



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Irányítástechnika és Informatika Tanszék

Gyenes Zoltán Bálint

Közlekedési szabályok integrálása mobilis robotok mozgástervező algoritmusába

KONZULENS

Gincsiné Dr. Szádeczky-Kardoss Emese

BUDAPEST, 2018

Tartalomjegyzék

Kivonat.....	4
Abstract.....	5
1 Bevezetés	6
2 A Velocity Obstacles módszer bemutatása	9
2.1 CC és VO meghatározása	9
2.2 Elérhető elkerülő sebességek térrésze.....	12
2.3 Velocity Obstacles módszer tulajdonságai	14
2.3.1 I. tétel	14
2.3.2 II. tétel.....	15
2.4 Sebességvektor választása a robot számára	15
2.4.1 Heurisztikus mozgástervezés	16
2.4.2 A Safety Velocity Obstacles módszer	17
3 A mozgástervező algoritmus	19
4 Ütközést okozó robotsebességek meghatározása	21
5 Mozgástervezés hajók számára szabálykészlet alapján	23
5.1 Szabályok alapján elkülöníthető szituációk	23
5.2 Előszűrés a szabályok alkalmazására.....	25
6 Traffic Regulation Velocity Obstacles módszer	26
6.1 Szabályok alapján történő tervezés	28
6.1.1 Előzés.....	29
6.1.2 Szemből érkezik az akadály.....	30
6.1.3 Jobbról érkezik az akadály.....	31
6.1.4 Balról érkezik az akadály.....	33
6.2 Megkülönböztető jelzést használó jármű esetén történő pályatervezés.....	36
6.2.1 A robot megkülönböztető jelzéssel rendelkezik:	36
6.2.2 Megkülönböztető jelzéssel rendelkező akadály.....	38
6.3 Több akadály esetén történő szabály alapú pályatervezés	39
6.4 SVO és TRVO módszer párhuzamos használata a mozgástervezés során	45
7 Elérhető elkerülő sebesség térrészek számossága, számítási igény	48
7.1 1 akadály esetén az elérhető elkerülő sebesség területek felosztása.....	49
7.2 2 akadály esetén az elérhető elkerülő sebesség területek felosztása.....	51

7.3 Futásidő vizsgálata.....	55
8 Értékelés, továbbfejlesztési lehetőségek.....	58
Köszönetnyilvánítás	59
Irodalomjegyzék.....	60
Függelék.....	62

Kivonat

A mobilis robotok legfontosabb feladata minden esetben, hogy egy kiindulási helyről a célpozícióba eljussanak, ütközésmentes mozgást biztosítva a robot és környezete között. Dinamikus környezet esetén nemcsak statikus, hanem mozgó akadályok is jelen vannak a robot munkaterében. Ezen mozgás tervezésére szolgálnak a mozgástervező algoritmusok.

TDK dolgozatom célja egy olyan mozgástervező algoritmus kifejlesztése és implementálása, mely egyaránt biztosítja a robot munkaterében jelenlevő statikus és mozgó akadályok elkerülését. Ehhez a *Velocity Obstacles* módszert használok fel, amely egyidőben tervezi meg a mobilis robot sebesség- és pályaprofilját.

Munkám során a leggyorsabb és a legbiztonságosabb pályatervezési megoldások mellett kidolgozok egy új módszert, mely a mozgástervezést a közlekedési szabályok (KRESZ szabályok) felhasználásával valósítja meg, azaz pl.: ha jobbról érkezik egy akadály, akkor a robotnak elsőbbséget kell adnia. Az algoritmus megvalósításához minden egyes mintavételi időpillanatban meg kell határozni a robot által elérhető sebességvektorok halmazát, melyek a bevezetett szabálykészletnek megfelelnek. Ezen sebesség térrészek metszetéből választhatunk sebességvektort a robot számára. Az új módszert *Traffic Regulation Velocity Obstacles (TRVO)* módszernek neveztem el. A módszer a robot és az akadályok között elkerülő manővereket tervez, figyelembe véve a robot és az akadályok – ismertnek feltételezett – aktuális pozícióját és sebességét. A fejlesztés során külön megvizsgálásra kerül, hogy az akadályok számának növekedése miként befolyásolja a számítási igényt.

A mozgástervezés eredményét minden esetben szimulációban vizsgálom, a robot és az akadályok mozgásából videót is készítek.

Ez a mozgástervező algoritmus alkalmazható az egyre nagyobb teret hódító autonóm járművek mozgásának tervezéséhez is, ezáltal a jövőben nagyon sok lehetőség nyílna a továbbfejlesztésre.

Abstract

The main task of a mobile robot is to reach the goal from a start position ensuring a collision-free motion between the robot and its environment. In dynamic environment there are not only static but also moving obstacles in the workspace of the robot. We can use the motion planning algorithms to plan this movement.

The main purpose of my work is to develop and implement a motion planning algorithm that can ensure the avoidance of static and moving obstacles in the workspace of the robot. I used the *Velocity Obstacles* method, which is suitable for planning the velocity- and trajectory of the robot at the same time.

In my work next to the fastest and safest motion planning solutions I develop a new method, that executes the motion planning of the robot by using the traffic regulations, so e.g.: if an obstacle comes from right, then the robot has to give way to it. The algorithm calculates the set of velocities of the robot in every sampling time that are sufficient for the previous regulations. We can choose a velocity vector for the robot from the intersection of the sets of robot's velocities. I named this method *Traffic Regulation Velocity Obstacles (TRVO)* method. With this method, I plan evasive maneuvers by using information about actual velocity and actual position of the robot and obstacles which information are supposed to be known. During the development it is also a research topic, how the increasing number of the obstacles can influence the computational demand.

I always test the result of the robot's motion planning by simulations and I make videos of the motion of the robot and the obstacles.

This motion planning algorithm is suitable for motion planning of autonomous vehicles, that conquering more and more space. So, in the future there are a lot of opportunities for further development.

1 Bevezetés

Korunk egyik legjobban fejlődő iparága a robotika. Gyárak ezrei automatizálják gyártósoraikat robotok, robotkarok felhasználásával, ezzel is növelve a termelés hatékonyságát. Robotok felhasználásával megkönnyíthetjük mind a mérnöki, mind a mindennapi élet nehézségeit.

A robotoknak mindig van egy célfeladatuk, melynek megvalósítására hivatottak. Az autonóm robotok külső beavatkozás nélkül képesek elvégezni feladatukat, szenzoraikkal érzékelni tudják a környezet változásait, és ezekre a változásokra reagálni képesek.

Sok esetben ez a célfeladat a robot mozgása egy kezdő pozícióból egy célpozícióba (pl. gyárakban anyagok, vagy elkészült termékek szállítása egyik helyről a másikra). A robot munkaterében különböző akadályok fordulhatnak elő. Dinamikus környezetről akkor beszélhetünk, ha a térben nemcsak statikus, hanem mozgó akadályok is jelen vannak. A robotnak minden esetben úgy kell a célpozícióba eljutnia, hogy mozgása során egyetlen akadállyal se ütközzön. Ez az elsődleges célja a mozgásának. Természetesen az is fontos, milyen gyorsan tud eljutni a robot a kívánt pozícióba, illetve, hogy mekkora utat kell megtennie ehhez.

A mozgástervező algoritmusok segítségével meghatározhatjuk, hogy a robotok milyen pálya mentén és milyen sebességgel tudják elérni a megadott célt.

A robotok pályatervezésére nagyon sok kísérlet volt már [1], különösképpen az elmúlt két évtizedben. Adottak a robot dinamikai tulajdonságai, a környezet tulajdonságai, a robot kezdő helyzete és a célpozíció helyzete. A mozgástervezés legfőbb feladata tehát eljuttatni a robotot kezdő helyzetéből a kívánt pozícióba úgy, hogy közben engedelmeskedjünk a környezet szabályainak (ne ütközzünk az akadályokkal). Egy algoritmusnak, amely megoldja ezt a feladatot, véges idő alatt le kell futnia, és megvalósítható megoldással kell visszatérnie, amennyiben létezik, ellenkező esetben jelezni kell, hogy a mozgástervezés valamilyen oknál fogva nem megvalósítható. A mozgástervezés dinamikus környezetben nehéz problémának számít, mivel egy pályatervezési és sebességprofil tervezési feladatot is végre kell hajtani. Az alap verziójú pályatervezési problémát „Generalised piano movers” problémának nevezzük, mely NP-nehéz probléma [2]. Habár létezik a teljes tervezési algoritmus ennek a

mozgástervezésnek, mégis olyan bonyolult, hogy nem használható gyakorlati alkalmazásokban.

A mozgástervezés dinamikus környezetben egyik megoldásként a robot konfigurációs teréhez egy idődimenziót adtak hozzá, feltételezve az akadályok pályáját és sebességét. Korábbi munkákban poligon robotok és poligon mozgó akadályok esetére oldották meg a mozgástervezési problémát egy láthatósági gráf keresésével a konfiguráció-idő térben [3], [4].

Másik megközelítés alapján a mozgástervezés dinamikus környezetben felbontható kisebb alproblémákra: pályatervezés és sebesség tervezés. Ez a módszer először meghatározza az elérhető pályát az álló akadályok figyelembevételével, ezután pedig a pálya menti sebességet úgy, hogy a robot elkerülje a mozgó akadályokat [5], [6].

Újabb pályatervezési megoldás a széleskörben használt Rapidly exploring Random Tree (RRT) módszer [7], [8]. Ezt a módszert kiterjesztették dinamikus környezetre is, ahol a mozgó akadályok pályája ismeretlen [9].

Egy másik megoldási lehetőség, a lokális és visszaható navigáció abban különbözik az általános globális pályatervező megközelítésektől, mint pl. a [10]-[12], hogy ahelyett, hogy a teljes pályát megterveznék a célig, a robot kizárólagosan a közvetlen környezetével van kölcsönhatásban bármely időpillanatban. Néhány jól ismert visszaható navigációs módszer:

- dinamikus ablak módszer [13], [14].
- elkerülhetetlen ütközési állapotok módszere [15]-[17].
- potenciálmező módszer [18], [19].
- elérhető rés navigáció [20].
- sebesség akadályok módszere (*Velocity Obstacles*) [21].

TDK dolgozatom során egy elsőrendű módszert mutatok be, mellyel meghatározom az időben változó környezetben mozgó robotom pályáját, felhasználva a sebesség információkat (mind az akadályok, mind a robotom sebességét), hogy meghatározzam az esetleges ütközéseket. Ez a módszer felhasználja a *Velocity Obstacles* (sebesség akadály) elgondolást [21], amely feltérképezi a változó környezetet és a robot sebesség terét, így határozza meg a pálya geometriáját a sebességprofilal együtt. A *Velocity Obstacles* egy elsőrendű közelítése a robotsebességek halmazának, amelyek egy

jövőbeli időpillanatban ütközéshez vezetnének valamely akadállyal. Az elkerülő manővereket úgy tudjuk meghatározni, hogy ezen *Velocity Obstacles* térrészen kívülről választunk sebességet a robotnak. Az elkerülő manővernek minden esetben biztosítania kell, hogy dinamikailag elérhető legyen, teljesítve az összes fizikai kényszert. A gyorsaság mellett a biztonság is fontos szerepet játszhat a pályatervezés folyamán, ennek érdekében került kifejlesztésre a *Safety Velocity Obstacles (SVO)* módszer [22].

A *Velocity Obstacles* módszert alkalmazták korábban hajók pályájának tervezésére, felhasználva a hajózási szabályokat [23]. TDK dolgozatomban kifejlesztettem a *Traffic Regulation Velocity Obstacles (TRVO)* módszert, mely KRESZ szabályok felhasználásával tervezi meg a robot sebesség- és pályaprofilját.

A dolgozat felépítése: a 2. fejezetben *Velocity Obstacles* módszer, tulajdonságai kerülnek bemutatásra. Ezt követően a 3. fejezetben bemutatom a mozgástervező algoritmus lépéseit. A 4. fejezetben az ütközést okozó robotsebességek számítását szemléltetem. Majd az 5. fejezetben megvizsgálom, hogy hajók esetén milyen szabálykészletet alkalmaztak korábban a mozgástervezésre, milyen ötleteket lehet átvenni a mobilis robotok és járművek szabályalapú mozgástervezésére. A 6. fejezetben bevezetem és leírom a *TRVO* módszer működését. Ezt követően a 7. fejezetben az elérhető elkerülő sebesség térrészek számának vizsgálata kerül bemutatásra a futási idővel kiegészítve. Legvégül a 8. fejezetben értékelem a kapott eredményeket és a továbbfejlesztési lehetőségeket mutatom be.

2 A Velocity Obstacles módszer bemutatása

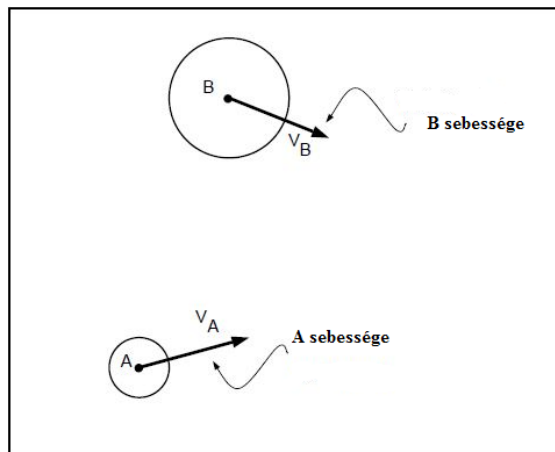
Ebben a fejezetben bemutatom a *Velocity Obstacles*, azaz a sebesség akadály módszert, melynek megismerésének forrásaként a [21] és a [24] szakirodalmat használtam. A módszer több akadály esetén is érvényes, figyelembe véve, hogy statikus vagy mozgó akadályról van-e szó.

A vizsgálatot kör alakú robotra és akadályokra korlátozzuk. Ám ez nem jelent nagyon szigorú korlátozást, amióta felfedezték, hogy általános sokszögeket minden esetben helyettesíteni tudunk több körrel [21].

Az akadályok pályáját úgy közelítjük, hogy egy időhorizonton belül állandó sebességvektort feltételezünk. Amennyiben ez nem elég pontos, akkor az *NLVO* használatával [25], tetszőleges akadálypályák is kezelhetővé válnak.

2.1 CC és VO meghatározása

Tegyük fel, hogy van 2 kör alakú objektum A és B , melyeket a következő ábrán láthatunk:



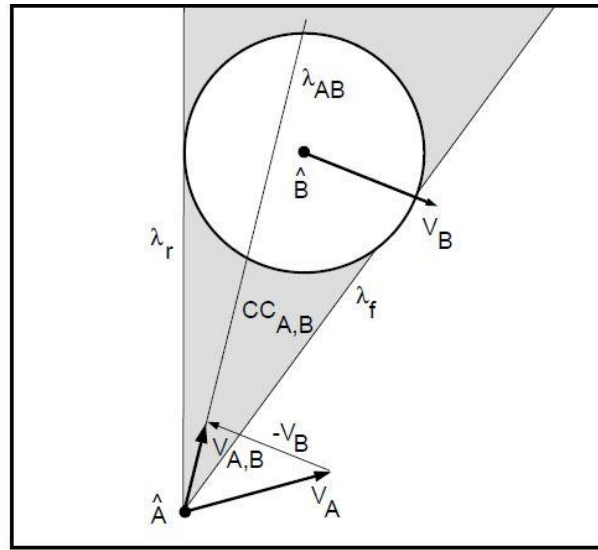
2-1. ábra: Robot és akadály helyzete, a hozzájuk tartozó sebességekkel [21]

A jelölje a robotot, B pedig az akadályt, a robotnak a sebességvektora egy adott időpillanatban v_A , az akadály sebességvektora ugyanebben az időpillanatban v_B . Minden egyes akadálynál, a robotnál is a sebességvektorukat a középpontból húzzuk.

Ezután változtassuk meg a konfigurációs terünket, csökkentsük a robotot egy pontra, az akadály méretét pedig növeljük a robot sugarának nagyságával. A robotot

jelölje $\hat{A}$, az akadályt $\hat{B}$. Az A robot és a B akadály mozgásuk során akkor és csak akkor fognak ütközni, ha $\hat{A}$ és $\hat{B}$ ütköznek.

A Collision Cone-t, azaz CC-t, melyet ütközési kúpnak fordíthatunk, úgy definiáljuk, hogy azon relatív sebességek halmaza a robot és az akadály között, amelyek ütközést okoznak. (Ez a tartomány csak térben alkot egy kúpot, síkban nem, de a megoldás térbeli kiterjeszthetősége miatt a szakirodalomban a cone elnevezés terjedt el, ezért a magyar fordításban is a kúpot alkalmazom.)

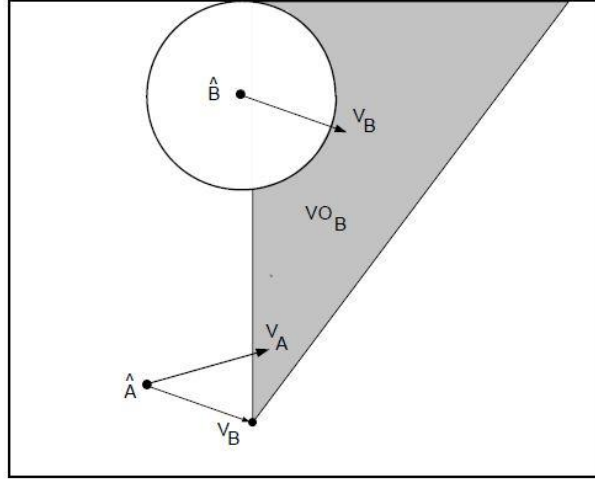


2-2. ábra: Ütközési kúp [21]

A 2-2. ábra mutatja a robotból húzott két érintő félegyenest az akadály felé (λ_r , λ_f). Az ütközési kúp csúcsa a robot pozíciója. Az ütközési kúp tehát nem más, mint a csúcs és a két érintő félegyenes közötti térrész. A képen látható a robot és az akadály sebességvektora (v_A , v_B), illetve a relatív sebesség ($v_{A,B} = v_A - v_B$), illetve $\lambda_{A,B}$ jelöli a relatív sebesség félegyenesét. Abban az esetben, ha ez a félegyenes, azaz a relatív sebesség vektor a két érintő között helyezkedik el, akkor ütközés fog történni a robot és az akadály között. Ha pedig a relatív sebesség ezen ütközési kúpon kívül esik, akkor ütközésmentes mozgást valósítunk meg, feltételezve, hogy az akadály ugyanilyen sebességvektorral rendelkezik a jövőben is. (Amennyiben λ_r mellől választ a robot sebességvektort, akkor az akadály mögött fog elmenni, λ_f mellől sebességvektort választva az akadály előtt fog elhaladni a robot (ha a sebességvektor a CC-n kívül esik)). Tehát az ütközési kúp a következőképpen írható le képlettel:

$$CC_{A,B} = \{v_{A,B} | \lambda_{A,B} \cap \hat{B} \neq \emptyset\} \quad (2.1)$$

Minden egyes akadályhoz tartozik egy ilyen ütközési zóna. Ha több akadály van jelen a térben, akkor előnyös, ha a robotnak mindig csak az abszolút sebességével kell számolnunk, nem a relatív sebességekkel. Ezt úgy érhetjük el, hogy az adott ütközési kúphoz hozzáadjuk az adott akadály sebességét (eltoljuk az ütközési kúpot), így kapjuk meg a *Velocity Obstacles*-t, azaz *VO*-t, melyet sebesség akadállynak fordíthatunk.



2-3. ábra VO térrész [21]

Tehát a *VO*-t a következőképpen definiálhatjuk:

$$VO_{A,B} = CC_{A,B} \oplus v_B \quad (2.2)$$

Az összeadás alatt a Minkowsky összeadást értjük [21].

Ha a robotnak a *VO*-n kívülről választunk robotsebességet, akkor megakadályozzuk az ütközést:

$$A(t) \cap B(t) = \emptyset \text{ ha } v_A(t) \notin VO(t) \quad (2.3)$$

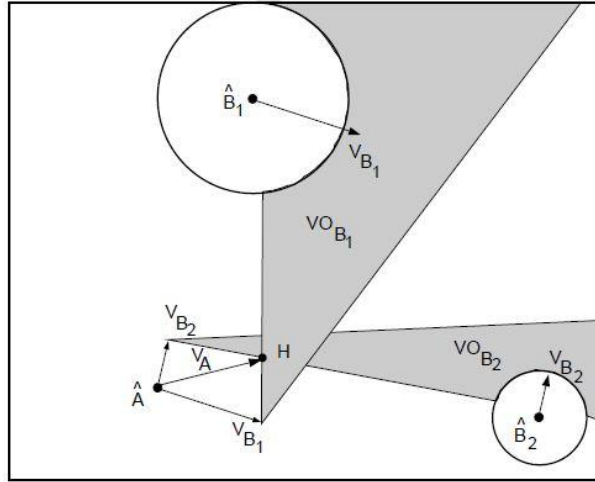
Abban az esetben, ha álló akadályt vizsgálunk, akkor a *VO* megegyezik a *CC* –vel, mivel ekkor az akadály sebességvektorának hossza 0, így az ütközési kúp eltolása nem eredményez változást.

Ha *VO* határáról választunk sebességet, akkor érinteni fogja a robot mozgása során az akadályt, de nem okoz ütközést. A gyakorlatban úgy szoktunk ilyen esetekben tervezni, hogy a mozgástervező algoritmus elején az akadályok sugarának nagyságát a biztonsági távolsággal megnöveljük. Így az algoritmus végrehajtása során, ha érintés történne, akkor sem fog a valóságban összeérni a robot és az akadály.

Több akadály esetén a VO nem más, mint az egyes akadályokhoz tartozó VO -k uniója.

$$VO = \cup_{i=1}^n VO_{Bi} \quad (2.4)$$

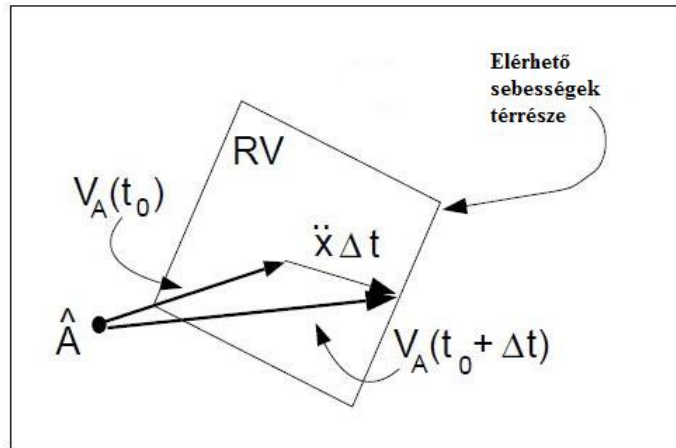
Az egyenletben n jelöli az akadályok számát, VO_{Bi} pedig az i . akadályhoz tartozó VO -t.



2-4. ábra: Több akadály esetén VO meghatározása [21]

2.2 Elérhető elkerülő sebességek térrésze

Az $\hat{A}$ robot által egy adott állapotból Δt idő alatt elérhető sebességeket ki tudjuk számítani a beavatkozó jelek és a mozgásegyenlet alapján.



2-5. ábra: Elérhető sebességek [21]

A *Feasible Accelerations* ($FA(t)$), azaz elérhető gyorsulásokat kiszámíthatjuk:

$$FA(t) = \{\ddot{x} | \ddot{x} = f(x, \dot{x}, u), u \in U\} \quad (2.5)$$

Itt x jelöli a pozíciót, $f(x, \dot{x}, u)$ a robot dinamikáját, u a beavatkozó jel, U pedig a lehetséges beavatkozó jelek halmaza.

A Δt idő alatt elérhető sebességek, *Reachable Velocities* ($RV(t + \Delta t)$) kiszámíthatóak:

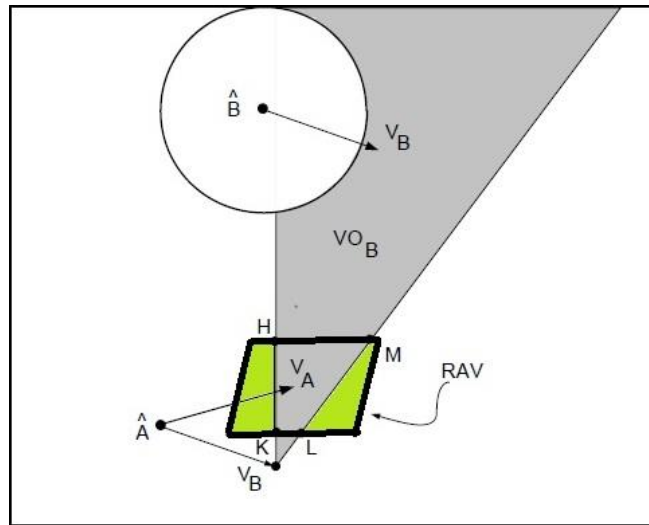
$$RV(t + \Delta t) = \{v \mid v = v_A(t) + \Delta t * FA(t)\} \quad (2.6)$$

A *Reachable Avoidance Velocities* (RAV), azaz elérhető elkerülő sebességek pedig kiszámíthatóak, amennyiben ismerjük az elérhető sebességeket, illetve a VO térrészt, mivel minden elérhető elkerülő sebesség úgy definiálható, hogy elérhető és nem okoz ütközést, azaz RV -ben benne van, VO -ban pedig nincs. Tehát definíció szerint az elérhető elkerülő sebességek:

$$RAV(t + \Delta t) = RV(t + \Delta t) - VO(t) \quad (2.7)$$

Az egyenletben a "-" operátor a RV és VO térrészek különbségét képezi.

Ezáltal az akadály elkerülését szolgáló manővert úgy tudjuk számítani, hogy ebből az elérhető elkerülő sebességek térrészből választunk sebességet.



2-6. ábra: RAV bemutatása [21]

A 2-6. ábra vastagított fekete keretében látható a robotnak az elérhető sebesség térrésze (RV), amelyet adott időegység alatt elérni képes. Amennyiben ebből a térrészből eltávolítjuk a B akadályhoz tartozó VO -t, akkor meg is kapjuk a képen zölddel sátozott elérhető elkerülő sebességek térrészét (RAV). Amennyiben ebből a térrészből választunk sebességvektort a robotnak, akkor mindenféleképpen ütközésmentes mozgást valósítunk meg.

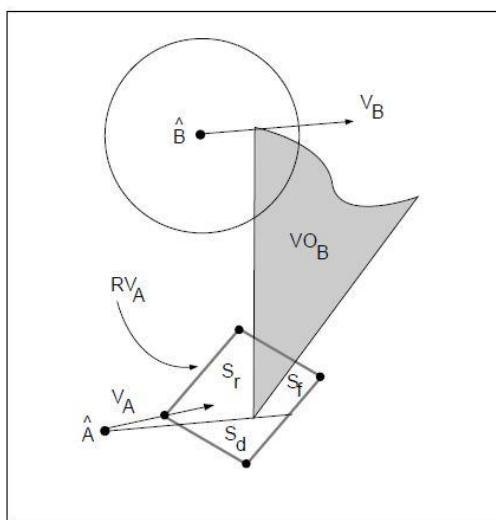
Korábbi munkám során [24] megvizsgáltam az elérhető sebességek térrészét differenciális meghajtású robotok esetén. A kinematikai egyenletek figyelembevételével megállapított elérhető sebességek térrésze során azt tapasztaltam, hogy minél nagyobb a robotnak a sebessége, oldal irányban egyre kevésbé képes elfordulni. Ám minden esetben az RV térrész jól közelíthető volt egy 3 pont által közrezárt térrésszel. A munkámban minden esetben ezzel a 3 pont által közrezárt térrésszel közelítem az elérhető sebességek térrészét, hiszen ez a kívánt elvárásoknak teljes mértékben eleget tesz.

2.3 Velocity Obstacles módszer tulajdonságai

A soron következő tételek bizonyítása megtalálható a [21]-es szakirodalom függelékében.

2.3.1 I. tétel

Tétel: Az elérhető elkerülő sebesség térrésze egy akadály esetén mindig felosztható legfeljebb 3 részre: S_f , S_r , S_d . Ezek a térrészek azon sebességeket tartalmazzák, melyek kiválasztása esetén az akadályt rendre előlről, hátulról, illetve kitérő irányban történő elkerülő manőverek valósulnak meg.



2-7. ábra: Elérhető elkerülő sebesség térrészének felosztása [21]

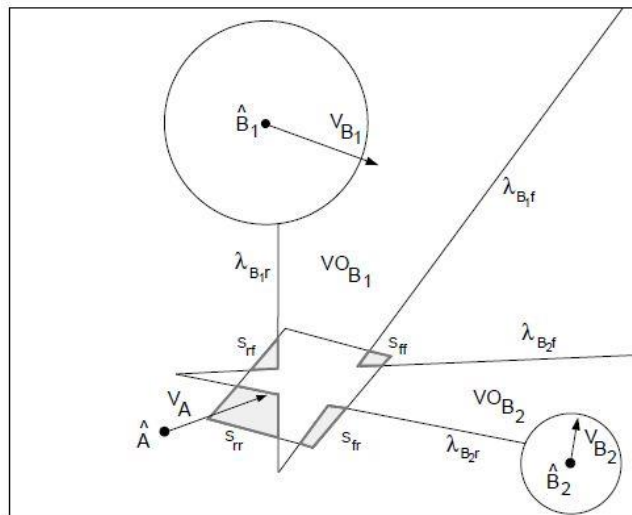
A képen látható elérhető elkerülő sebességek térrészét úgy osztjuk fel, hogy az akadályhoz tartozó VO_B csúcsát összekötjük a robotunkkal, így a VO_B határai és ezen félegyenes által 3 egymástól jól elkülöníthető térrészt kapunk: S_f , S_r , S_d . Ha az S_f térrészből választunk sebességet, akkor elkerülő manőverünk során az akadály előtt fogunk elhaladni, S_r térrészből választott sebesség esetén az akadály után fogunk

elhaladni, míg az alsó S_d térrész esetén teljesen másfelé fogunk menni, a robot és az akadály pályája nem keresztezi egymást.

Természetesen elképzelhető, hogy az elérhető elkerülő sebességek csak 1 vagy 2 részre tagolódnak. Utóbbira mutat példát a 2-6. ábra. Ebben az esetben az elérhető sebességek térrésze a VO csúcsa fölött helyezkedik el, és ennél fogva a VO csak 2 részre osztja a sebességek térrészét. Így egy S_f , S_r térrész képződik.

2.3.2 II. tétel

Tétel: N darab mozgó akadály esetén az elérhető elkerülő sebességek térrésze legfeljebb $3N$ részre osztható fel [21], mely részek tartalmazzák az elkerülő manőverekhez biztosított választható sebességeket. Ennek a tételnek a cáfolása a 7. fejezetben található.



2-8. ábra: Több akadály esetén az elérhető elkerülő sebességek felosztása [21]

Több akadály esetén a felosztott elérhető elkerülő sebességek térrészeit jelölhetjük úgy, hogy a különböző akadályokhoz képest melyik irányból fog elhaladni a robot. A fenti ábrán így az S_{ff} térrész tehát azokat a sebességeket jelöli, amelyek választása esetén a robot mindkét akadály előtt fog elhaladni, míg például az S_{fr} térrész sebességei esetén a $\hat{B}_1$ -es akadály előtt, míg a $\hat{B}_2$ -es akadály mögött fog elmenni a robot.

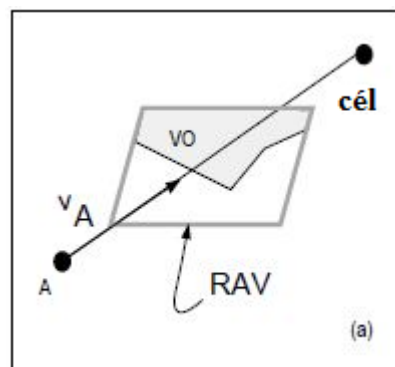
2.4 Sebességvektor választása a robot számára

Különböző lehetőségek állnak rendelkezésre, hogy a robot miként válasszon sebességvektort a RAV térrészből minden egyes mintavételi időpillanatban.

2.4.1 Heurisztikus mozgástervezés

A heurisztikus mozgástervezés legfontosabb tulajdonsága, hogy „kis számítási igénye miatt” mozgás közben, valós időben tudunk tervezni, mindig meghatározott időközönként térképezzük fel az adott teret és választjuk ki a robotnak a megfelelő sebességet, amellyel egészen a következő időpillanatig haladunk. A heurisztikus mozgástervezés során korábbi munkámban [24] két fő stratégiát implementáltam: *TG* (*to goal*), illetve az *MV* (*maximum velocity*) stratégiákat.

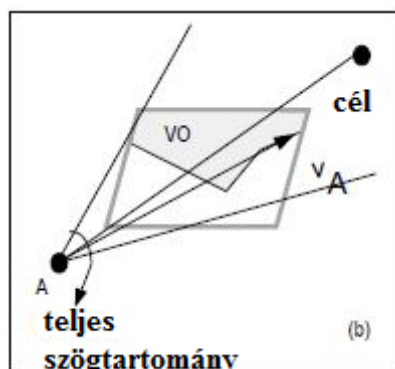
TG stratégia esetén minden adott időpillanatban kiválasztjuk a robotnak azt a legnagyobb elérhető sebességet, amely a cél irányába mutat, és nem okoz ütközést.



2-9. ábra: TG stratégia által kiválasztott legnagyobb sebesség [21]

A fenti ábrán jól látható, hogy a cél irányába mutató legnagyobb sebességet választjuk ki, amely nincs benne a *VO*-ban, azaz nem okoz ütközést, de elérhető sebesség. Tehát úgy is fogalmazhatunk, hogy kiválasztjuk a cél irányába mutató legnagyobb sebességet az elérhető elkerülő sebességek térrészből.

MV stratégia esetén minden adott időpillanatban kiválasztjuk a robotnak a cél iránya körül egy adott szögtartományba eső legnagyobb elérhető elkerülő sebességet.



2-10. ábra: MV stratégia által kiválasztott legnagyobb sebesség [21]

A 2-10. ábra mutatja, hogy a cél iránya körül egy adott szögtartományban végigpásztázzuk a környezetet, és kiválasztjuk a legnagyobb elérhető elkerülő sebességet ezen térrészen belül.

Természetesen semmi sem garantálja, hogy ezen algoritmusok felhasználásával a cél bármely időpillanatban elérhető. Lehetséges, hogy egyáltalán nem létezik megoldás, például, ha körbe van véve a robotunk statikus akadályokkal. Ekkor soha nem tudjuk elérni a célt. Ám az is elképzelhető, hogy létezne megoldás, de ezekkel a heurisztikákkal nem tudjuk megtalálni azt. Például, ha a cél és a robot között szorosan egymás mellett 2-3 nagy sugarú statikus akadály van, abban az esetben nagy kerüléssel el tudnánk érni a célt, de *TG* és *MV* stratégiákkal nem tudjuk megközelíteni.

2.4.2 A Safety Velocity Obstacles módszer

A legbiztonságosabb sebességkomponens kiválasztása érdekében kifejlesztettem a *Safety Velocity Obstacles (SVO)* módszert [22].

Minden egyes mintavételi időpillanatban meg kell vizsgálni, milyen messze található az esetlegesen kiválasztásra kerülő v (robot sebességvektorának végpontja) a legközelebbi sebesség akadálytól, VO_i -től (ahol i jelöli az adott akadályt). Nevezzük ezt a távolságot VO_{tav} -nak. VO_{tav} a legkisebb Euklideszi normát jelöli a sebességvektor végpontja és VO_i . legközelebbi pontja között. Tehát VO_{tav} a következőképpen definiálható:

$$VO_{tav}(v, VO_i) = \min_{v_{VO} \in VO_i} \|v - v_{VO}\| \quad (2.8)$$

Amennyiben a távolság nagyobb, mint egy határ, d_{max} , akkor VO_{tav} értékének ez a maximális távolság érték kerül beállításra. A maximális távolság kiszámítható a maximális sebesség és a mintavételi idő felhasználásával:

$$d_{max} = v_{max} t_{mintavétel} \quad (2.9)$$

A normalizált távolságot a következőképpen definiálhatjuk:

$$VO_{norm} = \frac{VO_{tav}}{d_{max}} \quad (2.10)$$

ahol $VO_{norm} \in [0,1]$. Ezt követően definiálni lehet a VO költség értéket:

$$VO_{koltseg} = 1 - VO_{norm} \quad (2.11)$$

Az a sebességkomponens eredményezi a legbiztonságosabb mozgást a robot számára, melynek a $VO_{koltseg}$ értéke minimális, hiszen ekkor található az összes ütközést okozó térrésztől a legtávolabb.

Ezt követően bevezetésre kerülhet egy Költségfüggvény, melynek a felhasználásával mind a sebesség, mind a biztonság ugyanabban az időpillanatban tud szerepet játszani a mozgástervezés során.

A költségfüggvény a következőképpen definiálható:

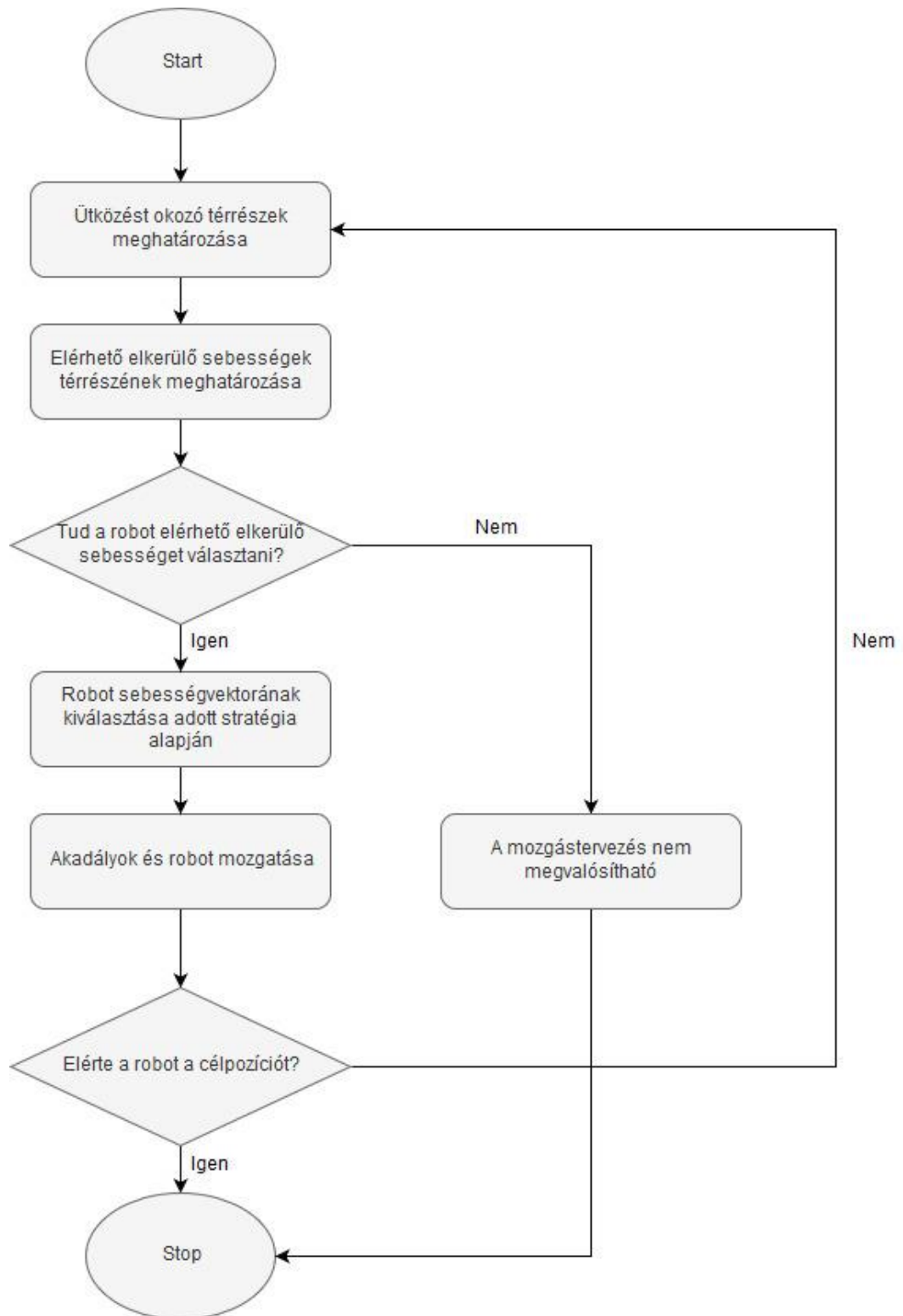
$$KTG_{ertek} = \alpha Tav_{norm} + (1 - \alpha)VO_{koltseg} \quad (2.12)$$

ahol a Tav_{norm} jelöli, hogy milyen távolságban található a célpozíció attól a pozíciótól, ahova a robot akkor tud elérni, ha a kiválasztott sebességvektort használja fel egy mintavételi időegységig. Ezt normálni kell a robot kiindulási pozíciója és célpozíciója közötti távolság értékével:

$$Tav_{norm}(v) = \frac{\|vt_{mintavétel} - P_{cel}\|}{\|P_A(0) - P_{cel}\|} \quad (2.13)$$

ahol $t_{mintavétel}$ a mintavételi idő, P_{cel} a célpozíció, $P_A(0)$ a robot kezdőpozíciója, $VO_{koltseg}$ korábban definiálásra került (2.11); α az aránytényező, az értékét tekintve 0 és 1 közötti számokat vehet fel. Az aránytényező azt mutatja meg, hogy a sebesség, illetve a biztonság mekkora szerepet játszik a pályatervezés során. A legbiztonságosabb mozgás $\alpha = 0$ paraméterérték mellett következik be. Amennyiben $\alpha = 1$ érték teljesül, ez eredményezi a leggyorsabb pályatervezést. Ha $\alpha \in [0,1]$, akkor a sebesség és a biztonság különböző mértékben játszik szerepet a robot pályatervezése során. Ezeket a módszereket felhasználva tud a robot sebességkomponenst választani.

3 A mozgástervező algoritmus



3-1. ábra: Mozgástervező algoritmus folyamatábrája

Az ütközésmentes mozgástervezés során a 3-1. ábra által bemutatott lépésekből álló algoritmust kell megvalósítanunk az eredményes célelérés érdekében.

Tehát első lépésben mindig az összes akadályhoz tartozó VO térrészeket kell meghatározni. Az így kapott egyes akadályokhoz tartozó VO térrészek uniója alkotja az ütközést okozó sebességek térrészét.

Majd a robot aktuális pozíciójának, előző sebességvektorának ismeretében meg tudjuk határozni az elérhető sebességek térrészét.

Amennyiben nem található elérhető elkerülő sebességkomponens, akkor a mozgástervező algoritmus nem tud a robot számára sebességvektort választani az adott mintavételi időpillanatban, ezáltal a mozgástervezés nem megvalósítható.

Ellenkező esetben a robotnak sebességet választunk az elérhető, nem ütközést okozó sebességek térrészből meghatározott szempontok alapján. (Leggyorsabb, legbiztonságosabb, szabálykészlet alapján).

Ezzel a sebességvektorral haladunk előre meghatározott időegységig, majd amennyiben még nem értük el a célt, akkor újból kezdődik az egész környezet feltérképezése.

Így haladva jutunk el a kívánt célpozícióig.

Az algoritmus implementálását MATLAB fejlesztői környezetben végeztem. Az elérhető sebességek térrészének pontos meghatározása és egy adott sebességkomponens elérhetőségének, ütközésének vizsgálata korábbi munkámban olvasható [24].

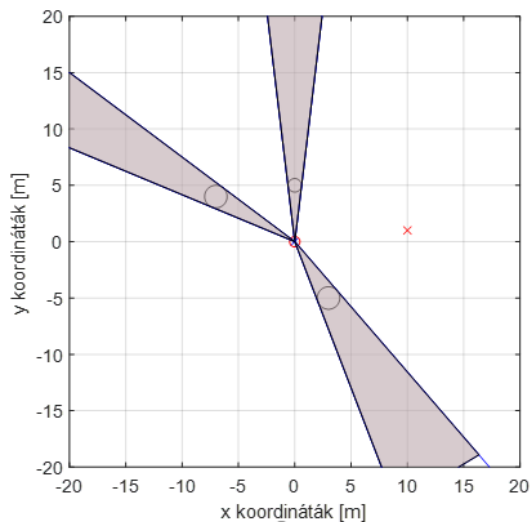
4 Ütközést okozó robotsebességek meghatározása

Ahhoz, hogy meghatározzuk az ütközést okozó sebességek térrészét egy adott időpillanatban, ismernünk kell, vagy a megfelelő szenzorokkal valós időben mérnünk kell a robot, illetve az akadályok sugarát, sebességvektorát, a robot- és az akadályok pozícióját. A robot és az akadályok kezdeti pozícióját, sugaraikat, az akadályok sebességkomponenseit ismertnek feltételezzük, illetve a robot a Descartes koordinátarendszer origójában van, orientációja 0 fok. Amennyiben a robot kezdetben nem a koordinátarendszer origójában helyezkedik el, abban az esetben is egy egyszerű geometriai transzformációval, eltolással és forgatással a középpontba helyezhető (transzformálva az akadályok középpontjait is).

Minden egyes akadály esetén a következőképpen lehet meghatározni az ütközést okozó robotsebességek térrészét:

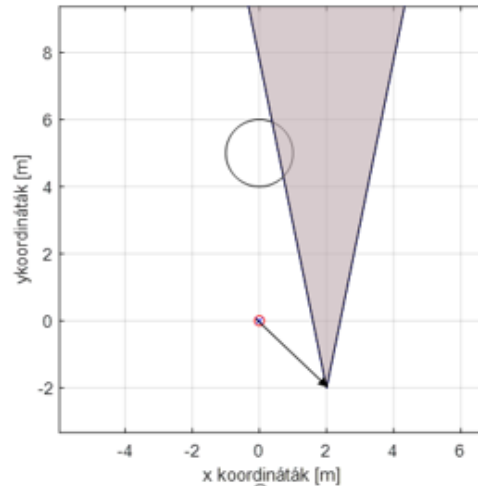
Először érintőket kell húznunk a robot pozíciójából az adott akadályhoz, ehhez ki kell számítanunk az érintési pontokat. A megoldás implementálása és részletes kidolgozása megtalálható a korábbi munkámban [24].

Az érintési pontokhoz félegyenéseket húzva meghatározhatóak a CC térrészek, ahogy a 4-1. ábra mutatja.



4-1. ábra: 3 akadályhoz tartozó CC térrészek meghatározása

Egy adott akadályhoz tartozó CC térrész ismeretében, az akadálysebességgel való eltolás eredményeképpen megkapjuk a teljes VO térrészt:



4-2. ábra: Ütközést okozó sebességek

4-2. ábra szürke színű területe jelöli a teljes VO térrészt, piros kör a robot pozícióját, fekete kör az akadályt szemlélteti, a nyíl pedig az akadály sebességvektorát, amellyel eltoltuk a CC térrészt.

Az ütközést okozó sebességek térrészét minden egyes mintavételi időpillanatban kiszámítjuk, hiszen ez alapján tudjuk meghatározni, hogy melyek a „szabad” térrészek, ahonnan sebességet választva a jövőben nem történik majd ütközés a robot és az adott akadály között. Láthattuk, hogy a kiszámításhoz minden egyes időpillanatban szükség van az akadály sebességvektorának ismeretére. Az akadályokról azt feltételezhetjük, hogy nem kell egyenes vonalú egyenletes mozgást végezniük, de azzal a közelítéssel élünk, hogy egy adott mintavételi időegységig nem változtatják meg a sebességvektorukat.

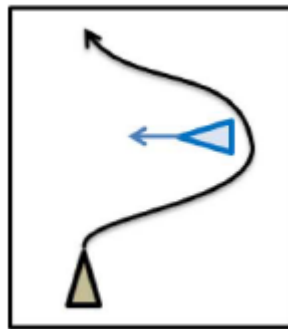
5 Mozgástervezés hajók számára szabálykészlet alapján

A hajók számára történő mozgástervezés során egy új irányelvet, szabálykészlet alapján történő pályatervezés megoldását alkalmazták a Velocity Obstacles módszer felhasználásával, amelyben mindig 1 hajó és 1 akadály esetében oldották meg a mozgástervezést. [23]. Ebben a fejezetben bemutatom a hajók esetén milyen szabályok alapján történik a mozgástervezés.

5.1 Szabályok alapján elkülöníthető szituációk

A szabályok alapján 4 különböző szituációt különböztetünk meg, amelyekre alkalmazni lehet a szabálykészletet (minden esetben fekete háromszög jelöli a saját hajót, kék háromszög a másik hajót, nevezzük akadállynak), ezek a következők:

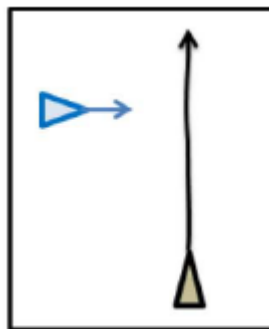
1. Crossing form right (jobbról érkezik az akadály)



5-1. ábra: Jobbról érkezik az akadály [23]

Az 5-1. ábra bemutatja azt a szituációt, amikor jobbkéz szabály alapján elsőbbséget kell adni, utána lehet tovább haladni a hajóval.

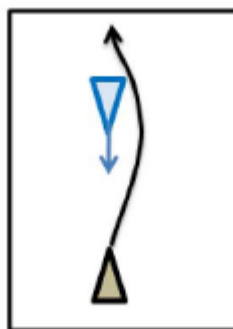
2. Crossing from left (balról érkezik az akadály)



5-2. ábra: Balról érkezik az akadály [23]

A 5-2. ábra szemlélteti azt a szituációt, amikor a hajó előbb elhaladhat a cél irányába, mivel balról érkezik az akadály

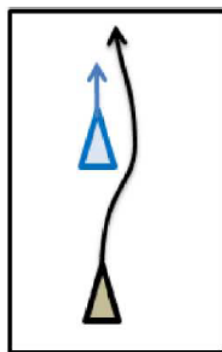
3. Head-on (szemből érkezik az akadály):



5-3. ábra: Szemből érkezik az akadály [23]

Az 5-3. ábra prezentálja azt az esetet, amikor szemből érkezik az akadály, ebben az esetben jobbra tér ki a hajó.

4. Overtaking (előzés):



5.

5-4. ábra: Előzési szituáció [23]

Abban az esetben, amennyiben a hajó előtt halad az akadály, akkor jobbról történhet az előzés, ahogy az 5-4. ábra szemlélteti.

Azt, hogy melyik szituációról van szó és milyen sebességek közül választhat a robot, a szabálykiválasztó dönti el. Ám nem minden esetben kell szabályok alapján dönteni, csak ha a hajó és az akadály mozgásuk során belátható időn belül megfelelő közelségbe kerülnek.

5.2 Előszűrés a szabályok alkalmazására

Annak eldöntésére, hogy kell-e szabályokat alkalmazni, egy előszűrést kell végezni: ki kell számítani a legközelebbi pontot, mikor lesz a hajó és az akadály egymáshoz legközelebb a mozgásuk során, felhasználva a hajó és az akadály aktuális pozícióját és sebességkomponenseiket. Ez az előszűrés biztosítja, hogy csak azon akadályok esetén alkalmazzuk a szabályokat, amelyek valóban hatással vannak a hajó mozgására. Ezáltal gyorsabb feldolgozást is elérünk. A számításokhoz a következő egyenleteket alkalmazhatjuk:

$$t_{LKP} = \frac{(p_A - p_B) * (v_A - v_B)}{\|v_A - v_B\|^2} \quad (5.1)$$

ahol t_{LKP} jelöli azt az időpontot, amikor a hajó és az akadály a legközelebb lesznek egymáshoz, p_A jelöli a hajó pozíciójának vektorát, p_B az akadály pozíciójának vektorát, v_A jelöli a hajó sebességvektorát, v_B az akadály sebességvektorát az adott időpillanatban, a $*$ pedig skaláris szorzatot jelöl.

$$d_{LKP} = \|(p_A + v_A t_{LKP}) - (p_B + v_B t_{LKP})\| \quad (5.2)$$

ahol d_{LKP} jelöli azt a távolságot, amely akkor mérhető a hajó és akadály között, amikor legközelebb vannak egymáshoz (a többi paraméter megegyezik az (5.1)-ben felsorolt elemekkel).

Azonban a [23]-as szakirodalomban található fenti egyenletekben található egy hiba, melynek levezetése a következő, 6. fejezetben található.

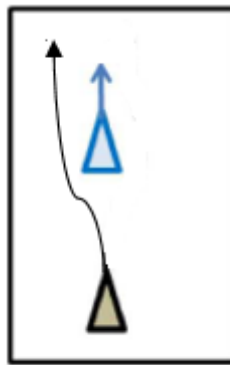
6 Traffic Regulation Velocity Obstacles módszer

Az általam kifejlesztett Traffic Regulation Velocity Obstacles módszer esetén a KRESZ szabályokat használjuk fel a robot pályatervezése folyamán mobilis robotok és autószerű járművek esetén.

A TDK dolgozatomban során azokat a közlekedési szituációkat vizsgálom, melyeknél közlekedési tábla és lámpa nem irányítja a forgalmat. Természetesen a módszer a későbbiekben kibővíthető olyan kereszteződésekre is, ahol táblák szabályozzák az elsőbbséget. Az 5. fejezetben felsorolt hajózási szabályokat tekintve az első három esetben megegyeznek a szabályok, azaz:

- Amennyiben jobbról érkezik az akadály, akkor elsőbbséget kell neki adni.
- Ha balról érkezik, akkor elmehet a robot előtte.
- Amennyiben szemből érkezik az akadály, akkor jobbról kerüljük ki.

Viszont amennyiben a robot előtt található az akadály, és előzni szeretne, akkor a KRESZ értelmében balról kell a manővert végrehajtani, ahogy az 6-1. ábra mutatja:



6-1. ábra: Amennyiben a robot előtt található az akadály, és előzni szeretne, akkor balról lehet végrehajtani a manővert

Az előszűrést tekintve a következő egyenletekkel vezethető le a (5.1) és (5.2), melyekkel megkapható a helyes összefüggés:

Legyen a robot pozíciója $t_0 = 0$ -ban $P_{r0} = [p_{rx}, p_{ry}]$, a robot sebessége pedig $v_r = [v_{rx}, v_{ry}]$. Az akadály pozíciója $t_0 = 0$ -ban $P_{a0} = [p_{ax}, p_{ay}]$, az akadály sebessége pedig $v_a = [v_{ax}, v_{ay}]$.

Kiszámítható a robot és az akadály pozíciója t idő múlva:

Robot pozíciója:

$$P_r(t) = P_{r0} + t * v_r = \begin{bmatrix} prx \\ pry \end{bmatrix} + t * \begin{bmatrix} vrx \\ vry \end{bmatrix} = \begin{bmatrix} prx + t*vrx \\ pry + t*vry \end{bmatrix} \quad (6.1)$$

Akadály pozíciója:

$$P_a(t) = P_{a0} + t * v_a = \begin{bmatrix} pax \\ pay \end{bmatrix} + t * \begin{bmatrix} vax \\ vay \end{bmatrix} = \begin{bmatrix} pax + t*vax \\ pay + t*vay \end{bmatrix} \quad (6.2)$$

Ezt követően kiszámítható az akadály és a robot távolsága t időpillanatban:

$$\begin{aligned} d(t) &= || \begin{bmatrix} prx + t*vrx \\ pry + t*vry \end{bmatrix} - \begin{bmatrix} pax + t*vax \\ pay + t*vay \end{bmatrix} || = \\ &= || \begin{bmatrix} (prx - pax) + t*(vrx - vax) \\ (pry - pay) + t*(vry - vay) \end{bmatrix} || = \\ &= \sqrt{[(prx - pax) + t*(vrx - vax)]^2 + [(pry - pay) + t*(vry - vay)]^2} \end{aligned} \quad (6.3)$$

Azt a minimális időpillanatot keressük, amikor a legközelebb lesz egymáshoz a robot és az akadály:

$$t_{min} = \operatorname{argmin} d(t) \quad (6.4)$$

Tehát keressük a távolság idő szerinti deriváltjának a minimumát:

$$\begin{aligned} \frac{d d(t)}{dt} &= \frac{1}{2} * \frac{1}{\sqrt{[(prx - pax) + t*(vrx - vax)]^2 + [(pry - pay) + t*(vry - vay)]^2}} * \\ &= [2 * (prx - pax) * (vrx - vax) + 2 * t * (vrx - vax)^2 + \\ &+ 2 * (pry - pay) * (vry - vay) + 2 * t * (vry - vay)^2] \end{aligned} \quad (6.5)$$

Akkor lesz minimális, amennyiben a derivált értéke 0, $\frac{d d(t)}{dt} = 0$.

Tehát (6.5) alapján a következő egyenletet kapjuk:

$$\begin{aligned} &(prx - pax) * (vrx - vax) + t * (vrx - vax)^2 + \\ &(pry - pay) * (vry - vay) + t * (vry - vay)^2 = 0 \end{aligned} \quad (6.6)$$

A (6.6) alapján kifejezhető a t_{LKP} paraméter, amely azt mutatja meg, hogy mennyi idő múlva lesz legközelebb egymáshoz a robot, illetve az akadály:

$$t_{LKP} = \frac{-(prx - pax) * (vrx - vax) - (pry - pay) * (vry - vay)}{(vrx - vax)^2 + (vry - vay)^2} \quad (6.7)$$

A (6.7) számlálója felírható skaláris szorzással is, a nevezője pedig Euklideszi norma négyzeteként. Legyen, p_A robot pozíciója, p_B az akadály pozíciója, v_A jelölje a robot sebességvektorát, v_B az akadály sebességvektorát. Ezáltal kapjuk a (6.8)-at:

$$t_{LKP} = \frac{-(p_A - p_B) * (v_A - v_B)}{||v_A - v_B||^2} \quad (6.8)$$

Látható, hogy a (6.8) az (5.1) (-1)-szerese. A negatív előjel nélkül minden esetben negatív időtartamokat kapnánk, amely helytelen megoldás. Ezzel bizonyítottuk, hogy a [23]-ban szereplő (5.1) hibás volt. ■

Amennyiben ismerjük az időpontot, hogy mikor lesz legközelebb a robot az akadályhoz, akkor kiszámítható a kettejük közötti távolság:

$$d_{LKP} = ||(p_A + v_A t_{LKP}) - (p_B + v t_{LKP})|| \quad (6.9)$$

Tehát az adott akadály esetén ki kell számítani, hogy a közeljövőben mikor lenne az akadály legközelebb a robothoz és ennek felhasználásával tudjuk eldönteni, hogy használjuk-e a szabályok alapján történő tervezést. Abban az esetben, ha a következő egyenlet teljesül, akkor alkalmazhatjuk a szabályok alapján történő tervezést:

$$0 < t_{LKP} < t_{max} \text{ és } d_{LKP} < d_{min} \quad (6.10)$$

ahol $t_{max} = 3 * t_{mintavétel}$, $d_{min} = 3 * v_{maxseb} * t_{mintavétel}$ értékeket állítottam be ($t_{mintavétel}$ a mintavételi időt, v_{maxseb} a maximálisan megengedett sebességnagyságot jelöli).

Ha teljesül a (6.10), akkor alkalmazhatjuk a szabály alapú tervezést a következő megvalósítással.

6.1 Szabályok alapján történő tervezés

Legelőször egy akadály esetén történik a szabályok figyelembevétele. Ezt követően kerül kibővítésre az algoritmus. Minden egyes szituációban meg kell vizsgálni, hogy az elérhető sebességekből milyen elérhető elkerülő sebességeket kapunk, ha kivonjuk a területből az ütközést okozó VO térrészt. Ekkor 1, 2, vagy 3 területrésze oszthatjuk fel az elkerülő elérhető sebességek térrészét a 2.3.1. fejezetet alapján. Amennyiben 3 részre oszthatjuk fel, akkor megkülönböztetünk elkerülő sebességkomponenseket, olyan sebességeket, amelyeket a robotnak választva az adott akadály előtt haladunk el, illetve olyan sebességeket, melyeket választva az akadály

mögött haladunk el. A különböző szituációk esetén az a kérdés, hogy mely területből választhatunk a robotnak sebességet.

Minden egyes időpillanatban egy transzformációt hajtunk végre, annak érdekében, hogy a szabályokat könnyebben tudjuk azonosítani. Ez a transzformáció egy forgatásnak felel meg a z tengely körül (amely merőleges mind az x, mind az y tengelyre), melyet az (6.11), (6.12)-vel tudunk megtenni:

$$x_{uj} = x * \cos(\theta_{forgatas}) - y * \sin(\theta_{forgatas}) \quad (6.11)$$

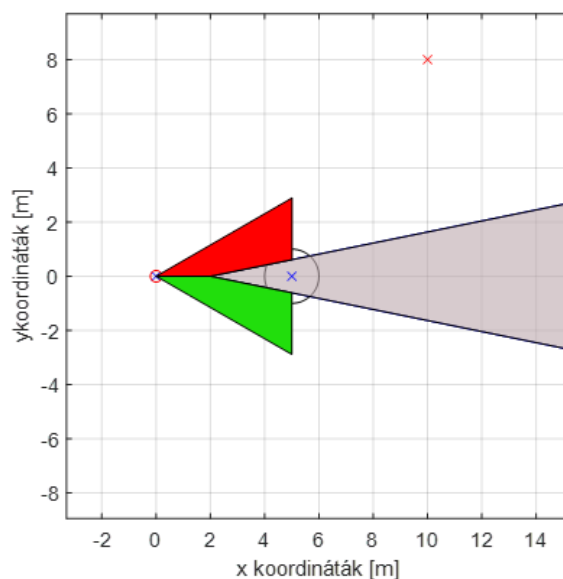
$$y_{uj} = x * \sin(\theta_{forgatas}) + y * \cos(\theta_{forgatas}) \quad (6.12)$$

ahol a $\theta_{forgatas}$ az a szög, amellyel forgatni szeretnénk. Jelenleg a robot aktuális sebességvektorának x tengellyel bezárt szögének (-1)-szeresével forgatunk annak érdekében, hogy a robot sebességvektora az x tengely pozitív irányába mutasson, azaz 0° legyen, így az akadály sebességvektorával bezárt szögét könnyű kiszámítani. Ezt a forgatást az akadály pozíciójára és sebességvektorára hajtjuk végre. A pozíció esetén az akadály koordinátáiból még transzformáció előtt ki kell vonni a robot aktuális pozícióját annak érdekében, hogy a szabályokat úgy tudjuk meghatározni, mintha a robot az origóban lenne. Innentől a transzformált értékekre vonatkoznak a kritériumok, vizsgálatok.

A KRESZ előírásait tekintve a következő szabályokat alkothatjuk (a szabályok minden esetben úgy kerülnek felírásra, hogy a robot kezdetben az origóban van, és a pozitív x tengely irányába mutat az orientációja):

6.1.1 Előzés

Abban az esetben, ha az akadály x koordinátája nagyobb, mint a robot x koordinátája, az akadály sebességvektorának a szögére, melyet $\Delta\theta$ -val jelölünk, igaz, hogy $-\theta < \Delta\theta < \theta$ (egy tipikus numerikus érték $\theta = 5^\circ$) között van és az y koordinátájából kivonva a robot pozíciójának y koordinátáját, ha kisebb értéket kapunk, mint az akadály megnövelt sugara, akkor a robot előtt található az akadály. Erre mutat egy példát az 6-2. ábra. Ebben az esetben össze kell kötni az elérhető sebességek kezdőpozícióját a VO térrész csúcsával és ezáltal lehet 2 részre osztani az elérhető elkerülő sebességeket, ezt is láthatjuk:



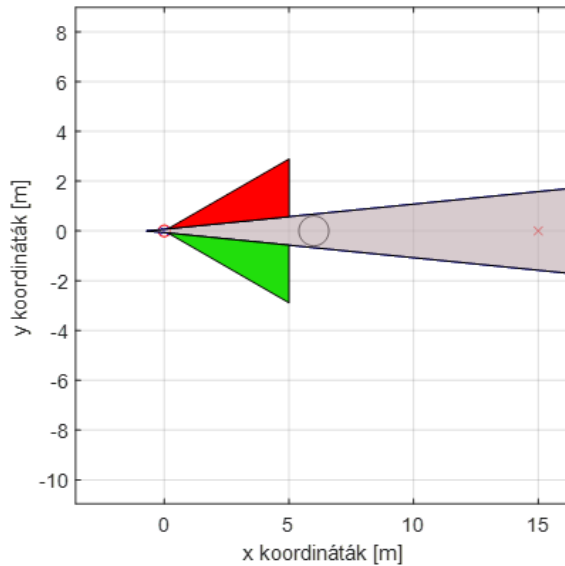
6-2. ábra: Előzés

6-2. ábra egy előzési szituációt mutat be. Piros kör jelzi az origóban a robotot, (5;0)-ban található az akadály, piros és zöld térrész jelöli az elérhető elkerülő sebességeket, szürke a VO térrész. Látható a két részre bontott elérhető sebességek térrésze. Piros jelöli azokat a sebességeket, melyből választva az akadályt balra kerüljük ki, zöld azokat, melyekből választva jobbra kerüljük ki az akadályt. Ebben az esetben KRESZ szabályok alapján a piros színnel körülfogott területet adjuk vissza, ebből a térrészből választunk sebességet a robotnak.

Tehát abban az esetben, ha előttünk található az akadály és előzni szeretnénk a célelérés érdekében, akkor a robot balról megkerülve tudja ezt a mozgást végrehajtani. Ezt mutatja be a következő videó: <https://youtu.be/4Q27cNcPJ7I>.

6.1.2 Szemből érkezik az akadály

Akkor beszélhetünk ilyen szituációról, ha az akadály sebességének orientációja esetén teljesül, hogy $\Delta\theta < \theta_1$ vagy $\theta_2 < \Delta\theta$ (ahol $\theta_1 = -180^\circ + d\theta$, $\theta_2 = 180^\circ - d\theta$ tipikus értékekkel rendelkezik, $d\theta$ a legtöbbször 5° értéket vesz fel), illetve x, y koordinátáira ugyanaz a feltétel teljesül, mint az 6.1.1-es fejezetben bemutatott szabály esetén. Erre a szabályra mutat egy példát az 6-3. ábra:



6-3. ábra: Szemből érkezik az akadály

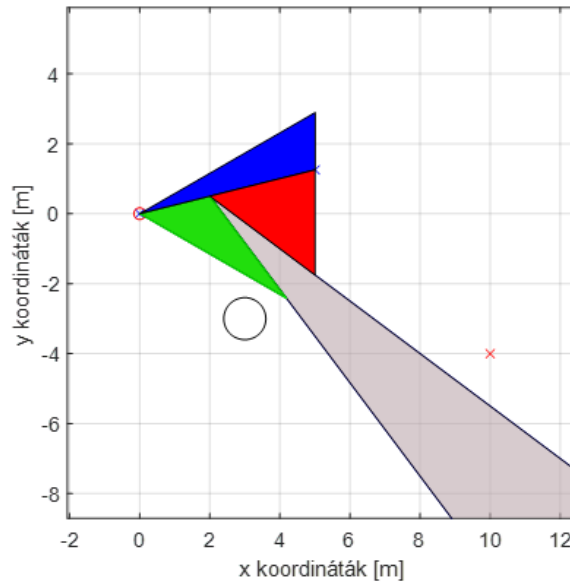
6-3. ábra egy robot számára szemből érkező akadály esetét szemlélteti. Piros kör jelzi az origóban a robotot, (5;0)-ban található az akadály, piros és zöld térrész jelöli az elérhető elkerülő sebességeket, szürke a VO térrész. Mint az előző esetben is, két részre bontotta VO az elérhető sebességek térrészét. Piros jelöli azokat a sebességeket, melyből választva az akadályt balra kerüljük ki, zöld azokat, melyekből választva jobbra kerüljük ki az akadályt. Ebben az esetben KRESZ szabályok alapján a zöld színnel körülfogott területet adjuk vissza, ebből a térrészből választunk sebességet a robotnak.

Tehát abban az esetben, ha szemből érkezik az akadály, akkor azt jobbról kell kikerülnie a robotnak mozgása során. Erre mutat példát a következő videó: <https://youtu.be/-2Wsvlcxslo>.

6.1.3 Jobbról érkezik az akadály

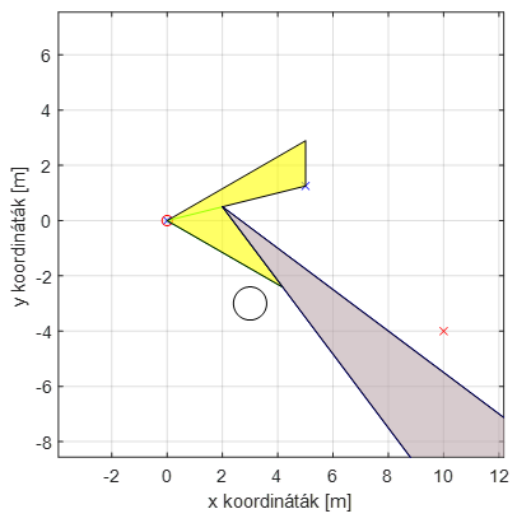
Abban az esetben beszélhetünk erről a szabályról, amennyiben az akadály transzformált y koordinátája kisebb, mint 0 értékű, illetve az akadály sebességvektorának a szöge $\theta_1 < \Delta\theta < \theta_2$ (ahol $\theta_1 = 0^\circ + d\theta$, $\theta_2 = 180^\circ - d\theta$ tipikus értékekkel rendelkezik, ahol $d\theta$ a legtöbbször 5° értéket vesz fel).

Amennyiben a VO térrész csúcsa az elérhető sebességek térrészén belül helyezkedik el, akkor 3 térrészre osztható az elérhető elkerülő sebességek, melyet az 6-4. ábra szemléltet:



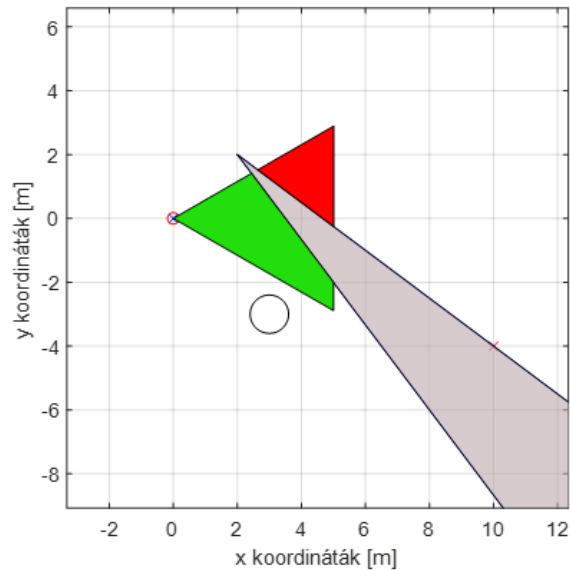
6-4. ábra: Jobbról érkezik az akadály

6-4. ábra szemlélteti azt az esetet, amikor jobbról érkezik az akadály. Piros kör jelzi az origóban a robotot, (3,-3)-ban található az akadály, szürke a VO térrész. Zöld terület rész jelöli azokat a sebességeket, melyeket választva az akadály mögött haladunk el, kék szín jelöli azon robotsebességeket, melyekből választva nem találkozik a robot mozgása során az akadállyal (elkerülő sebességek), a piros térrészből választva az akadály előtt haladunk el. Mivel jobbról érkezik az akadály, ezért elsőbbséget kell neki adni, így csak a zöld és kék színnel jelölt térrészekből választhatunk sebességet a robot számára. A 6-5. ábra szemlélteti azokat a sebességkomponenseket, melyeket kiválaszthatja a robot az adott mintavételi időben.



6-5. ábra: Sárga térrész jelöli azokat a sebességkomponenseket, amelyeket kiválaszthat a robot

Abban az esetben, ha a VO csúcsa kívül esik az elérhető sebességek tartományán és két térrészre osztja az elérhető sebességek térrészét, akkor akadály előtt, illetve akadály mögött elhaladó robotsebességeket különböztetünk meg. Ebben az esetben egyértelműen csak az akadály mögött elhaladó sebességkomponensekből választhatunk sebességet a robot számára (azaz a zöld színnel jelölt részből), ezt mutatja a 6-6. ábra.



6-6. ábra: Az elérhető sebességek térrésze két részre oszlik

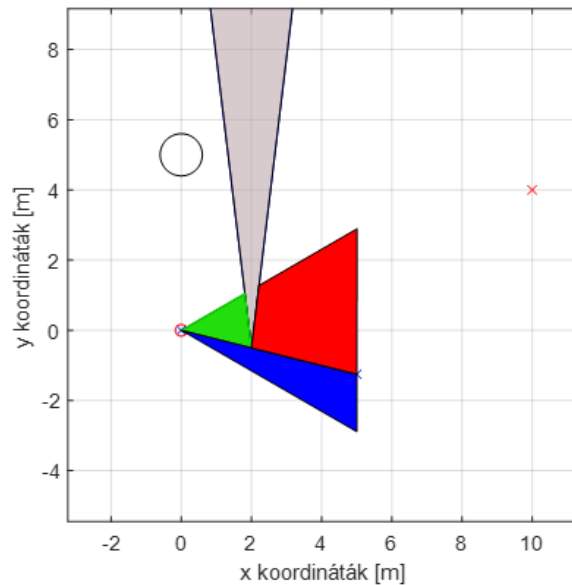
Tehát abban az esetben, ha jobbról érkezik az akadály, elsőbbséget kell neki adni. (Akadály mögött elhaladó sebességkomponensekből választhatunk, vagy ha van, elkerülő sebességkomponensekből). Ezt a mozgást szemlélteti a következő videó: <https://youtu.be/bmAqTccjXVU>.

Abban az esetben, ha a VO teljes térrésze az elérhető sebességek térrészén kívül esik, akkor a teljes elérhető sebességek térrészéről választhatunk sebességkomponenst a robot számára.

6.1.4 Balról érkezik az akadály

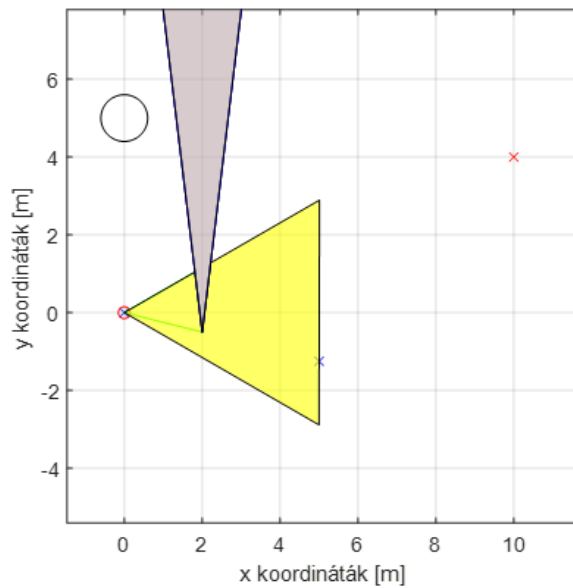
Abban az esetben beszélhetünk erről a szabályról, amennyiben az akadály y koordinátája nagyobb, mint 0, illetve az akadály sebességvektorának a szögére teljesül, hogy $\theta_1 < \Delta\theta < \theta_2$ (ahol $\theta_1 = -180^\circ + d\theta$, $\theta_2 = 0^\circ - d\theta$ tipikus értékekkel rendelkezik, $d\theta$ a legtöbbször 5° értékű).

Úgy, mint a 6.1.3-as fejezetben, itt is megkülönböztethetünk olyan esetet, amikor 3, illetve amikor 2 részre osztja a VO térrész az elérhető sebességek térrészét, ezt mutatja az 6-7. ábra és a 6-9. ábra:

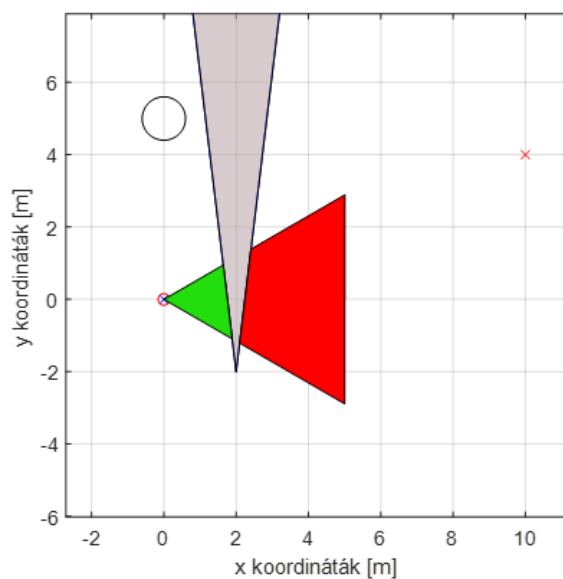


6-7. ábra: Balról érkezik az akadály

A 6-7. ábra jelöli azt a szituációt amikor balról érkezik az akadály. Piros kör jelzi az origóban a robotot, (0,5)-ben található az akadály. Zöld területrészt jelöli azokat a sebességeket, melyeket választva az akadály mögött haladunk el, kék szín jelöli azon robotsebességeket, melyekből választva nem találkozunk a robot mozgása során az akadállyal (elkerülő sebességek), a piros térrészből választva az akadály előtt haladunk el. Mivel balról érkezik az akadály. Így a KRESZ szabályai alapján a robotnak van elsőbbsége, de le is mondhat az elsőbbségéről, ezáltal, amikor sebességet keresünk, akkor az összes olyan elérhető sebesség térrészben kereshetünk, mely nem okoz ütközést. Ez látható a 6-8. ábra szemléltetésében.



6-8. ábra: Sárga térrész jelöli a robot által kiválasztható sebességkomponenseket



6-9. ábra: 2 térrészre oszlik az elérhető sebességek térrésze

A balról érkező akadály esetére mutat egy példát a következő videó:
<https://youtu.be/wBKSq1TtcTs>.

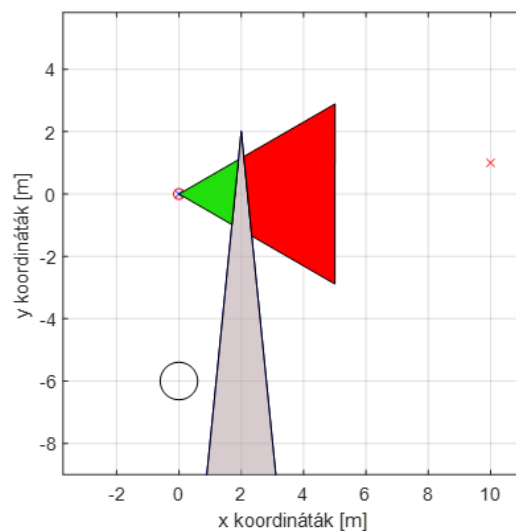
Abban az esetben, ha a VO teljes térrésze az elérhető sebességek térrészén kívül esik, akkor a teljes elérhető sebességek térrészről választhatunk sebességkomponenst a robot számára.

6.2 Megkülönböztető jelzést használó jármű esetén történő pályatervezés

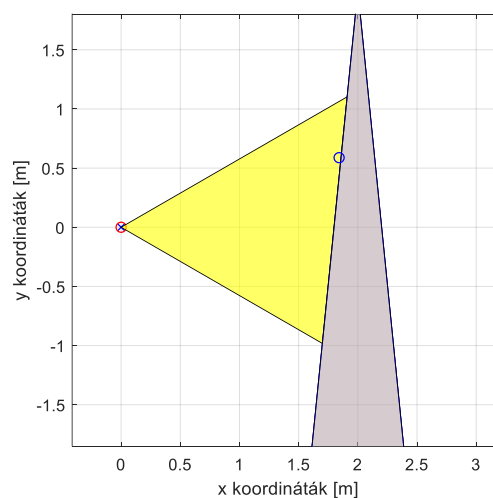
Abban az esetben, ha a robot, vagy az akadályok bármelyike megkülönböztető jelzéssel rendelkezik (azaz tegyük fel, hogy rendőr, tűzoltó, vagy mentőről beszélünk), akkor változik azoknak a sebességkomponenseknek a térrésze, amelyekből a robot sebességet választhat magának a mozgástervezés során.

6.2.1 A robot megkülönböztető jelzéssel rendelkezik:

Elsőként vizsgáljuk meg azt a szituációt, amikor jobbról érkezik egy akadály a 6-10. ábra szerint.



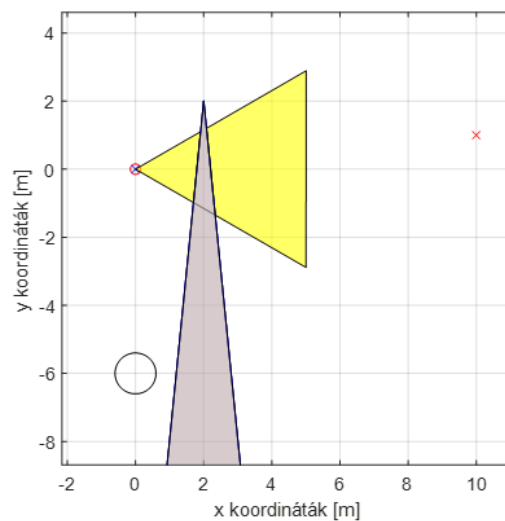
6-10. ábra: Jobbról érkező akadály 2 részre bontja az elérhető elkerülő sebességek térrészt



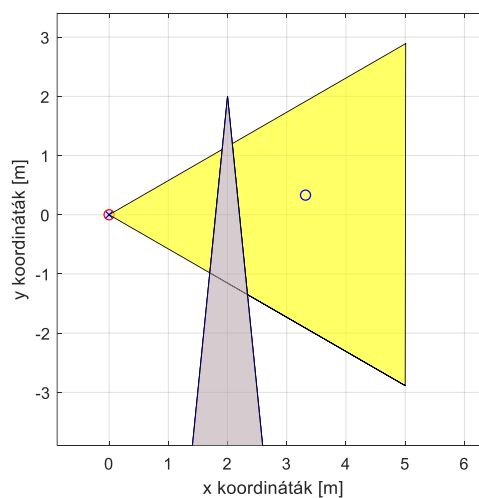
6-11. ábra: Kék kör jelzi a kiválasztott robotsebesség vektorának végpontját

Amennyiben sem a robot, sem az akadály nem rendelkezik megkülönböztető jelzéssel, akkor a robot csak az akadály mögött elhaladó sebességkomponensekből választhat, ahogy azt a 6-11. ábra szemlélteti. Ezt a mozgást mutatja be a következő videó: <https://youtu.be/ItvAXCVHvdA>.

Amennyiben a robot megkülönböztető jelzéssel rendelkezik, akkor a bármely elérhető, ütközést nem okozó sebességkomponens közül választhat, ahogy a 6-12. ábra mutatja. Ebben az esetben a robot az akadály előtt elhaladó sebességek közül is választhat sebességet, ahogy a 6-13. ábra szemlélteti.



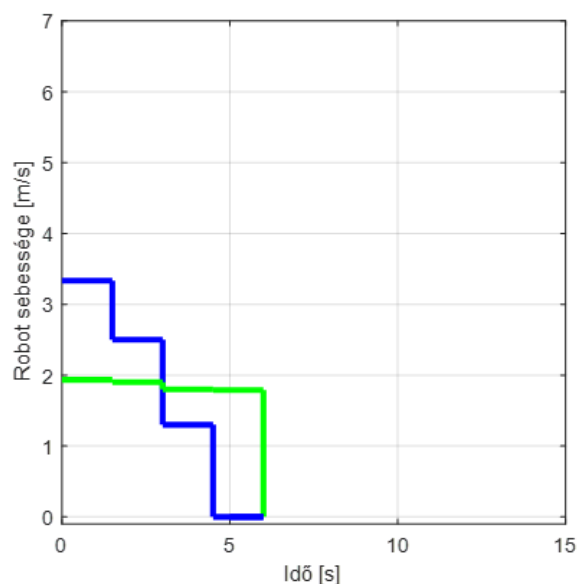
6-12. ábra: Sárga szín jelzi azokat a sebességkomponenseket, melyeket a robot kiválaszthat, amennyiben megkülönböztetett jelzésű



6-13. ábra: A robot az akadály előtt elhaladó sebességkomponensek közül választ sebességet (kék kör)

A megkülönböztető jelzés esetén történő mozgást mutatja be a következő videó:
<https://youtu.be/z-kjEQWlQRI>

Érdekes összehasonlítani a kiválasztott sebességek nagyságának alakulását a különböző eseteket megvizsgálva. A 6-14. ábra mutatja be ezeket a sebességeket. Az implementációban figyelembe vettük, hogy a célhoz közeledve csökken a robot maximális sebességének a nagysága, amelyet kiválaszthat annak érdekében, hogy soha ne fusson túl a célpozíción. Kék szín jelzi a robot sebességének változását amennyiben megkülönböztető jelzéssel bír, zöld szín jelzi a megkülönböztető jelzés nélküli sebességkomponenseket. Azt tapasztaljuk, hogy a megkülönböztető jelzés esetén, lévén, hogy az akadály előtt elhaladó sebességekből is választhat önmagának sebességkomponenst, 1 mintavételi időegységgel előbb ($t_{mintavétel} = 1,5 \text{ s}$) éri el a célpozíciót.



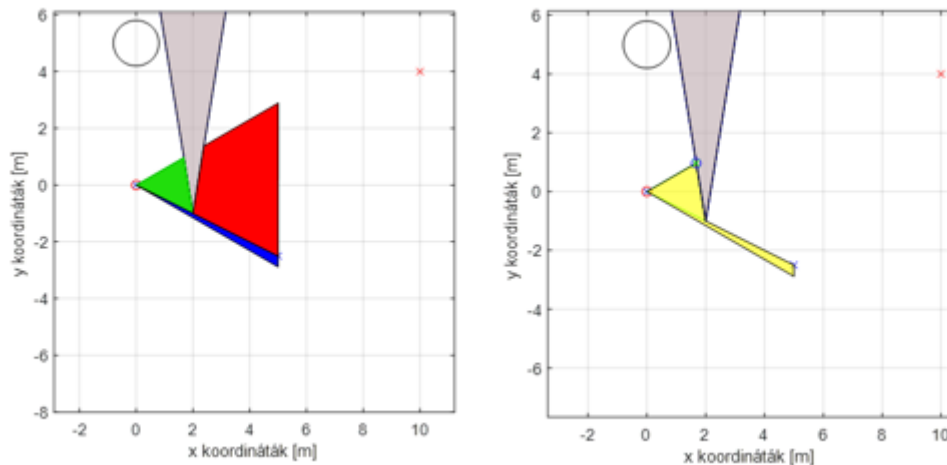
6-14. ábra: Sebességek nagysága a mozgás során megkülönböztető jelzéssel és anélkül

6.2.2 Megkülönböztető jelzéssel rendelkező akadály

Abban az esetben, ha az akadály rendelkezik megkülönböztető jelzéssel, akkor a robotnak le kell mondania az elsőbbségéről abban az esetben is, amennyiben normál közúti szituációban neki lenne lehetősége előbb elhaladni a cél irányába.

Erre az esetre a legszemléletesebb példa, ha egy robothoz képest balról érkező akadály esetét tesszük vizsgálatunk célkeresztjébe. A balról érkező akadály esetén a robot eredetileg a teljes elérhető elkerülő sebességek térrészéből választhatott

sebességkomponenst, mint ahogy ezt a 6-8. ábra szemléltette, ám most a 6-15. ábra által bemutatott esetet vizsgálva a robot a felosztott elérhető elkerülő sebességek térrészeiből csak a zöld, illetve a kék színnel jelölt akadály mögött elhaladó sebességek illetve az akadályt elkerülő sebességek közül választhat sebességet magának. Sárga színnel látható a két térrész uniója.



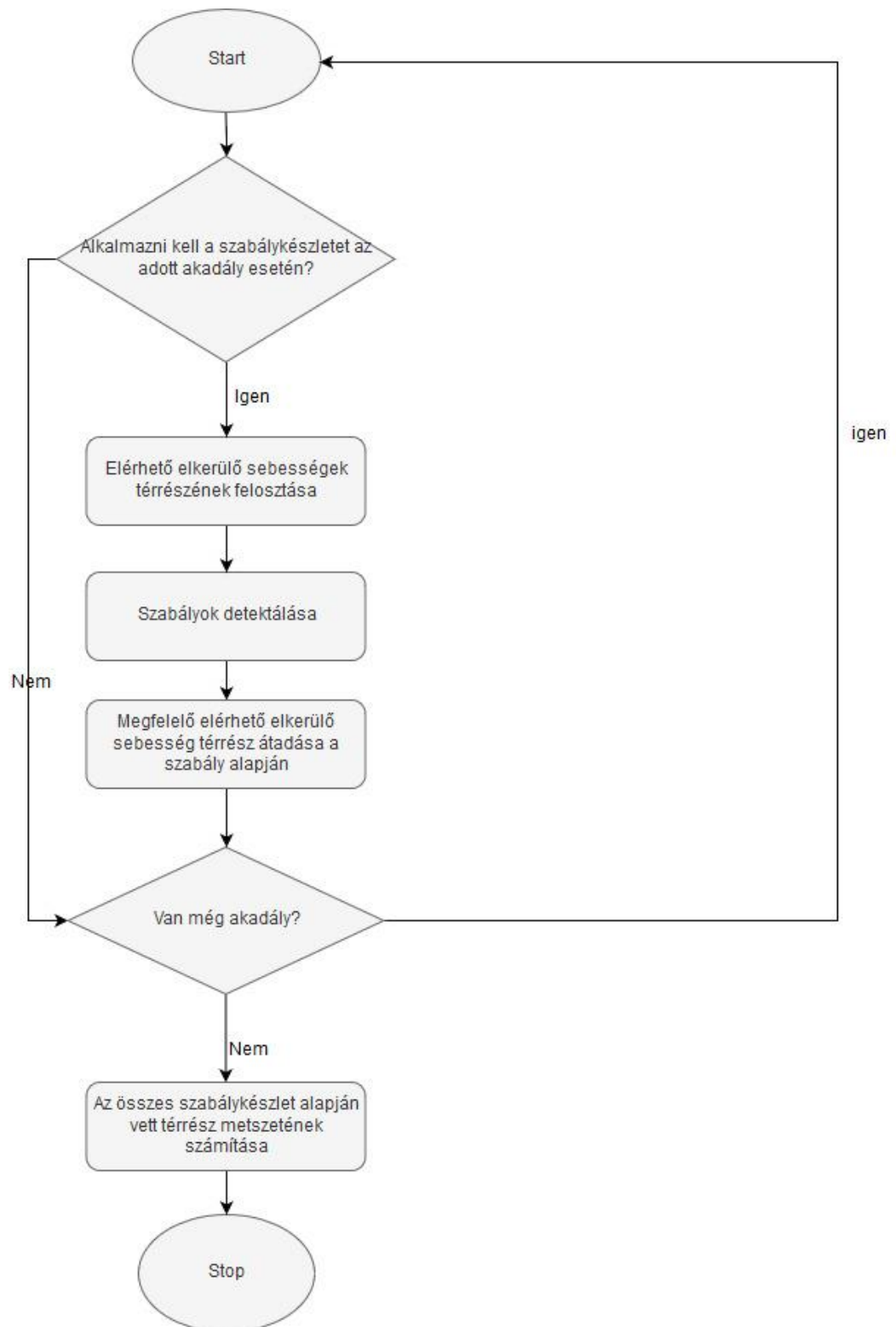
6-15. ábra: Elérhető elkerülő sebességek felosztása

Tehát amennyiben az akadály megkülönböztető jelzéssel rendelkezik, akkor a robotnak minden esetben el kell engednie. Erre az esetre mutat példát a következő videó: <https://youtu.be/Dlevbzk2dZA>.

6.3 Több akadály esetén történő szabály alapú pályatervezés

Amennyiben több akadály is jelen van a robot munkaterében, ki kell terjeszteni a szabályok alapján történő mozgástervezési algoritmust. Minden akadályra meg kell vizsgálni, hogy kell-e közlekedési szabályokat alkalmazni a (6.10) alapján. Minden olyan akadálnál, ahol a szabálykészlet alkalmazása szükségesnek bizonyul, ki kell számítani azokat az elérhető elkerülő sebesség térrészeket, melyekből a robot az adott szabály alapján sebességet választhat magának. Ezt követően az összes ilyen térrész metszetét kell venni, hiszen az összes közlekedési szabály betartását biztosítani kell a robotnak. Ebből a metszetből választhat sebességvektort önmaga számára a robot. A fejezetben bemutatott eseteknél sem az akadályok, sem a robot nem rendelkezik megkülönböztető jelzéssel.

A 3. fejezetben bemutatott mozgástervezési algoritmus Elérhető elkerülő sebességek meghatározásának alpontja tehát a következő folyamatábrával foglalható össze:



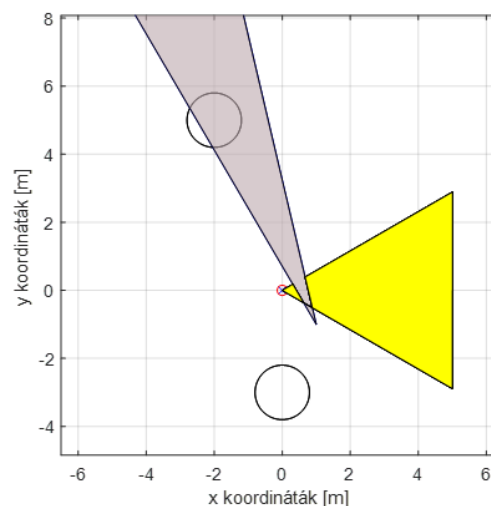
6-16. ábra: Elérhető elkerülő sebességek meghatározása több akadály esetén

Ezt követően 2 példán keresztül szemléltetném a több akadály esetén történő szabály alapú mozgástervezést:

1. Elsőként tekintsünk egy példát, amely esetben egy akadály jobbról, egy pedig balról érkezik.

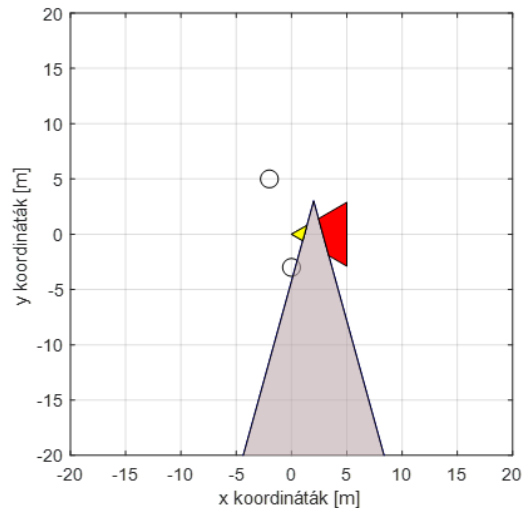
Ebben az esetben a mozgástervezés legelején mindkét akadályra érvényesül, hogy a szabályok alapján kell meghatároznunk azokat a sebesség térrészeket, melyekből a robot sebességet választhat.

A balról érkező akadály esetén a teljes elérhető elkerülő sebességek térrész átadásra kerül, ahogy azt a 6-17. ábra mutatja sárga színnel:



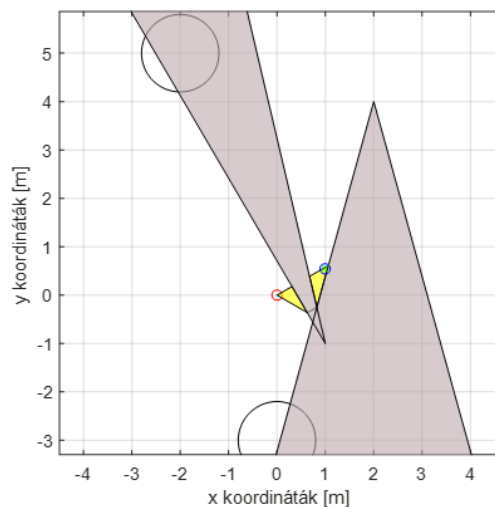
6-17. ábra: Balról érkező akadály esetén az elérhető elkerülő sebességek

A jobbról érkező akadály esetén a korábbi fejezetekben bemutatott eljárások alapján csak az akadály mögött elhaladó sebességek közül választhat a robot, amennyiben 2 térrészre bomlik az elérhető elkerülő sebességek térrésze. Ezt szemlélteti sárga színnel a 6-18. ábra.



6-18. ábra: Jobbról érkező akadály

Ezt követően a két szabály által meghatározott terület térrész metszetét kell venni, ebből választhat a robot sebességet a mozgásának tervezése során, amint azt a 6-19. ábra jelöli:



6-19. ábra: Szabályok alapján meghatározott területek metszete

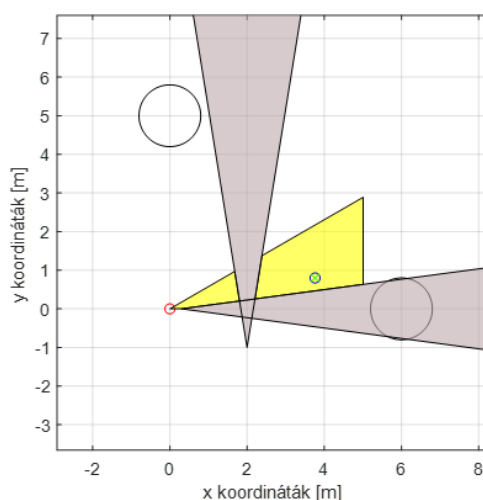
A robot ebben az esetben a bal akadályhoz képest akadály előtt elhaladó sebességet választ, a jobb akadályhoz képest pedig akadály mögött elhaladó sebességkomponenst, így biztosítva a KRESZ szabályok betartását a mozgása során. Ezt prezentálja a következő videó: <https://youtu.be/eJRdx9dm-Fk>.

2. Érdemes megvizsgálni egy olyan esetet is, amikor 3 akadály található a munkatérben, ám a 3. csak később közeledik a robothoz, ezáltal a

mozgástervezés kezdetén a szabályokat tekintve még nem kell figyelembe vennünk.

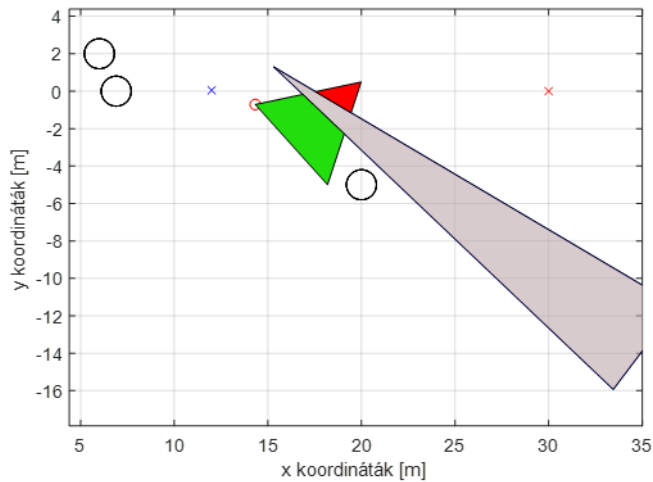
A munkatérben 3 akadály található: Egy balról közeledik, egy a robot előtt található és kis sebességgel a cél irányába halad, illetve egy jobbról közeledik de csak távolabbról.

Kezdetben tehát a balról érkező akadály, illetve a robot előtt található akadály VO térrészei alapján lehet meghatározni az elérhető elkerülő sebességek térrészét. A 6-20. ábra ábrázolja, hogy a két akadály által alkalmazott szabályokat alkalmazva mely sebesség térrészekből választhat a robot sebességet (sárga térrész). Fekete körök jelölik az akadályokat, szürke színű területek a VO térrészek, piros kör a robotot szemlélteti, zöld bekarikázott x jelöli a kiválasztott sebességkomponenst.



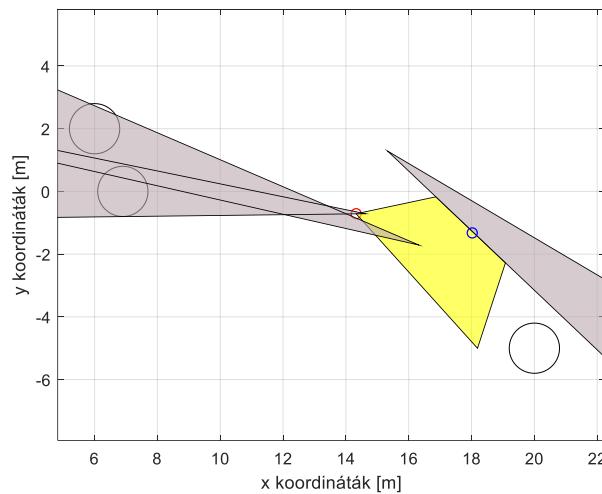
6-20. ábra: Balról érkező és robot előtt található akadály esetén történő sebességválasztás

Azután, hogy a robot elhaladt az előtte lassan haladó akadály mellett, illetve a balról érkező akadály előtt, ezzel a két akadállyal már szabályokat tekintve nem kell foglalkozni a mozgástervezés során. Ezt követően jelenik meg a jobbról érkező akadály, amelynél a korábbi fejezetekhez hűen a felosztott elérhető elkerülő sebességek közül csak azokat választhatja ki a robot, melyeket alkalmazva az akadály mögött halad el, ahogy ez a 6-21. ábra vetíti elénk.



6-21. ábra: Már csak a jobbról érkező akadály játszik szerepet a szabályválasztásban

Tehát hiába esik bele a másik két akadály VO térrészének a csúcsa az elérhető elkerülő sebességek térrészébe, mivel ezen akadályokat már elhagyta a robot, már nem játszanak szerepet a sebesség kiválasztásában, a robot ezek figyelmen kívül hagyásával választja ki a 6-22. ábra kék színű körével jelzett sebességét.



6-22. ábra: Sebességválasztás a már elhagyott akadályok figyelmen kívül hagyásával

A robot mozgása során tehát elengedi a jobbról érkező akadályt és így éri el a célját. Összességében tehát minden alkalommal csak az aktuális időpillanatban szabályok tekintetében szerepet játszó akadályok határozzák meg, hogy mely térrészekből választhat a robot sebességet. A 3 akadály mozgása itt tekinthető meg: <https://youtu.be/SaGNEcBnVzs>.

6.4 SVO és TRVO módszer párhuzamos használata a mozgástervezés során

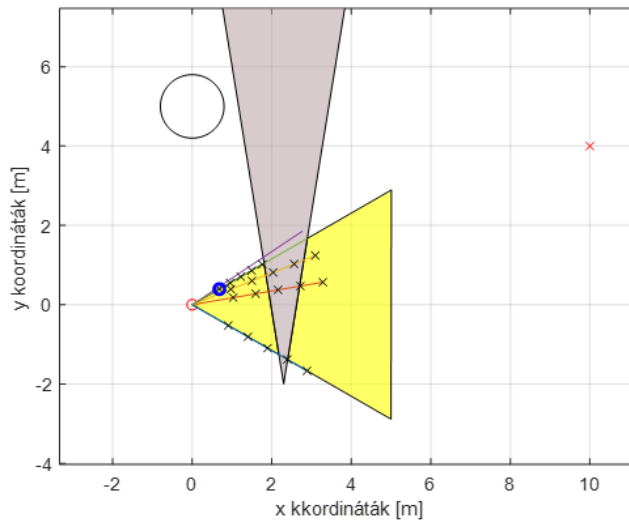
Az eddigi vizsgálat során minden esetben a leggyorsabb célélérés függvényében választott a robot sebességet magának. A szabály alapú mozgástervezés során figyelembe tudjuk venni (2.12) alapján egy költségfüggvény felírásával, hogy a mozgástervezés során a gyorsaság, vagy a biztonság játszon-e nagyobb szerepet. A legkisebb költségértékű sebességkomponenst fogja a robot választani.

A sebességkomponensek 5 irányban kerülnek vizsgálat alá: az elérhető sebességek határoló vonalainak irányában, a cél irányában, illetve a cél iránya körül pozitív és negatív tartományban egy adott szögtartományban. Ahogy közeledik a robot a célpozícióhoz, egyre csökken az a maximális sebességnagyság, amelyet kiválaszthat a robot (annak érdekében, hogy a célon ne fusson túl, meg tudjon állni). Egy adott irányban az aktuális időpillanatban maximális sebességnagyságtól kiindulva egy adott lépésközzel visszafelé lépve meg kell keresni az első elérhető elkerülő sebességkomponenst. Az elérhető elkerülő sebességek kezdőpontja és ezen legnagyobb sebességkomponens között egyenletes távolságban 5 pont kerül felvételre, ezekben a pontokban vizsgáljuk a költségfüggvény értékét. Ezt az algoritmust minden irányban végre kell hajtani. Ezáltal összesen maximum 25 pontban (ha valamelyik komponens ütközést okozna, nem kell azt a komponenst vizsgálni) történik meg a költségfüggvény kiértékelése. A robot sebességének a minimális költségértékkel rendelkező sebességkomponense kerül kiválasztásra. Ez teljes mértékben felfogható úgy, mintha egy ráccsal lefednénk az elérhető sebességek térrészét egy adott szögtartományban. Több vizsgált pont esetén a futási idő nagyon megnövekedne, és a mozgástervezés szempontjából nem kapnánk jobb eredményt, ezért született meg a korábban bemutatott módszer.

Amennyiben balról érkezik az akadály, a KRESZ szabályok alapján a teljes elérhető elkerülő sebességek térrészből választhat a robot sebességet. Tekintsük meg, hogy melyik sebességet választja a robot, amennyiben a (2.12)-ben bemutatott α aránytényező 0 illetve 1 értéket vesz fel.

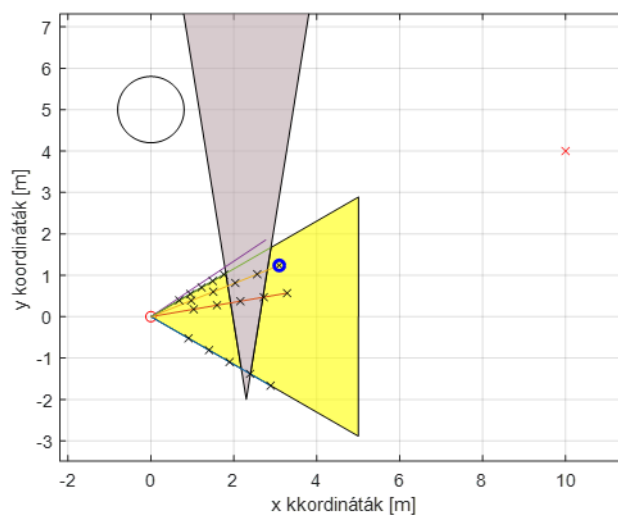
Amennyiben $\alpha = 0$ érték teljesül, akkor a legbiztonságosabb sebességkomponens kerül kiválasztásra, amely a VO térrésztől legtávolabb található, ahogy ezt a 6-23. ábra mutatja. Természetesen a VO térrészbe eső sebességkomponenseket nem választhatja a robot. Az ábrán piros kör jelzi a robotot (0,0) pozícióban, fekete kör az akadály, mely a

robothoz képest bal oldalról érkezik. Fekete x-ek jelölik a sebességkomponenseket, melyek a költségfüggvényben kiértékelésre kerülnek. Sárga színű terület mutatja az elérhető elkerülő sebességeket, piros x a célt (10,4)-ben. Kék kör jelzi a kiválasztott robotsebességet.



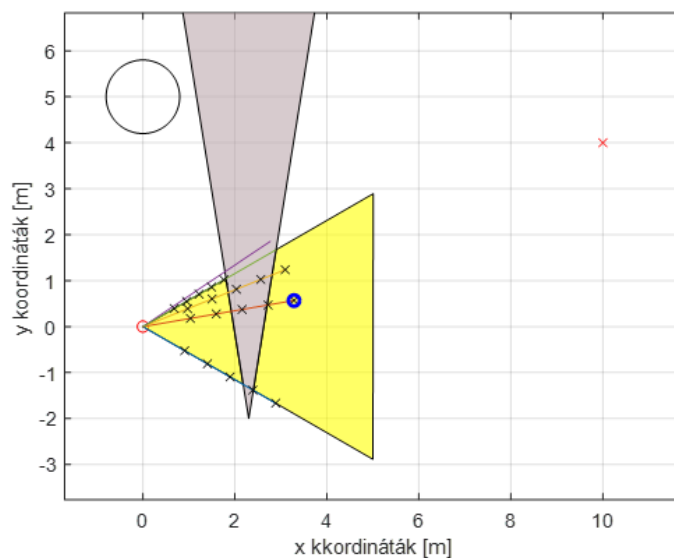
6-23. ábra: Sebességválasztás $\alpha = 0$ paraméter mellett

Amennyiben $\alpha = 1$ értékű, a leggyorsabb célelérést eredményező sebességkomponens kerül kiválasztásra, mellyel egy adott mintavételi időegység alatt a legközelebb érkezik a robot a célhoz. Ezt ábrázolja a 6-24. ábra. A jelölések megegyeznek az korábban bemutatott esettel. Látható, hogy a cél irányában a legnagyobb sebességet választja a robot.



6-24. ábra: Sebességválasztás $\alpha = 1$ paraméter mellett

A költségfüggvény által vizsgált sebességkomponensek közel találhatók egymáshoz, ezért biztonságot tekintve nincs olyan nagyon nagy különbség köztük (VO -tól való távolságok között nincs nagyságrendi különbség), ezért amennyiben $\alpha < 0.2$ akkor a 6-23. ábra által bemutatott legbiztonságosabb megoldást választja az algoritmus. Amennyiben teljesül, hogy $0.2 < \alpha < 0.7$, akkor a 6-25. ábra által szemléltetett sebességet választja a robot, mely nagyobb biztonsággal rendelkezik, mint a leggyorsabb sebességkomponens (távolabb található a VO -tól), de gyorsaságot tekintve közelebb kerülünk ezzel a sebességgel a célhoz, mint ha a VO másik oldaláról választana a robot sebességet.



6-25. ábra: Sebességválasztás köztes α aránytényező esetén

Tehát összefoglalva, a mozgástervezés során egyszerre figyelembe tudja venni az algoritmus a szabálykészlet alapján történő tervezést, illetve a költségfüggvény bevezetésével a gyorsaság és a biztonság különböző mértékű alkalmazását. Ezáltal az adott célfeladatra lehet optimalizálni a mozgástervezést.

7 Elérhető elkerülő sebesség térrészek számossága, számítási igény

A fejezetben elméleti kutatási eredményeimet mutatom be az elérhető elkerülő sebesség térrészek számosságát illetően. A 2.3.2. fejezet bemutatta, hogy N darab mozgó akadály esetén az elérhető elkerülő sebességek térrésze legfeljebb $3N$ részre osztható fel, mely Theorem 1-ként szerepel [21]-ben. Ezzel szemben én két Lemmát fogalmaztam meg:

- 1) **Lemma 1**: N darab mozgó akadály esetén az elérhető elkerülő sebességek térrésze nem igaz, hogy legfeljebb $3N$ részre osztható fel.
- 2) **Lemma 2**: N darab mozgó akadály esetén az elérhető elkerülő sebességek térrésze legfeljebb $(3^N - 1)$ részre osztható fel, amennyiben $N > 1$ érvényesül.

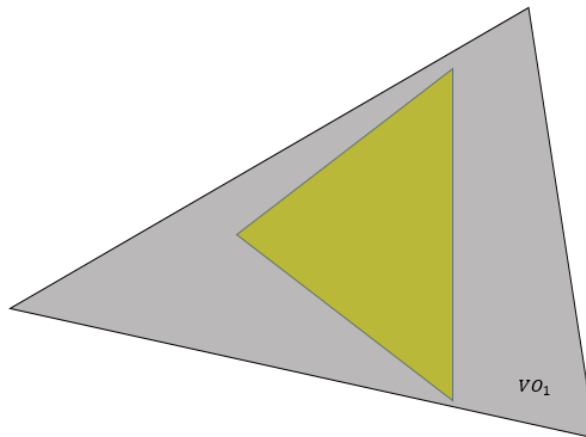
A lemmák bizonyításához tekintsük végig, hogy 1, illetve 2 mozgó akadály esetén, hogyan osztható fel az elérhető sebességek térrésze az akadály(ok)hoz tartozó $VO(k)$ figyelembevételével.

A területek elnevezéséhez a 2.3. fejezetben használt jelöléseket használom, az VO_i jelöli az adott akadályhoz tartozó sebességakadályt, ahol N akadály esetén $i \in [1; N]$. S_x jelöli, hogy az akadályhoz képest az adott elérhető elkerülő sebességterület milyen manővernek felel meg, ahol $x \in [r, f, d]$. Egy manőver esetén r jelöli, ha a hozzátartozó sebességet választva a robot az akadály mögött halad el, f esetén az akadály előtt megy el a robot, d esetén pedig elkerülő úton mozog. Több akadály esetén N db x van egymás mellett a területek felosztásánál, ahogy ezt a 2.3.2. fejezet is bemutatta. Az akadályok tetszőleges sorrendben definiálhatók, viszont a területek felosztásánál minden területnél figyelni kell arra, hogy az akadályok sorrendjében szerepeljenek a rájuk vonatkozó stratégiák.

7.1 1 akadály esetén az elérhető elkerülő sebesség területek felosztása

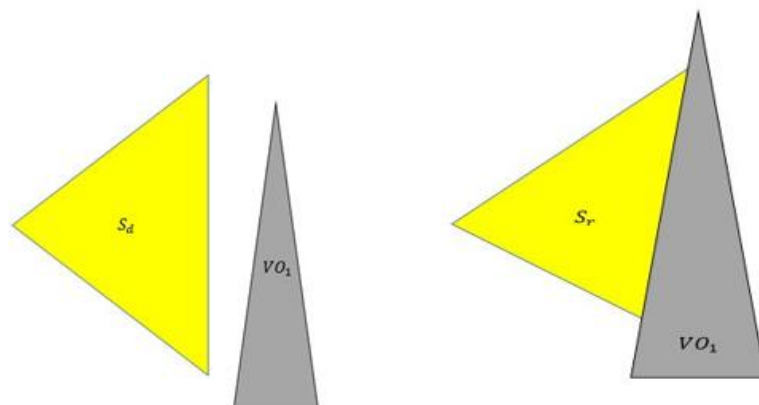
Egy akadály esetén az akadályhoz tartozó VO térrész a következőképpen tudja befolyásolni az elérhető elkerülő sebességek területfelosztását. A Theorem 1 alapján maximum 3 területre lehet felosztani az elérhető sebességek térrészét, ez 1 akadály esetén teljesül is, 0-3 között bármelyik eset előfordulhat.

a) 0 térrész



7-1. ábra: Az elérhető elkerülő sebességek térrésze 0 térrészre oszlik, hiszen a teljes VO lefedi az elérhető sebességek térrészét

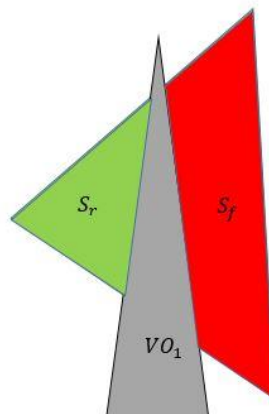
b) 1 térrész:



7-2. ábra: Az elérhető elkerülő sebességek térrésze 1 térrészre oszlik.

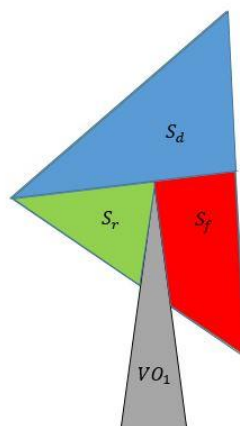
A 7-2. ábra bal oldalán található példa esetén a robot akadályhoz viszonyított mozgását tekintve kitérőnek számít az összes sárga területen található sebességkomponens, mivel a VO térrész egyáltalán nem érintkezik az elérhető elkerülő sebességek térrészével, ezáltal a robot mozgása során semmilyen kontaktusba nem kerül az akadállyal. A második esetben akadály mögött elhaladó sebességeket tartalmaz a sárgával jelölt terület térrész.

c) 2 térrész:



7-3. ábra: Az elérhető elkerülő sebességek térrésze 2 térrészre oszlik

d) 3 térrész:



7-4. ábra: Az elérhető elkerülő sebességek térrésze 3 térrészre oszlik

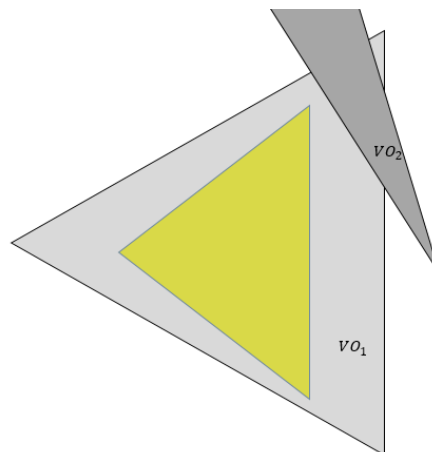
Egy akadály esetén tehát 1, 2, vagy 3 térrészre lehet osztani az elérhető elkerülő sebességek térrészét.

7.2 2 akadály esetén az elérhető elkerülő sebesség területek felosztása

Két akadály esetén minden felosztott területnél meg kell határoznunk, hogy a különböző akadályokhoz képest milyen manővert határoz meg. $S_{x1,x2}$ esetében mindig $x1$ határozza meg, hogy az 1. akadályhoz képest milyen manővert képvisel a térrész (ugyanígy a 2. index-szel is). A Theorem 1 alapján ebben az esetben maximálisan 6 térrészre lehetne osztani az elérhető sebességek térrészét, viszont látható, hogy ez nem igaz.

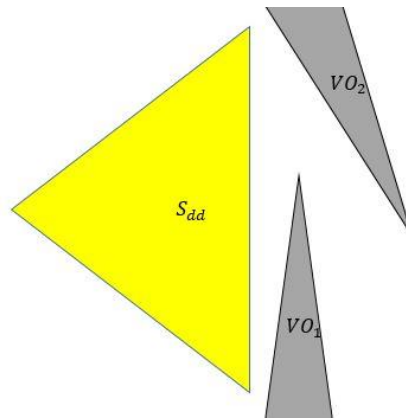
Egy-egy konstelláció többféleképpen előállítható a sebesség térrészek felosztását illetően, mindegyikre mutatok egy példát. 2 akadály a következő lehetséges elérhető elkerülő sebesség térrészek felosztását tudja eredményezni:

a) 0 térrész:



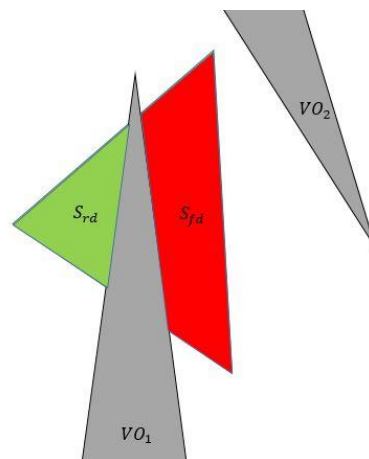
7-5. ábra: Az elérhető elkerülő sebességek térrésze 0 térrészre oszlik, hiszen az egyik VO térrész teljes mértékben fedi az elérhető sebességek térrészét

b) 1 térrész:



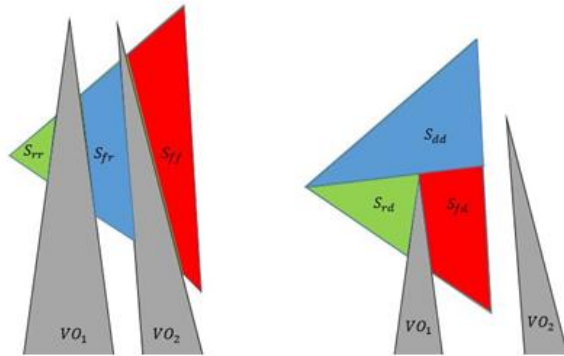
7-6. ábra: Az elérhető elkerülő sebességek térrésze 1 térrészre oszlik

c) 2 térrész:



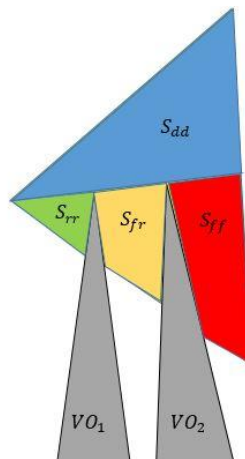
7-7. ábra: Az elérhető elkerülő sebességek térrésze 2 térrészre oszlik

d) 3. térrész:



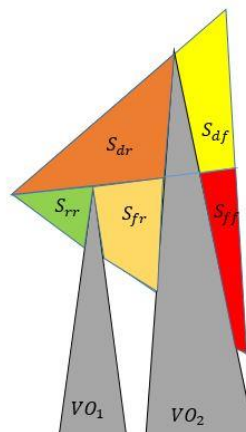
7-8. ábra: Az elérhető elkerülő sebességek térrésze 3 térrészre oszlik

e) 4 térrész:



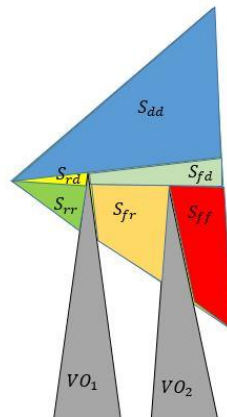
7-9. ábra: Az elérhető elkerülő sebességek térrésze 4 térrészre oszlik

f) 5 térrész:



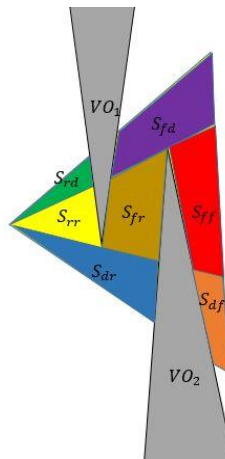
7-10. ábra: Az elérhető elkerülő sebességek térrésze 5 térrészre oszlik

g) 6 térrész:



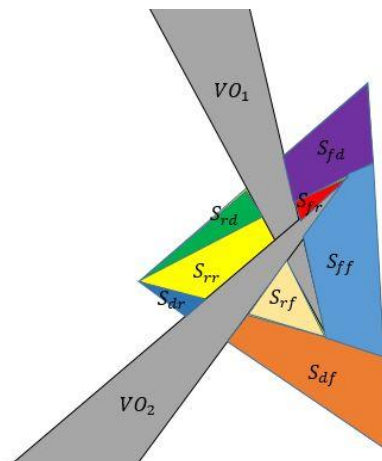
7-11. ábra: Az elérhető elkerülő sebességek térrésze 6 térrészre oszlik

h) 7 térrész:



7-12. ábra: Az elérhető elkerülő sebességek térrésze 7 térrészre oszlik

i) 8 térrész:



7-13. ábra: Az elérhető elkerülő sebességek térrésze 8 térrészre oszlik

Az 1. Lemma a h) és i) ábrákkal egyből igazolható, hiszen két akadály esetén több mint 6 térrészre oszlik az elérhető elkerülő sebességek térrésze.■

A 2. Lemma esetében amennyiben két különböző oldalról érkezik az akadály (VO térrészek ellentétes irányúak), abban az esetben mindig hiányzik az S_{dd} térrész, hiszen az lehetetlen eseménynek minősül, hogy az egyik és a másik VO csúcsot követő területről válasszunk sebességet, hiszen ellentétes a VO -k iránya. Ebben az esetben lehetett a legtöbb terület térrészt elérni, ahogy azt a i) ábra mutatta. (a számosság $3^N - 1 = 3^2 - 1 = 8$) volt. Amennyiben pedig egy irányúak a VO térrészek, akkor amelyik VO térrész magasabb (mint pl. g) példában), a „fölött” már biztosan csak S_{dd} térrész lesz, tehát kimarad mindenképpen az S_{df} és S_{dr} térrész (indexeknél most az első akadályhoz tartozó VO volt a magasabb, ez természetesen fordítva is igaz). Ezáltal igaz a 2. Lemma, miszerint N darab mozgó akadály esetén az elérhető elkerülő sebességek térrészére legfeljebb $(3^N - 1)$ részre osztható fel.■

7.3 Futásidő vizsgálata

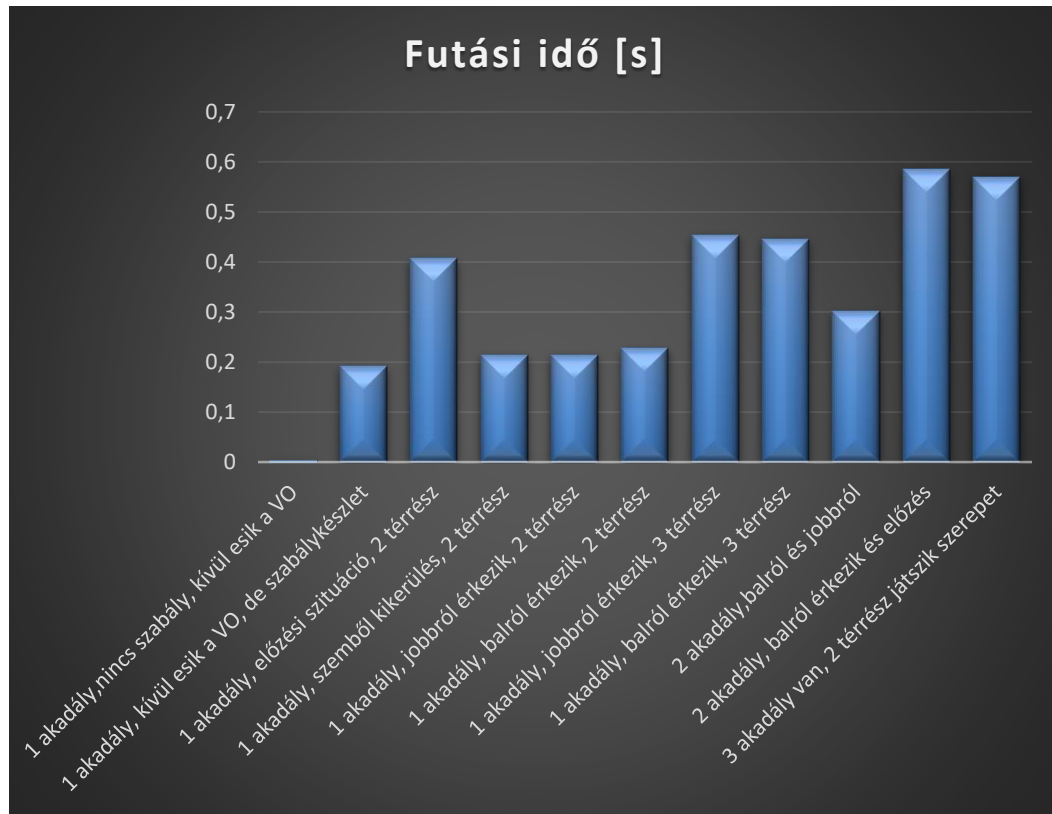
A 7.2. fejezetben bemutattam, hogy worst case esetben exponenciálisan nő N -nel a térrészek száma. Most különböző eseteket vizsgálva bemutatom, hogy a futásidő növekedése miként változik különböző akadályszám esetén a sebességterületek felosztásának függvényében.

A méréseket a laptopomon végeztem, mely a következő adatokkal rendelkezik:

Laptop	Dell Latitude E6430s
Processzor	Intel(R) Core(TM) i5-3320M CPU@ 2.60 GHZ
Memória (RAM) mérete	8 GB
Rendszer típusa	64 bites operációs rendszer, x64-alapú processzor

Fontos megjegyezni, hogy a futási idő minden egyes esetben az elérhető sebességek térrészenek kiszámítása során került feljegyzésre (az elkerülő sebességek térrészből történő robotsebesség kiválasztás nem került bele a vizsgálatba, hiszen ezt az akadályok száma nem befolyásolja).

Az eredményeket a 7-14. ábra mutatja be, minden mérés másodperc értékben került rögzítésre, 10-10 futtatás átlaga alapján:



7-14. ábra: Számítási igény az akadályok számának függvényében

Amennyiben 1 akadály esetén nem kell szabályt alkalmazni, akkor 0,003877 s a számítási igény, ezt a 7-14. ábra 1. oszlopa szinte 0-val szemlélteti. Ebben az esetben a teljes elérhető sebességek térrészből választhat a robot sebességvektort magának a mozgástervezés során. Amennyiben 1 akadály esetén 2 térrészre oszlik az elérhető elkerülő sebességek térrésze, akkor az előzési szituáció kivételével $\sim 0,2$ s a futási idő. Előzési helyzet esetén, illetve ha 3 térrészre oszlanak a sebességek, akkor a VO térrész csúcsa az elérhető sebességek térrészen belülre esik, ezzel megnövekszik a számítási igény, ebben az esetben $\sim 0,4$ s a futási idő. 2 akadály esetén amennyiben az egyik jobbról, a másik balról érkezik (mindkét esetben 2 térrészre osztódik az elérhető elkerülő sebességek térrésze), akkor a számítási igény 0,3 s-nak adódott, amely kevesebb, mint 1 akadály esetén ha 3 térrész adódott, bizonyítva ezzel, hogy a 3 térrész számítása sokkal bonyolultabb művelet, mint a 2 térrészé. 2 akadály esetén, amennyiben előzési szituáció is szerepet játszik a mozgástervezésnél, akkor egyből megnövekszik a számítási igény, hiszen a VO csúcsa ismét az elérhető sebességek térrészába esik bele, megnövelve a számítások komplexitását. Az eredmény ebben az esetben körülbelül 0,6 s ugyanúgy,

mint 3 akadály esetén, amikor csak 2 akadályt figyelembe véve kell a szabályok alapján kiszámítani az elérhető elkerülő sebességek területének térrészét, mivel a 3. akadály csak később kerül megfelelő közelségbe a robothoz képest. A táblázatban nem került több adat feljegyzésre, mivel a megfigyelések alapján egy adott időpillanatban maximálisan 2 olyan akadály található, amelyre a (6.10) teljesül, rájuk vonatkozóan a szabályok alapján kell meghatároznunk az elérhető elkerülő sebességek térrészét. Ezáltal több akadály esetén, amennyiben az adott akadályt tekintve nem kell szabályok alapján tervezni a lehetségesen kiválasztható elérhető elkerülő sebességeket, akkor ezek az akadályok nem befolyásolják a szabályok alapján történő tervezés végeredményét, a futási időt csak infinitezimális mértékben növelik.

Tehát látható, hogy amennyiben a VO csúcsa az elérhető sebességek térrészen belülre esik, ez eredményezi a legtöbb sebesség térrészre történő felosztást, ezáltal a számítások bonyolultságának függvényében növekedett a futásidő is. Amennyiben mindig két térrészre osztja a VO az elérhető sebességek térrészét, abban az esetben a futásidőben lineáris növekedés volt tapasztalható az akadályok számának növekedésével.

8 Értékelés, továbbfejlesztési lehetőségek

Munkám során egy mozgástervező algoritmust fejlesztettem ki és implementáltam, mely alkalmas mobilis robotok és járművek mozgástervezésére dinamikus környezetben. Ehhez a mozgástervező algoritmushoz alapvetően a *Velocity Obstacles* módszert használtam fel, mely egyszerre térképezi fel a változó környezetet a robot sebességterével, és így határozza meg a pálya geometriáját a sebességprofillal együtt.

A KRESZ szabályok alapján történő mozgástervezés érdekében kifejlesztettem és implementáltam a *Traffic Regulation Velocity Obstacles (TRVO)* módszert. A megoldás közlekedési táblák nélküli közlekedési helyzetekre biztosít a KRESZ szabályok figyelembevételével ütközésmentes mozgást a robot számára.

Külön kifejlesztésre került, hogy megkülönböztető jelzést használó jármű, vagy akadályok esetén hogyan változik a mozgástervező algoritmus. A munkám során felhasználtam az általam korábban kifejlesztett *Safety Velocity Obstacles (SVO)* módszert is. Így a szabálykészlet alapján történő tervezés mellett a gyorsaság, illetve biztonság is szerepet tud játszani a mozgástervezés során.

TDK dolgozatomban két Lemmát is megfogalmaztam az elérhető elkerülő sebességek területének felosztásának számosságát illetően, illetve a futási idő nagysága is vizsgálat alá került.

A későbbiekben továbbfejlesztésként figyelembe lehetne venni, hogy amennyiben adott időben több jármű is megkülönböztető jelzéssel rendelkezik, akkor milyen preferenciák alapján történjen a tervezés. A *TRVO* kiterjeszthető lenne közlekedési táblák és lámpák esetére is. Az algoritmust tovább lehetne fejleszteni téglalap alakú akadályok és robot esetére is, közelítve, hogy autószerű járművekről beszélhessünk. Több robot esetére is ki lehetne terjeszteni a módszert.

Ez a mozgástervezési algoritmus a jövőben nagy lehetőségként szolgálhat az önvezető autók mozgástervezése esetében, illetve segítséget nyújthat a városi forgalom irányításában is.

Köszönetnyilvánítás

Hálás vagyok Istennek, hogy éltet, erőt ad és kegyelmének irgalmassága szerint vezet az úton.

Hálás köszönettel tartozom Gincsainé Dr. Szádeczky-Kardoss Emesének, konzulensemnek, aki minden alkalommal idejét és energiáját rám áldozva tanácsaival ellátott, mindig támogatott, inspiráló közeget biztosított az előrehaladás érdekében.

Köszönettel tartozom szüleimnek is, akik szüntelenül biztosítják a lehetőséget a tanulásra, fejlődésre. Szeretetükkel és gondoskodásukkal mindig támogatnak, amelyért hálás a szívem.

Irodalomjegyzék

- [1] Steven M. LaValle: Planning Algorithms, Cambridge University Press, (2006).
- [2] Canny, J. and Reif, J.: New lower bound techniques for robot motion planning problems, In 28th IEEE Symposium on Foundation of Computer Science, (1987).
- [3] Reif, J. and Sharir, M.: Motion planning in the presence of moving obstacles, In 25th IEEE Symposium on Foundation of Computer Science, pp 144-153., (1985).
- [4] Fujimura, K. and Samet, H.: Time-minimal path among moving obstacles, In IEEE International Conference on Robotics and Automation, pp1110-1115, Scottsdale, AZ, (1989).
- [5] Kant, K. and Zucker, S: Towards efficient trajectory planning the path-velocity decomposition, The International Journal of Robotic Research, 5(3):72-89, (1986).
- [6] Lee, B and Lee, C.: Collision-free motion planning of two robots. IEEE Transactions on System Man and Cybernetics, SMC-17(1):pp 21-32, (1987).
- [7] Szádeczky-Kardoss, E. and Kiss, B.: Motion planning in dynamic environments with the Rapidly exploring Random Tree method. Int. Review of Automatic Control, 1(1),pp 109–117, (2008).
- [8] Adam Bry, Nicholas Roy: Rapidly-exploring Random Belief Trees for Motion Planning Under Uncertainty, 2011 IEEE International Conference on Robotics and Automation Shanghai, (2011)
- [9] Matt Zucker, James Kuffner, Michael Branicky: Multiparticle RRTs for Rapid Replanning in Dynamic Environments, 2007 IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April (2007)
- [10] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock: Randomized kinodynamic motion planning with moving obstacles, Int. J. Robot. Res., vol. 21, . 3, pp. 233–255, Mar., (2002).
- [11] J. J. Kuffner Jr and S. M. LaValle: RRT-connect: An efficient approach to single-query path planning, in Proc. IEEE Int. Conf. Robot. Autom., San Francisco, CA, Apr. 2000, vol. 2, pp. 995–1001, (2000).
- [12] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars: Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 566–580, Aug. (1996)
- [13] D. Fox, W. Burgard, and S. Thrun: The dynamic window approach to collision avoidance, IEEE Robot. Autom. Mag., vol. 4, no. 1, pp. 23–33, Mar. (1997).

- [14] Seder, M. and Petrovic, I.: Dynamic window based approach to mobile robot motion control in the presence of moving obstacles, In Proc. of Int. Conf. on Robotics and Automation, pp. 1986–1991, (2007).
- [15] S. Petti and T. Fraichard: Safe motion planning in dynamic environments, in Proc. IEEE RSJ Int. Conf. Intell. Robot. Syst., Edmonton, AB, Canada, pp. 2210–2215, Aug. (2005).
- [16] Fraichard, T. and Asama, H.: Inevitable collision states. A step towards safer robots?, *Advanced Robotics*, 18(10), pp. 1001–1024, (2004).
- [17] Martinez-Gomez, L. and Fraichard, T: Collision avoidance in dynamic environments: an ICS-based solution and its comparative evaluation, in Proc. of IEEE Int. Conf. on Robotics and Automation, (2009).
- [18] Qidan Zhu, Yongjie Yan, and Zhuoyi Xing: Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing, *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)* 0-7695-2528-8/06 IEEE, (2006).
- [19] Cao Qixin, Huang Yanwen, Zhou Jingliang: An Evolutionary Artificial Potential Field Algorithm for Dynamic Path Planning of Mobile Robot, *Proceedings of the 2006 IEEE/RSJ, International Conference on Intelligent Robots and Systems*, October 9 - 15, (2006).
- [20] Muhannad Mujahed *, Dirk Fischer, Bärbel Mertsching: Admissible gap navigation: A new collision avoidance approach, *Robotics and Autonomous Systems* 103 93–110, (2018).
- [21] Paolo Fiorini and Zvi Shiller: Motion Planning in Dynamic Environments using Velocity Obstacles, *The International Journal of Robotics Research* July vol. 17 no. 7 760-772., (1998).
- [22] Zoltán Gyenes and Emese Gincsiné Szádeczky-Kardoss: Motion planning for mobile robots using the Safety Velocity Obstacles method, *Proceedings of the 2018 19th International Carpathian Control Conference* (389-394). Szilvásvárad, Magyarország, (2018).
- [23] Yoshiaki Kuwata, Michael T. Wolf, Dimitri Zarzhitsky, and Terrance L. Huntsberger: Safe Maritime Autonomous Navigation with colregs, using Velocity Obstacles, *IEEE Journal of Oceanic engineering*, vol. 39, no. 1, (2014).
- [24] Gyenes Zoltán: Mobilis robotok pályatervezése dinamikus környezetben, *Budapesti Műszaki és Gazdaságtudományi Egyetem, BSc Szakdolgozat* (2016).
- [25] Frederic L. Large, Sepanta Sekhavat, Zvi Shiller and Christian Laughier: Using Non-Linear Velocity Obstacles to Plan Motions in a Dynamic Environment, *Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV'02)*, Dec (2002).

Függelék

1. Szabály alapú pályatervezés, a robot előtt van az akadály, megelőzi:
<https://youtu.be/4Q27cNcPJ7I>.
2. Szabály alapú pályatervezés, szemből érkezik az akadály <https://youtu.be/-2Wsvlcxslo>
3. Szabály alapú pályatervezés, jobbról érkezik az akadály:
<https://youtu.be/bmAqTccjXVU>
4. Szabály alapú pályatervezés, balról érkezik az akadály:
<https://youtu.be/wBKSq1TtcTs>
5. Jobbról érkező akadály esetén a mozgás, megkülönböztető jelzés nélkül:
<https://youtu.be/ItvAXCVHvdA>.
6. Megkülönböztető jelzést használó robot mozgása: <https://youtu.be/z-kjEQWIQRI>
7. Megkülönböztető jelzést használó akadály esetén történő pályatervezés:
<https://youtu.be/Dlevbzk2dZAT>
8. 2 akadály esetén történő szabály alapú mozgástervezés:
<https://youtu.be/eJRdx9dm-Fk>
9. 3 akadály esetén történő szabály alapú mozgástervezés:
<https://youtu.be/SaGNEcBnVzs>