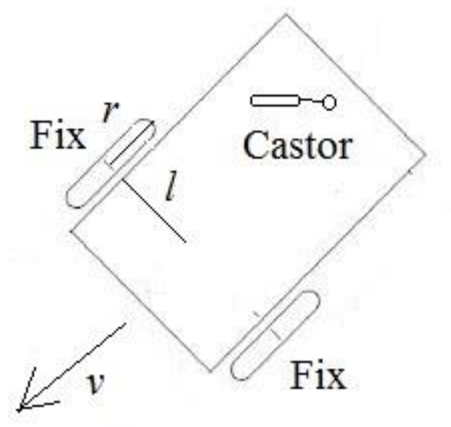


Omnidirekcionális robot irányítási lehetőségei

A robotok felépítése

Kisebb méretű mobilis robotokat általában három kerékkel szerelik fel. Mivel mindig három pontban támaszkodik a földhöz, egyenetlen terepen is biztosítja az állandó robot-föld kontaktust. Nagyobb méretű vagy omnidirekcionális robotoknál alkalmazhatunk négy kereket. Ebben az esetben egyenleten terep esetén az állandó robot talaj kontaktust rugalmas felfüggesztéssel lehet biztosítani.

Az egyik legelterjedtebb kialakítás az alábbi ábrán látható. A robot két, egymástól függetlenül vezérelhető aktív fix kerékkel rendelkezik. Emellett a robotnak van még egy passzív, általában Castor kereke a könnyebb elfordulás megvalósításához. A következőkben ennek a robottípusnak a modellezését és irányítását tárgyaljuk.



X. Ábra: Elterjedt mobilis robot kialakítás

Mobilis robotok kinematikai modellje

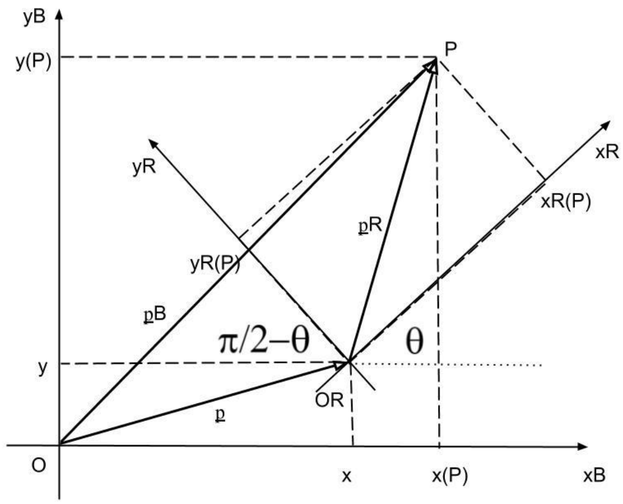
Legyen az X Ábrán látható robot architektúra.
Feltételezzük, hogy a robot merev szerkezetű, a kerekek szintén merevek, a robot síkban mozog és a kerekek nem csúsznak meg.
Legyen az alábbi két koordináta rendszer

$x_B O y_B$ - Bázis koordináta rendszer

$x_R O_R y_R$ - Robothoz rendelt mozgó koordináta rendszer. Az O_R origót a robot keréktengelyének közepén vesszük fel.

A robot síkbeli pozíciója a $x_B O y_B$ koordináta rendszerben: $\underline{x} = \begin{pmatrix} x & y & \theta \end{pmatrix}^T$. A vektorban *x*, *y* az O_R koordinátái az $x_B O y_B$ koordináta rendszerben, θ az Ox_B és $O_R x_R$ tengelyek által bezárt szög.

Koordináta transzformáció síkban: Legyen egy tetszőleges *P* síkbeli pont. Legyenek *P* koordinátái a $x_B O y_B$ koordináta rendszerben ($x^{(P)}, y^{(P)}$) valamint $x_B O y_B$ koordináta rendszerben ($x_R{}^{(P)}, y_R{}^{(P)}$). Keressük meg az összefüggést a ($x^{(P)}, y^{(P)}$) és ($x_R{}^{(P)}, y_R{}^{(P)}$) között.



Koordináta transzformáció síkban

Definiáljuk az alábbi vektorokat:

$$\underline{p} = \begin{pmatrix} x \\ y \end{pmatrix} = x\underline{i} + y\,\underline{j} = x\begin{pmatrix} 1 \\ 0 \end{pmatrix} + y\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\underline{p}_B = \begin{pmatrix} x^{(P)} \\ y^{(P)} \end{pmatrix} = x^{(P)}\underline{i} + y^{(P)}\,\underline{j}$$

$$\underline{p}_R = \begin{pmatrix} {x_R}^{(P)} \\ {y_R}^{(P)} \end{pmatrix} = {x_R}^{(P)}\underline{i}' + {y_R}^{(P)}\,\underline{j}'$$

A vektorokban *i, j* valamint *i',j'* az $x_B O y_B$ valamint az $x_R O_R y_R$ koordináta rendszerekben az egységvektorokat jelölik.
Az *i',j'* robot koordináta rendszer egységvektorai felírva a bázis $x_B O y_B$ koordináta rendszerben:

$$\underline{i}' = \cos(\theta)\underline{i} + \sin(\theta)\,\underline{j}$$

$$\underline{j}' = -\cos(\frac{\pi}{2} + \theta)\underline{i} + \sin(\frac{\pi}{2} + \theta)\,\underline{j} = -\sin(\theta)\underline{i} + \cos(\theta)\,\underline{j}$$

Alkalmazva a vektorösszeadást kapjuk:

$$\underline{p}_B = \underline{p} + \underline{p}_R$$

$$x^{(P)}\underline{i} + y^{(P)}\,\underline{j} = x\underline{i} + y\,\underline{j} + {x_R}^{(P)}\underline{i}' + {y_R}^{(P)}\,\underline{j}'$$

$$x^{(P)}\underline{i} + y^{(P)}\underline{j} = x\underline{i} + y\underline{j} + x_R{}^{(P)}(\cos(\theta)\underline{i} + \sin(\theta)\underline{j}) + y_R{}^{(P)}(-\sin(\theta)\underline{i} + \cos(\theta)\underline{j})$$

$$\begin{cases} x^{(P)} = x + x_R{}^{(P)}\cos(\theta) - y_R{}^{(P)}\sin(\theta) \\ y^{(P)} = y + x_R{}^{(P)}\sin(\theta) + y_R{}^{(P)}\cos(\theta) \end{cases}$$

$$p^{(P)} = \underline{p} + R^T(\theta)\underline{p}_R{}^{(P)}$$

A két koordináta rendszer origója közötti *eltolást* az alábbi vektorral jellemezzük:

$$\underline{p} = \begin{pmatrix} x \\ y \end{pmatrix}$$

A két koordináta rendszer közötti *elforgatást* (a síkra merőleges tengely körül) az alábbi *rotációs mátrix*xal jellemezzük:

$$R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

A mobilis robot sebessége: Legyen a robot haladási sebessége sebessége v és szögsebessége ω.
A robot felépítése miatt nem tud elmozdulni a kerekekre merőleges irányban. Így a tengely-menti sebességek az $x_R O_R y_R$ koordinátar endszerben:

$$v_{Rx} = v$$

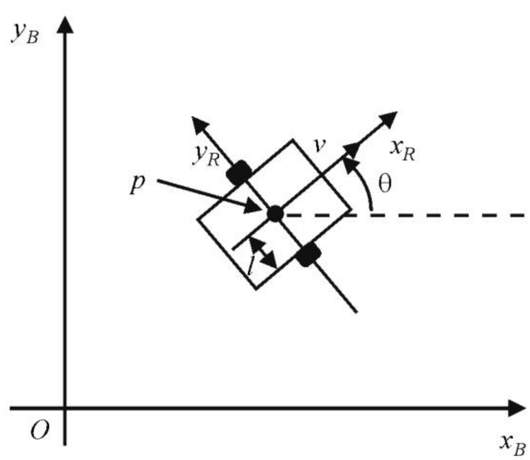
$$v_{Ry} = 0$$

A robot tengely-menti sebessége és szögsebessége az $x_B O y_B$ bázis koordináta rendszerben:

$$v_x = \dot{x}$$

$$v_y = \dot{y}$$

$$\omega = \dot{\theta}$$



A mobilis robothoz rendelt koordináta rendszerek

Az $x_B O y_B$ bázis koordináta rendszerben a tengely-menti sebességeket felírhatjuk a haladási sebesség függvényében:

$$v_x = v\cos\theta$$
$$v_y = v\sin\theta$$

A fenti egyenletekből kapjuk a robot *kinematikai modell*jét, amit irányítás tervezésre alkalmazhatunk.

$$\begin{cases} \dot{x} = \cos\theta \cdot v \\ \dot{y} = \sin\theta \cdot v \\ \dot{\theta} = \omega \end{cases}$$

A modellben kimenetnek a robot koordinátáit és orientációját, bemenetnek a sebességet és szögsebességet tekintjük.

A keréksebességek és a robot sebessége közötti összefüggés

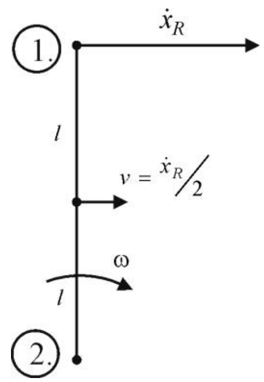
Arra keressük a választ, hogy ha ismerjük a két kerék szögsebességét, mekkora a robot lineáris sebessége illetve a szögsebessége. Legyen adva:

- a kerekek sugara – *r*
- a féltengely távolság – *l*
- a kerekek szögsebessége $\dot{\varphi}_1, \dot{\varphi}_2$

A kerekek haladási sebessége $\dot{x}_{r_1} = r \cdot \dot{\varphi}_1, \dot{x}_{r_2} = r \cdot \dot{\varphi}_2$

Először feltételezzük, hogy a 2. kerék rögzített és az 1. forog (lásd az alábbi ábrát). Ebben az esetben a robot középpontjának sebessége illetve a szögsebessége:

$$\begin{cases} v = \dfrac{\dot{x}_r}{2} = \dfrac{r \cdot \dot{\varphi}}{2} \\ \omega = \dfrac{v}{l} = \dfrac{r \cdot \dot{\varphi}}{2l} \end{cases}$$



Mobilis robot elfordulása egy kerék körül

Hasonlóan megkaphatjuk a sebességet és szögsebességet, ha a 2. kerék mozog és az 1. kerék rögzített. Ha mind a két kerék forog, a sebességek és szögsebességek összeadódnak. A szögsebességnél figyelembe kell venni, hogy a két kerék ellentétes irányba történő forgást generál. Tehát a robot sebessége és szögsebessége a kerék-szögsebességek függvényében:

$$\begin{cases} v = \dfrac{r \cdot \dot{\varphi}_1}{2} + \dfrac{r \cdot \dot{\varphi}_2}{2} \\ \omega = \dfrac{r \cdot \dot{\varphi}_1}{2l} - \dfrac{r \cdot \dot{\varphi}_2}{2l} \end{cases}$$

Mivel az egyenletrendszer lineáris, könnyen megkaphatjuk a kerék-szögsebességeket, ha ismerjük a robot lineáris illetve szögsebességét (*keréksebesség transzformáció*):

$$\begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{pmatrix} = \begin{pmatrix} r/2 & r/2 \\ r/2l & -r/2l \end{pmatrix}^{-1} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

A fenti összefüggés a mobilis robot irányításának megvalósításánál hasznos: az irányítás által előírt robotsebességre és szögsebességre pontosan meghatározhatjuk ezen sebességértékeknek megfelelő kerék-szögesebességeket.

Útvonaltervezés mobilis robotoknak

A kereken guruló mobilis robot útja alatt azon (síkbeli) pontok ($P_i(x_i, y_i)$) rendezett sorozatát értjük, amelyeken a robot át kell haladjon ahhoz, hogy a kiinduló pontból elérjen egy adott célpontba.

Az útvonal-tervezési eljárás feladata a robot útjának meghatározása. Ismertnek tekintjük a célpont (Goal Point) koordinátáit $G(x_G, y_G)$, kezdőpont koordinátáit (Start Point – $S(x_S, y_S)$), valamint annak a terepnek a térképét, amiben a robot mozoghat.

A térkép kell tartalmazza a terep határait (falak) valamint az akadályok síkbeli helyét, kiterjedését, amelyeket a robot el kell kerüljön.

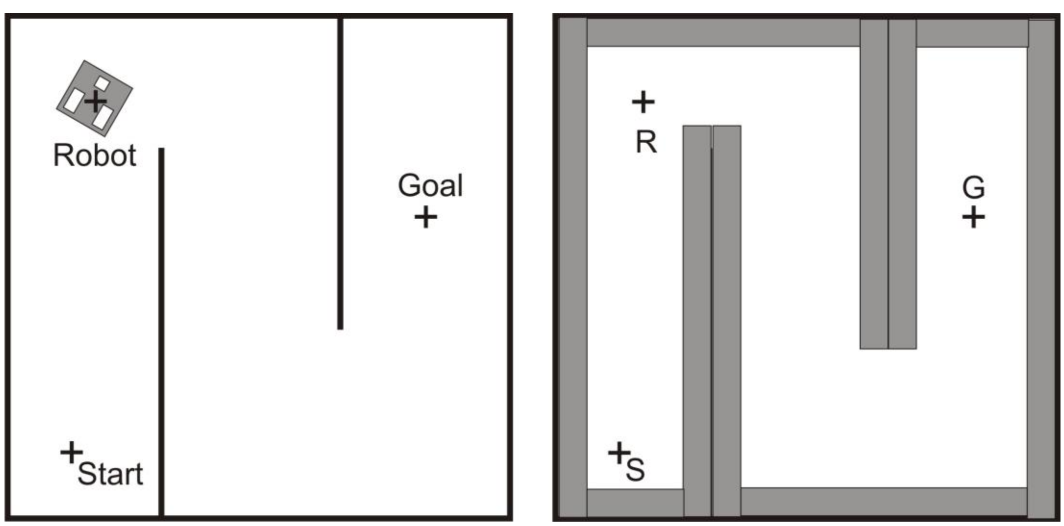
A tért, amiben a robot mozoghat, és amelyet leír a térkép, konfigurációs térnek nevezzük. A konfigurációs teret felosztjuk szabad térre (résztér, amelyben a robot haladhat) valamint az akadályok terére, amely nem tartalmazhatja a robot pályáját.

Ahhoz, hogy a robotnak útvonalat tudjunk tervezni, célszerű a robotot is egy síkbeli pontnak tekinteni ($R(x_R,y_R)$). Feltételezzük, hogy a robot váza befoglalható egy R sugarú körbe. Ahhoz, hogy a robot síkbeli kiterjedését is figyelembe vehessük, a teljes konfigurációs térben terjesszük ki az akadályok terének határait R-el, lásd XX Ábra.

Véletlenszerű pontgeneráláson alapuló útvonaltervezés:

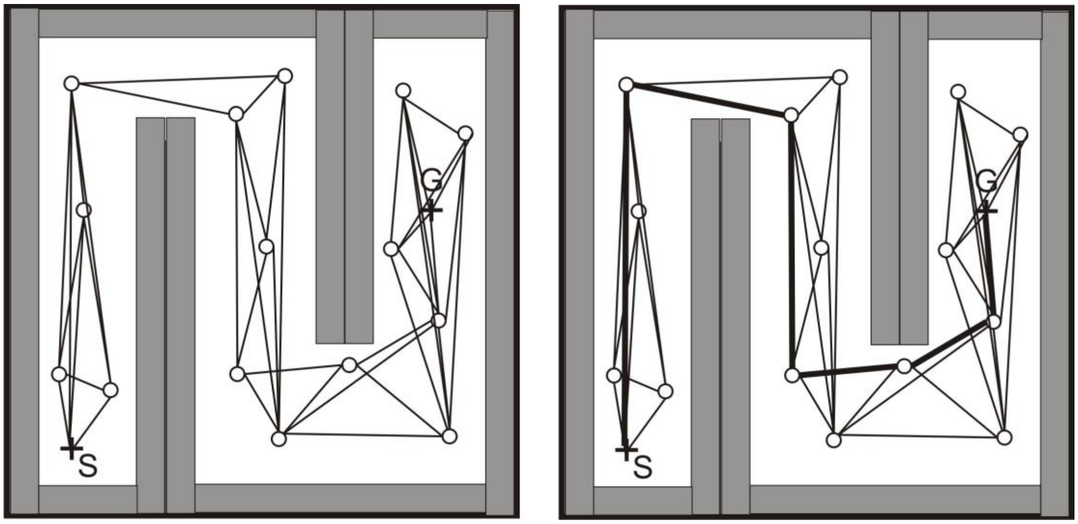
1. Generáljunk a szabad térbe véletlenszerűen N darab síkbeli pontot.
2. Szomszédos pontoknak nevezzük azokat a pontokat, amelyek közé egyenest tudunk húzni úgy, hogy az egyenes ne haladjon át az akadályok terén, illetve ne lépjen ki a konfigurációs térből (a két szomszédos pontot összekötő szakasz maradjon a szabad térben). Kössük össze a szomszédos pontokat. Eredményként egy irányítatlan gráfot kapunk. A gráf éleinek a súlya a szomszédos pontok távolsága.
3. Az S (Start) és G (Goal) pontokat is kössük be a gráfba a velük szomszédos csúcsokhoz.
4. Keressük a minimális utat a gráfban az S és G csúcsok között például a Dijkstra algoritmust alkalmazva.[1]
5. Ha nem létezik ilyen, generáljunk N-nél nagyobb számú pontot és ismételjük meg a 2-3-4 lépéseket.

Az útvonaltervezés eredménye az $S \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots \rightarrow P_i \rightarrow \ldots \rightarrow P_n \rightarrow G$ pontok rendezett sorozata.



Az akadályok terének kiterjesztése

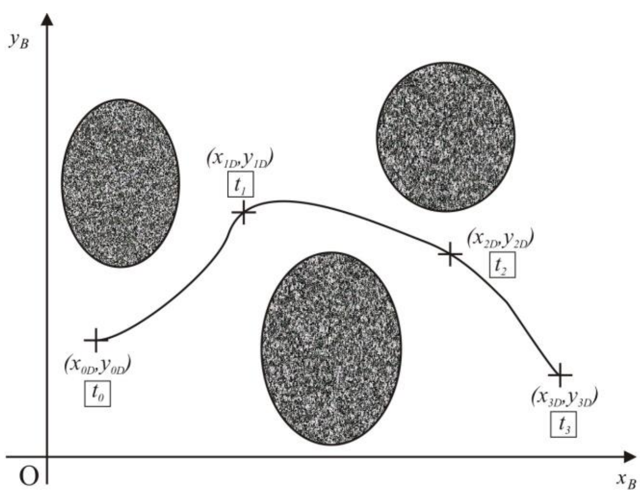
[1] Lásd például: KÁTAI ZOLTÁN, GRÁFELMÉLETI ALGORITMUSOK - http://www.ms.sapientia.ro/~katai_zoltan/documents/Katai_grafok_kinyomtatott.pdf vagy https://hu.wikipedia.org/wiki/Dijkstra-algoritmus



Véletlenszerű pontgeneráláson alapuló útvonaltervezés

Pályatervezés mobilis robotoknak

Ha a robot olyan terepen kell mozogjon, amelyen előre ismert helyzetű akadályok vannak a kiindulópont és a célpont között, a robot előírt mozgását úgy tervezzük meg, hogy az akadályokat kikerülje. Először meghatározunk egy pontsorozatot, amelyen a robot végig kell haladjon, a robot kiinduló pozíciója, célpozíciója és az akadályok helyzetének ismeretében. Legyen ez a pontsorozat: (x_{D0}, y_{D0}), …., (x_{Di}, y_{Di}), … (x_{Dn}, y_{Dn}). Ugyanakkor adottak az időpillanatok, amikor a robot az adott pontban kell legyen: t_0, t_1, …, t_n. Erre a pontsorozatra fektetünk egy pályát fektetni, amelyen a robot végig kell haladjon. Az egyenletes robotmozgás érdekében a pálya törésmentes kell legyen.



Robot előírt mozgása akadályok jelenlétében

A pálya tervezésénél figyelembe kell venni a robot maximális sebességét (v_{MAX}) is. Az alábbi feltételnek teljesülnie kell:

$$v_{MAX} > \frac{\sqrt{(x_{Di} - x_{Di-1})^2 + (y_{Di} - y_{Di-1})^2}}{t_i - t_{i-1}}$$

Válasszuk a pályapontokat összekötő időfüggvényeket (pályaszakaszokat) harmadfokú polinomoknak, tehát a pálya ($P_{0,n}$) harmadfokú polinomok sorozatából fog állni mind az x mind az y tengely mentén:

$$P_{0,n} = P_{0,1} \cup P_{1,2} \cup \dots P_{n-1,n}$$

$$\begin{cases} P_{xi-1,i}(t) = a_{xi}t^3 + b_{xi}t^2 + c_{xi}t + d_{xi} \\ P_{yi-1,i}(t) = a_{yi}t^3 + b_{yi}t^2 + c_{yi}t + d_{yi} \end{cases}, t \in [t_{i-1}, t_i]$$

A pályatervezés során keressük az a_{xi}, b_{xi}, c_{xi}, d_{xi} valamint a_{yi}, b_{yi}, c_{yi}, d_{yi} paramétereket.

Határozzuk meg az x tengely mentén a pályaszakaszokat az alábbi elvek szerint:

I. A pályaszakasz áthaladjon az előírt referenciapontokon:

$$P_{xi-1,i}(t_i) = x_{D_i}$$
$$P_{xi-1,i}(t_{i-1}) = x_{Di-1}$$

II. Az átmenetek a pályaszakaszok között törésmentesek legyenek, vagyis a pályaszakaszok találkozásánál az összefutó pályaszakaszok első- és másodrendű deriváltjai megegyeznek:

$$\dot{P}_{xi-1,i}(t_i) = \dot{P}_{xi,i+1}(t_i)$$
$$\ddot{P}_{xi-1,i}(t_i) = \ddot{P}_{xi,i+1}(t_i)$$

III. Az első és utolsó pályaszakasz teljesítse az előírt kezdő- és végsebességeket a kiindulási, illetve a célpontban.

$$\dot{P}_{x0,1}(t_0) = v_{xD0}$$
$$\dot{P}_{xn-1,n}(t_n) = v_{xDn}$$

Az y tengely mentén ugyanezt a tervezési stratégiát követjük.

Példa: $n=2$ esetén az x tengely mentén összesen 8 paramétert kell meghatározni. Adottak x_{D0}, x_{D1}, x_{D2}, t_0, t_1, t_2, v_{xD0}, v_{xD2}. Ebben az esetben x mentén két pályaszakasz van, tehát a pályát két harmadfokú polinom írja le. Ennek megfelelően a pálya x tengely menti szakaszainak meghatározásához összesen 8 paramétert kell kiszámítsunk, tehát nyolc egyenletre van szükség. A megoldandó egyenletek:

$$
\begin{cases}
P_{x0,1}(t_0) = x_0 & \\
P_{x0,1}(t_1) = x_1 & \\
P_{x1,2}(t_1) = x_1 & \textbf{(I.)} \\
P_{x1,2}(t_1) = x_2 & \\
\dot{P}_{x0,1}(t_1) = \dot{P}_{x1,2}(t_1) & \\
\ddot{P}_{x0,1}(t_1) = \ddot{P}_{x1,2}(t_1) & \textbf{(II.)} \\
\dot{P}_{x0,1}(t_0) = v_{xD0} & \\
\dot{P}_{x1,2}(t_0) = v_{xD2} & \textbf{(III.)}
\end{cases}
$$

Általában is kijelenthető, hogy a felsorolt feltételekkel a pályakövetési feladat megoldható. A kapott egyenletrendszer lineáris, ennek megfelelően a megoldása klasszikus módszerekkel elvégezhető.

Mobilis robotok irányítása

Bevezető

A mobilis robot irányítását a kinematikai modell alapján tervezzük. A kinematikai modell akkor alkalmazható irányítástervezéshez, ha a robot mozgása során jelentős gyorsulás értékekre nem számítunk, a robot dinamikus viselkedése elhanyagolható.

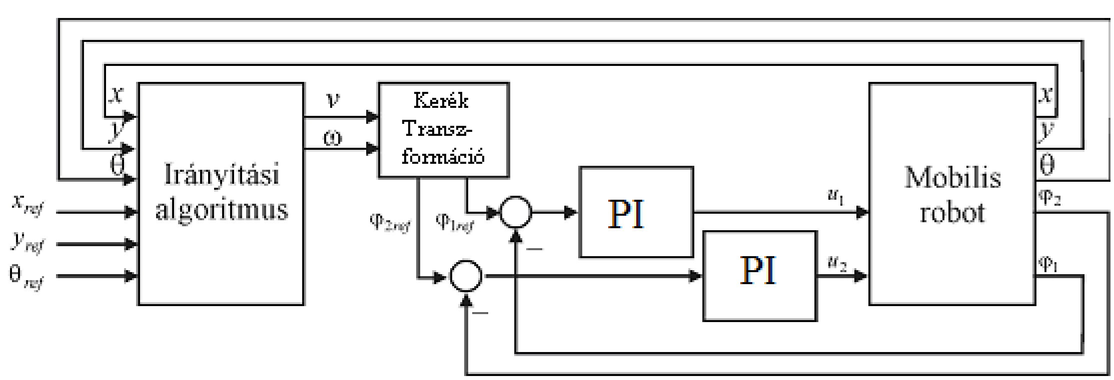
Megkülönböztetünk ponttól-pontig és pályakövető irányítást. Ponttól pontig irányítás esetében adott a síkbeli célpozíció koordinátái (x_{ref}, y_{ref}), illetve a robot kívánt orientációja a célban (θ_{ref}). Pályakövetés esetén adott a robot pályája a kiindulópont és a célpont között valamint az orientáció a pálya mentén. Ezeket folytonos időfüggvényekként adjuk meg:

$$\begin{cases} x_{ref} = x_{ref}(t) \\ y_{ref} = y_{ref}(t) \\ \theta_{ref} = \theta_{ref}(t) \end{cases}$$

A pályakövetést magvalósító irányításhoz ki kell számolni még a pályamerniti sebességeket is: $\dot{x}_{ref}, \dot{y}_{ref}, \dot{\theta}_{ref}$.

Feltételezzük, hogy mérhető a robotsíkbeli pozíciója illetve orientációja a bázis koordináta rendszerben (robot lokalizálás): x,y,θ. A pontos irányítás érdekében célszerű, ha a kerekek szögsebességei ($\dot{\varphi}_1, \dot{\varphi}_2$) is mérhetőek.

Az irányítási algoritmus az előírt és a mért pozíciók és orientációk alapján a robot sebességét és szögsebességét határozza meg (ezek lesznek a beavatkozó jelek). A v és ω alapján meghatározzuk a kerekek elvárt szögsebességét alkalmazva a keréksebesség transzformációt. Az előírt kerék-szögsebesség biztosítására kialakíthatunk egy-egy sebességszabályozó hurkot a két keréknek. A transzformáció által kiszámolt szögsebesség értékek lesznek az alapjelek a sebességszabályozó hurkokban. A sebességszabályozás megvalósításához alkalmazhatunk például PI (Proporcionális-Integráló) típusú szabályozót. Az ábrán u_1, u_2 a kerekeket meghajtó motorok beavatkozó jelét jelöli.



Mobilis robot szabályozási hurok tömbrajza

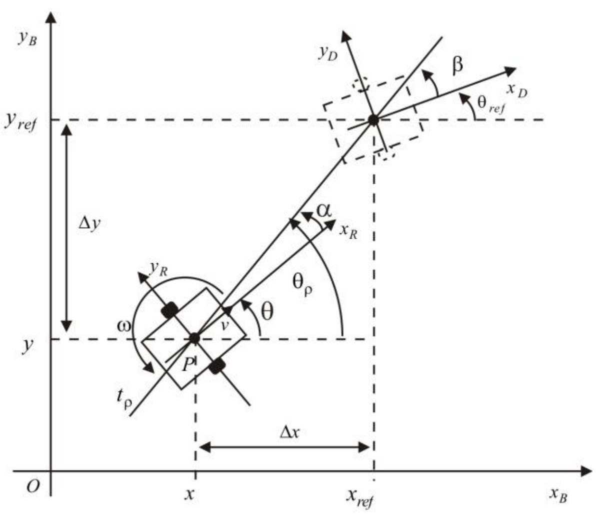
Ponttól-pontig irányítás

Legyen P a robot aktuális pozíciója, P_{ref} pedig a referencia pozíció. Adva van $x_{ref}, y_{ref}, \theta_{ref}$ a mobilis robot előírt pozíciójának koordinátái és orientációja, konstans értékek. Vegyük fel egy ennek megfelelő $x_D O y_D$ koordináta rendszert a mozgás síkjában, amelynek az origója az előírt pozíció és Ox_B és Ox_D által bezárt szög az előírt orientáció. Jelölje t_ρ az egyenest, amely áthalad P_{ref} –en és P-n.

$$\rho = d(P_{ref}, P)$$

$$\alpha = \left(x_R \hat{,} t_\rho\right)$$

$$\beta = \left(x_D \hat{,} t_\rho\right)$$



Ponttól pontig irányítás

Az irányítás a mobilis robotok kinematikai modellje alapján történik, a mért értékek a robot pozíciója és orientációja.

Vizsgáljuk meg a kinematikai modell irányíthatóságát. Linearizáljuk a modellt az $x = y = \theta = 0$ állapotban:

$$\begin{cases} \dot{x} = \cos\theta \cdot v \\ \dot{y} = \sin\theta \cdot v \\ \dot{\theta} = \omega \end{cases} \quad \theta = 0 \Rightarrow \begin{cases} \cos\theta = 1 \\ \sin\theta = 0 \end{cases}$$

Az állapotok $\underline{x} = (x \quad y \quad \theta)^T$. A linearizált modell $(\dot{x} = Ax + Bu)$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{A} \cdot \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}}_{B} \cdot \begin{pmatrix} v \\ \omega \end{pmatrix}$$

A rendszer irányíthatósági mátrixa: $M_C = \left[B \quad AB \quad A^2 B\right]$

$$M_C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad rank(M_C) = 2 \quad < 3$$

Tehát a rendszer lineáris állapot-visszacsatolással nem irányítható.

Nemlineáris állapot-visszacsatolás tervezése: vezessünk be egy új állapotvektort $(\rho\ \alpha\ \beta)$ A kezdőpont-célpont távolságot ρ jelöli. Fogalmazzuk át az irányítási feladatot: keressük úgy a beavatkozó jeleket (v, ω), hogy $\rho \to 0$, $\alpha \to 0$, $\beta \to 0$, ha $t \to \infty$ (lásd a fenti ábrát).

Az új állapotokat megkaphatjuk, a robot kinematikai modell állapotainak és a referenciaértékek függvényében:

$$\begin{cases} \rho = \sqrt{\Delta x^2 + \Delta y^2} \\ \alpha = -\theta + a\tan\left(\dfrac{\Delta y}{\Delta x}\right) \\ \beta = -\theta - \alpha + \theta_{ref} \end{cases} \qquad \Delta x = x_{ref} - x; \quad \Delta y = y_{ref} - y$$

Az új állapotok változásai, ha $\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, vagyis a sebességvektor a „cél felé néz”:

1. $\rho^2 = \Delta x^2 + \Delta y^2 \quad \Big| \dfrac{d}{dt}$

$$2\rho \cdot \dot{\rho} = 2\Delta x(\dot{x}_{ref} - \dot{x}) + 2\Delta y(\dot{y}_{ref} - \dot{y})$$

$$\dot{\rho} = -\frac{\Delta x}{\rho}\dot{x} - \frac{\Delta y}{\rho}\dot{y}$$

$$\frac{\Delta x}{\rho} = \cos(\theta + \alpha)$$
$$\frac{\Delta y}{\rho} = \sin(\theta + \alpha)$$

$$\dot{x} = v \cdot \cos(\theta)$$
$$\dot{y} = v \cdot \sin(\theta)$$

$$\dot{\rho} = -\cos\theta \cdot \cos\alpha \cdot v \cdot \cos\theta + \sin\theta \cdot \sin\alpha \cdot v \cdot \cos\theta - \sin\alpha \cdot \cos\theta \cdot v \cdot \sin\theta - \cos\alpha \cdot \sin\theta \cdot v \cdot \sin\theta$$

$$\dot{\rho} = -(\cos^2\theta + \sin^2\theta) \cdot v \cdot \cos\theta$$

$$\dot{\rho} = -v \cdot \cos\theta$$

2. $\dot{\alpha} = -\dot{\theta} + \dot{\theta}_\rho = -\omega + \dfrac{v}{\rho} \cdot \sin\alpha$

3. $\dot{\beta} = -\dot{\theta} - \dot{\alpha} + \dot{\theta}_{ref} = -\omega + \omega - \frac{v}{\rho} \cdot \sin\alpha = -\frac{v}{\rho} \cdot \sin\alpha$

Tehát az új állapotok változását leíró modell:

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{pmatrix} \cdot \begin{pmatrix} v \\ \omega \end{pmatrix}$$

Ha $\alpha \in \left[-\pi, -\frac{\pi}{2}\right) \cup \left(\frac{\pi}{2}, \pi\right]$, akkor a sebességvektor a céllal ellentétes irányba néz, és a $v \to -v$ helyettesítést kell elvégezzünk a modellben.

Válasszuk az irányítási algoritmust: $\begin{cases} v = K_\rho \cdot \rho \\ \omega = K_\alpha \cdot \alpha + K_\beta \cdot \beta \end{cases}$

Behelyettesítve kapjuk a zárt rendszert: $\begin{cases} \dot{\rho} = -\cos\alpha \cdot K_\rho \cdot \rho \\ \dot{\alpha} = \sin\alpha \cdot K_\rho - K_\alpha \cdot \alpha - K_\beta \cdot \beta \\ \dot{\beta} = -\sin\alpha \cdot K_\rho \end{cases}$

Linearizáljuk a zárt rendszert $\alpha = 0$ körül: $\begin{cases} \cos\alpha = 1 \\ \sin\alpha = \alpha \end{cases}$

A linearizált zárt rendszer: $\begin{cases} \dot{\rho} = -K_\rho \cdot \rho \\ \dot{\alpha} = \left(K_\rho - K_\alpha\right) \cdot \alpha - K_\beta \cdot \beta \\ \dot{\beta} = -K_\rho \cdot \alpha \end{cases}$

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \underbrace{\begin{bmatrix} -K_\rho & 0 & 0 \\ 0 & \left(K_\rho - K_\alpha\right) & -K_\beta \\ 0 & -K_\rho & 0 \end{bmatrix}}_{A} \cdot \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}$$

A rendszermátrix karakterisztikus polinomja:

$$\det(\lambda I - A) = \left(\lambda + K_\rho\right) \cdot \left(\lambda^2 - \left(K_\rho - K_\alpha\right) \cdot \lambda - K_\beta \cdot K_\rho\right)$$

Ahhoz, hogy a szabályozás stabil legyen, az összes sajátérték valós része negatív kell, hogy legyen $Re(\lambda_i) < 0, i = 1,2,3$. Ennek feltétele, hogy a szabályozó paramétereit az alábbi módon válasszuk meg:

$$\begin{cases} K_\rho > 0 \\ K_\beta < 0 \\ K_\rho - K_\alpha < 0 \end{cases}$$

A fenti paraméterválasztással az irányítási algoritmus garantálja, hogy az állapotok *(ρ, α, β)* zéróba konvergálnak, vagyis a robot eléri a referencia pozíciót és a referencia orientációt. Az irányítási stratégia három részmozgást valósít meg: a robot ráfordul a célpozícióra, a robot végighalad kiinduló pozíció és célpozíció között, a robot ráfordul az előírt orientációra.

Nemlineáris ponttól-pontig irányítás

Legyen az alábbi egyszerűsített irányítási feladat: tervezzünk egy olyan irányítási törvényt, amely garantálja, hogy $\rho \to 0, \alpha \to 0$. Az orientációt nem írjuk elő a referencia pozícióban.

A szabályozás tervezéséhez a Lyapunov tételt alkalmazzuk.

A Lyapunov tétel:

Legyen az $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ dinamikus rendszer, ahol $\mathbf{x} \in R^n$.

Vezessük be az egyensúlyi állapot fogalmát. Az $\boldsymbol{x}^*$állapot *a rendszer egyensúlyi állapota*, ha a t^* pillanatban a rendszer állapota $\boldsymbol{x}^*$, akkor bármely $t>t^*$ pillanatban az állapot $\boldsymbol{x}^*$ marad.

Az $\boldsymbol{x}=\boldsymbol{0}$ egyensúlyi állapot Lyapunov értelemben *aszimptotikusan stabil* a $t=t_0$ időpillanatban, ha létezik $r(t_0) >0$ úgy, hogy ha $||\boldsymbol{x}(t_0)||<r$ akkor $||\boldsymbol{x}(t)||\to 0$, ha $t\to\infty$.

Tétel: A $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ rendszer $\boldsymbol{x}=\boldsymbol{0}$ egyensúlyi állapota aszimptotikusan stabil ha létezik egy $V(\mathbf{x})$ Lyapunov energiafüggvény, amelyre igaz, hogy:

- $V(\boldsymbol{x}) > 0$, ha $\boldsymbol{x} \neq \boldsymbol{0}$,
- $V(\boldsymbol{x}) = 0$, ha $\boldsymbol{x} = \boldsymbol{0}$,
- $\dot{V}(\mathbf{x}) < 0$, ha $\boldsymbol{x} \neq \boldsymbol{0}$.

Az előző alfejezetben bemutatott modellezés alapján ρ ás α dinamikáját az alábbi egyenletek írják le:

$$\begin{cases} \dot{\rho} = -\rho \cos\alpha \, v \\ \dot{\alpha} = \dfrac{\sin\alpha}{\rho} v - \omega \end{cases}$$

Válasszuk az alábbi Lyapunov függvényt:

$$V = \frac{\rho^2}{2} + \frac{\alpha^2}{2}$$

A Lyapunov függvény deriváltja:

$$\dot{V} = \rho\dot{\rho} + \alpha\dot{\alpha} = -\rho\cos\alpha\, v - \alpha\left(-\frac{\sin\alpha}{\rho} v + \omega\right)$$

Annak garantálására, hogy a Lyapunov függvény deriváltja negatív legyen, válasszuk az irányítást:

$$\begin{cases} v = K_v \rho \cos\alpha \\ \omega = K_\omega \alpha + K_v \cos\alpha \sin\alpha \end{cases} \qquad K_v, K_\omega > 0.$$

Ezzel az irányítással a Lyapunov függvény derváltja negatív:

$$\dot{V} = -K_v \rho^2 \cos\alpha^2 - K_\omega \alpha^2 \leq 0$$

Tehát a nemlineáris irányítási törvények garantálja, hogy az irányított rendszer stabil, az irányított mobilis robot eljut a referencia pozícióba.

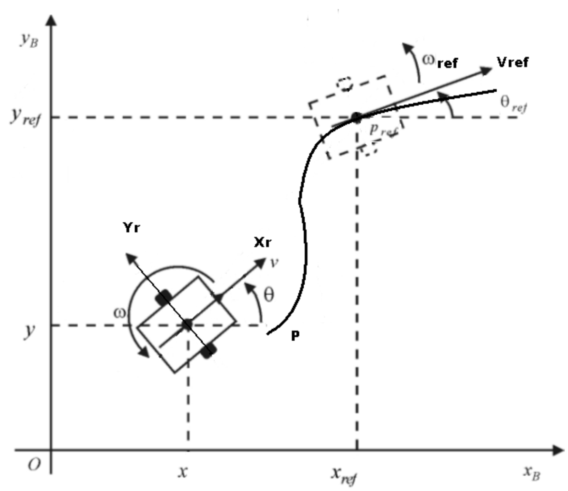
Pályakövetést megvalósító irányítás

Ebben az esetben a robot előírt pozíciója és orientációja az idő függvénye. Legyen $P_x=P_x(t)$ és $P_y=P_y(t)$ az előírt pálya. Az előírt pálya mentén a referencia sebességek a bázis koordináta rendszerben:

$$\begin{cases} v_{x_r} = \dot{P}_x(t) \\ v_{y_r} = \dot{P}_y(t) \end{cases} \qquad v_r = \sqrt{v_{xr}^2 + v_{yr}^2}\,, \theta_r = a\tan\frac{v_{y_r}}{v_{x_r}} \qquad \omega_r = \dot{\theta}_r$$

A pályakövetést megvalósító irányítás tervezéséhez először definiálnunk kell egy referencia robotot, amely az előírt pályán mozog. Ezután képezünk egy hibarendszert a referencia és a valós robot között. Az irányítást úgy keressük, hogy a hibarendszert stabilizáljuk. A referenciarobot:

$$\begin{cases} \dot{x}_r = \cos\theta_r \cdot v_r \\ \dot{y}_r = \sin\theta_r \cdot v_r \\ \dot{\theta}_r = \omega_r \end{cases}$$



$$\underline{\tilde{X}} = \begin{pmatrix} X - X_r \\ Y - Y_r \\ \theta - \theta_r \end{pmatrix}$$

Referencia robot alapú irányítás

Vezessünk be egy hibatranszformációt úgy, hogy az X_BOY_B koordináta rendszerben definiált hibát forgassuk be az x_ROy_R koordináta rendszerbe:

$$e = Rot(z,\theta)\cdot \underline{\tilde{X}} \ \ , \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{\theta} \end{pmatrix}$$

Kihasználjuk, hogy a rotációs mátrix az alábbi tulajdonságokkal rendelkezik:

I. $Rot^{-1}(z,\theta) = Rot^{T}(z,\theta)$

II. $\dot{R}ot(z,\theta)\cdot Rot^{-1}(z,\theta) = \begin{pmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \omega = \dot{\theta}$

Az e hibavektor dinamikája:

$$\dot{e} = \left(Rot(z,\theta)\cdot \tilde{X}\right)' = \dot{R}ot(z,\theta)\cdot \tilde{X} + Rot(z,\theta)\cdot \dot{\tilde{X}} =$$
$$= \dot{R}ot(z,\theta)\cdot Rot^{-1}(z,\theta)\cdot e + Rot(z,\theta)\cdot \dot{\tilde{X}}$$

$$\begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{pmatrix} = \begin{pmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} + \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v\cdot\cos\theta - v_r\cos\theta_r \\ v\cdot\sin\theta - v_r\sin\theta_r \\ \omega - \omega_r \end{pmatrix}$$

$$\begin{pmatrix}\dot e_1\\ \dot e_2\\ \dot e_3\end{pmatrix}=\begin{pmatrix}0 & \omega & 0\\ -\omega & 0 & 0\\ 0 & 0 & 0\end{pmatrix}\cdot\begin{pmatrix}e_1\\ e_2\\ e_3\end{pmatrix}+\begin{pmatrix}v-v_r\cos e_3\\ v_r\sin e_3\\ \omega-\omega_r\end{pmatrix}$$

Vezessük be a következő beavatkozó jel transzformációt:

$$\begin{cases}u_1=v-v_r\cdot\cos e_3\\ u_2=\omega-\omega_r\end{cases}$$

Ezekkel a jelölésekkel a hibarendszer modellje:

$$\begin{pmatrix}\dot e_1\\ \dot e_2\\ \dot e_3\end{pmatrix}=\begin{pmatrix}0 & \omega & 0\\ -\omega & 0 & 0\\ 0 & 0 & 0\end{pmatrix}\cdot\begin{pmatrix}e_1\\ e_2\\ e_3\end{pmatrix}+\begin{pmatrix}0\\ v_r\cdot\sin e_3\\ 0\end{pmatrix}+\begin{pmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{pmatrix}\cdot\begin{pmatrix}u_1\\ u_2\end{pmatrix}$$

Linearizáljuk a hibarendszert az $e_1 = e_2 = e_3 = 0$ pont körül.

$$\begin{matrix}\sin e_3\cong e_3,\\ e_3\cong 0\quad ,\\ \omega\cong\omega_r\end{matrix}\qquad \begin{pmatrix}\dot e_1\\ \dot e_2\\ \dot e_3\end{pmatrix}=\underbrace{\begin{pmatrix}0 & \omega & 0\\ -\omega & 0 & 0\\ 0 & 0 & 0\end{pmatrix}}_{A}\cdot\begin{pmatrix}e_1\\ e_2\\ e_3\end{pmatrix}+\underbrace{\begin{pmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{pmatrix}}_{B}\cdot\begin{pmatrix}u_1\\ u_2\end{pmatrix}$$

A rendszer irányíthatósági mátrixa:

$$M_C=\begin{bmatrix}B & AB & AB^2\end{bmatrix}=\begin{pmatrix}1 & 0 & 0 & 0 & -\omega_r^2 & \omega_r\cdot v_r\\ 0 & 0 & -\omega_r & v_r & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}$$

Annak a feltétele, hogy a rendszer irányítható legyen: $rank(M_C)=3 \leftrightarrow v_r \neq 0\ \omega_r \neq 0$, vagyis az irányítás minden pillanatában a referencia robot kell mozogjon.

Válasszuk a beavatkozó jeleket az alábbi formában:

$$\begin{cases}u_1=-k_1\cdot e_1\\ u_2=-k_4\cdot v_r\cdot\dfrac{\sin e_3}{e_3}\cdot e_2-k_3\cdot e_3\end{cases}$$

ahol k_1, k_3, $k_4 > 0$.

Megjegyzés: u_1, u_2 alapján a robot sebessége és szögsebessége visszaszámolható:

$$\begin{cases}v=u_1+v_r\cdot\cos e_3\\ \omega=u_2+\omega_r\end{cases}$$

Az irányítási algoritmus analíziséhez a Lyapunov tételt alkalmazzuk. Rendeljük az irányított rendszerhez (hibarendszer az irányítási algoritmussal) az alábbi Lyapunov függvényt:

$$V = \frac{k_4}{2} \cdot \left(e_1^2 + e_2^2\right) + \frac{1}{2} e_3^2 \ ,$$

Igazoljuk, hogy $\dot{V} < 0$. Felhasználva a dinamikus modellt:

$$\begin{aligned}
\dot{V} &= \frac{k_4}{2} \cdot \left(2 \cdot e_1 \cdot \dot{e}_1 + 2 \cdot e_2 \cdot \dot{e}_2\right) + \frac{1}{2} e_3 \cdot \dot{e}_3 \\
&= k_4 \cdot e_1 \cdot (\omega \cdot e_2 + u_1) + k_4 \cdot e_2 \cdot (-\omega \cdot e_1 + v_r \cdot \sin e_3) + e_3 \cdot u_2 \\
&= -k_4 \cdot k_1 \cdot e_1^2 - k_4 \cdot e_3^2 < 0
\end{aligned}$$

Tehát u_1, u_2 biztosítja, hogy a Lyapunov függvény zéróba konvergáljon, vagyis a hibarendszer stabilizálását és implicit a referencia robot követését, ha $v_r \neq 0$ és $\omega_r \neq 0$.