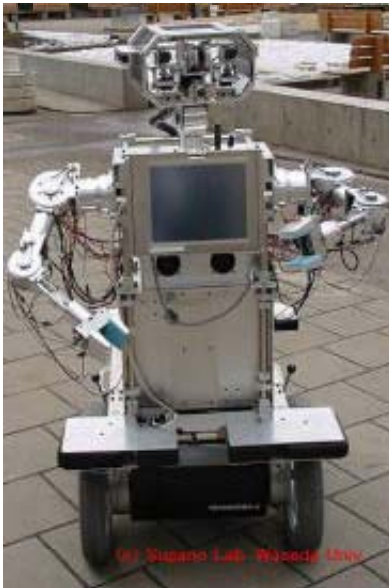


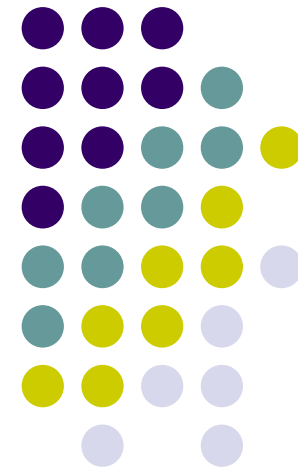
# Lekcija 12: *Planiranje kretanja mobilnog robota*



Prof.dr.sc. Jasmin Velagić  
Elektrotehnički fakultet Sarajevo

Kolegij: Mobilna robotika

2012/2013





## 12.1. Uvod

- Jedan od ključnih problema u mobilnoj robotici jest **sposobnost generiranja plana vlastitog kretanja s ciljem izvršavanja nekog zadatka.**
- Cilj planiranja kretanja je promjena stanja okoline na temelju računanja sekvence dopustivih kretanja robota.
- Tako se kod izvršavanja određenog kretanja zahtijeva dolazak robota iz početne u ciljnu tačku, a da se pri tome izbjegavaju sve prepreke i objekti koje se mogu naći na putu.
- **Proces kojim se određuje skup slobodnih putanja kretanja (engl. collision free paths) robota do željenog cilja naziva se *planiranje kretanja.***

# Uvod

- Idealno robot postiže svoje ciljeve uprkos mnogim mogućim ograničenjima na kretanje, koja mogu biti **unutarnja** ili **vanjska**.
- Fizičke prepreke u radnoj okolini predstavljaju vanjska ograničenja na kretanje mobilnog robota i jedina stvarna ograničenja koja se razmatraju u procesu planiranja kretanja.
- Unutarnja ograničenja uključuju prirodna mehanička ograničenja, kao što su neholonomna ograničenja koja onemogućavaju bočno kretanja mobilnog robota, i senzorska ograničenja, kao što je blokiranje pogleda kamera od strane prepreka.
- Ova unutarnja ograničenja su važna i moraju se uzeti u obzir u planiranju budućih kretanja.



# Uvod

- Postoje dvije fundamentalne paradigme za planiranje putanje s izbjegavanjem prepreka:
- **Globalni pristup** koji generira putanju off-line korištenjem poznatih informacija o okolini.
  - On zahtijeva precizno specificiran model okoline i znatan iznos vremena računanja.
  - Ovim pristupom nije moguće reagirati na nepredviđene promjene u okolini.
- **Lokalni pristup** se obavlja on-line korištenjem malog broja poznatih informacija o prostoru.
  - Moguće je reagirati na nepredviđene promjene i ne zahtijeva se velik iznos računarskog vremena.
  - Na ovaj način nije moguće dobiti optimalnu putanju.

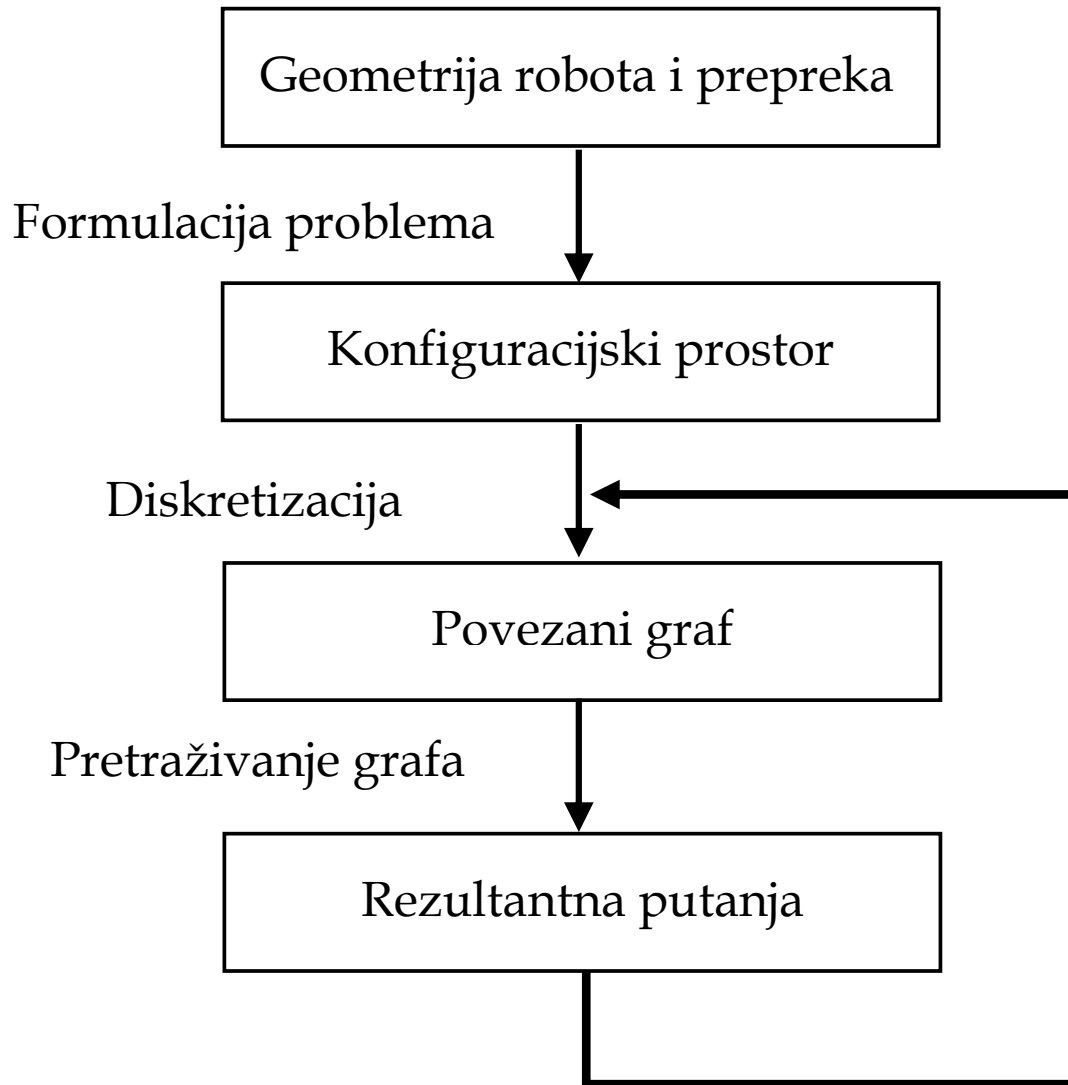


# Uvod

- Prvi korak je **preslikavanje (mapiranje) robota**, koji ima složen geometrijski oblik, u *tačku* u novom, apstraktnom prostoru, koji se naziva **konfiguracijski prostor** [Lozano-Perez i Wesley, 1989].
- Ovo preslikavanje transformira originalni problem u problem planiranja putanje pokretne tačke.
- Nakon toga se **diskretizira kontinuirani konfiguracijski prostor** i **konstruira povezani graf**.
- Na kraju se **pretražuje graf** kako bi se pronašla putanja kojom bi robot dosegao svoj cilj.



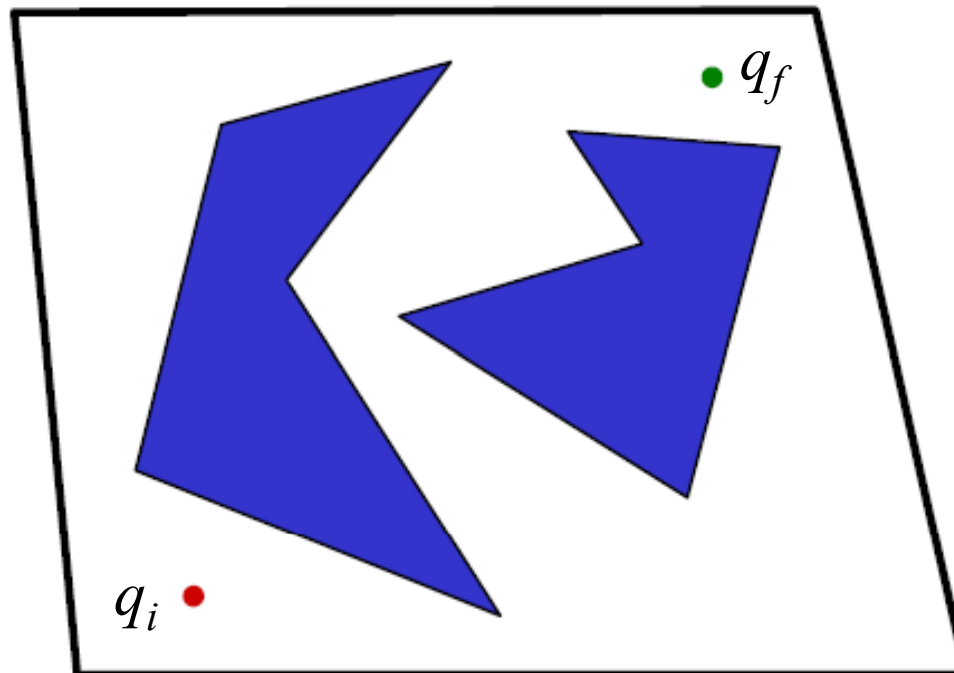
# Uvod



- Ako se putanja ne pronade, proces pretrage se ponavlja redefiniranjem procesa diskretizacije i pretraživanja putanje.
- Kompletni, navedeni proces, planiranja putanje je prikazan na slici.

# Uvod

- Na temelju navedenog može se reći da su ulazi u proces planiranja: **geometrijski opis robota i okruženja** i **početni i krajnji položaj robota** (pozicija i orijentacija).
- Izlaz procesa planiranja putanje je **putanja** od početnog  $q_i$  do ciljnog položaja  $q_f$



# Uvod

- Zahtjevi pri procesu planiranja trajektorije mogu biti:
  - **Minimalna duljina staze.**
  - **Sredina s pokretnim preprekama, što predstavlja dodatni izazov.**
  - **Sredina sa pokretnim ciljem.**
  - **Minimalan iznos vremena za prijelaz putanje.**
  - **Utrošak energije.**
  - **Izbor staza koje zadovoljavaju neka druga ograničenja.**
- Planirnje kretanja zahtijeva razmatranje svojstava robota i strukture okoline.





## Uvod

- Okolina, odnosno prostor u kome se kreće mobilni robot sastoji se od prepreka i slobodnog prostora.
- Prepreke predstavljaju već zauzeti dio prostora, odnosno dio prostora kroz koji se robot ne može kretati.
- Slobodni prostor je nezauzeti dio prostora unutar koga se robot može kretati.
- Za određivanje kuda se mobilni robot može kretati definira se konfiguracijski prostor.
- **Konfiguracija** robota je skup parametara koji jednoznačno određuju poziciju svake tačke na robotu.



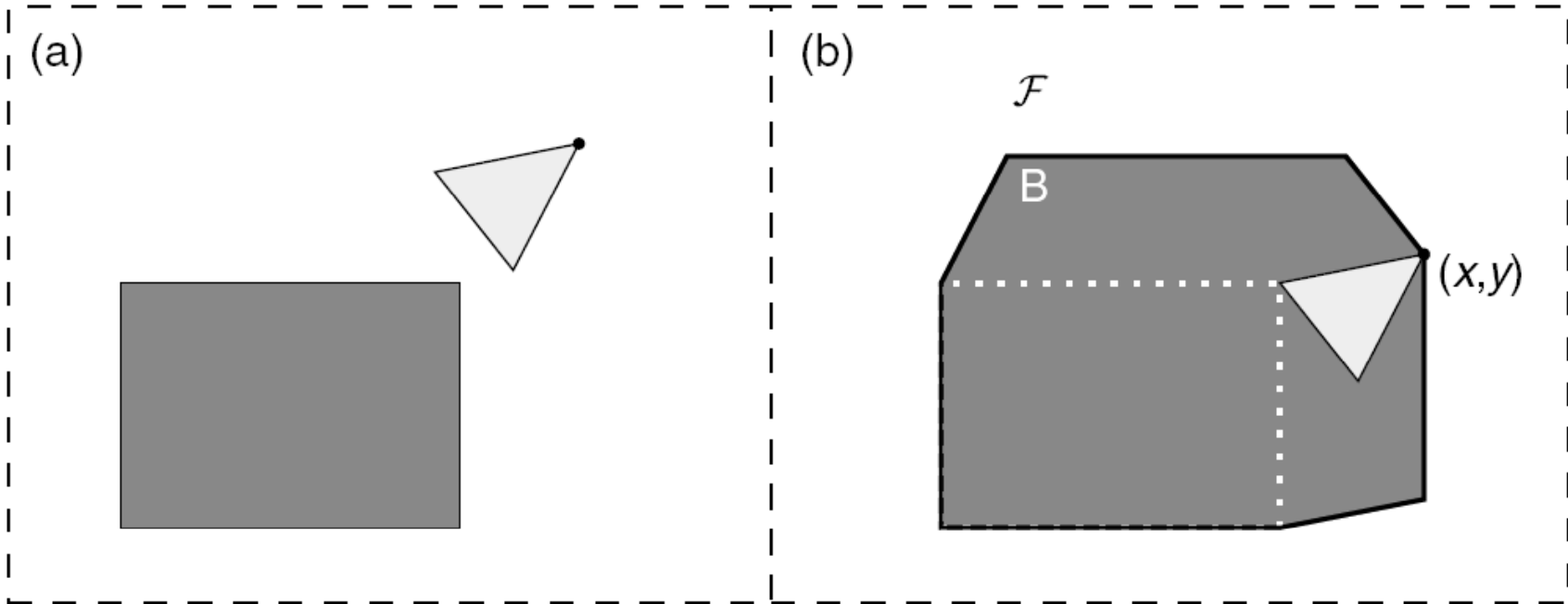
# Uvod

- Na primjer, konfiguracija mobilnog robota je zadana njegovom pozicijom  $(x,y)$  i orijentacijom  $\theta$  za  $\theta \in [-\pi, \pi]$  i predstavlja se trojkom  $\mathbf{q} = (x, y, \theta)$ .
- Ako se pretpostavi da se konfiguracija robota sastoji od  $d$  parametara, tada se ona može promatrati kao tačka u  $d$ -dimenzionalnom prostoru  $C$ , nazvanom **konfiguracijski prostor**.
- **Konfiguracijski prostor je skup svih mogućih položaja robota.**
- Konfiguracija  $q$  je slobodna ako je robot postavljen u  $q$  i ne dodiruje prepreke u prostoru.



# Uvod

- **Slobodni prostor  $F$**  predstavlja podskup svih slobodnih konfiguracija robota u  $C$ -u.
- Prostor prepreka  $B$  je komplement od  $F : B = C \setminus F$ .
- Za ilustraciju slobodnog, zauzetog i konfiguracijskog prostora promatra slika ispod.



# Uvod

- Za robota koji se samo translacija u ravnini može se eksplicitno konstruirati prostor  $C$  računanjem Minkowski razlike robota i prepreka.
- Intuitivno, može se razmišljati o računanju kao "širenju" prepreka oblikovanjem robota i smanjivanjem robota na veličinu tačke.
- Obično robot osim translacije obavlja i rotacijsko kretanje.
- U slučaju rotacije, računaju se isječci prostora  $C$  sa robotom u različitim fiksnim orijentacijama i nakon toga se oni gomilaju i slažu.
- Egzaktno računanje  $C$  je također moguće, s tim da je znatno kompliciranije (Avnaim i Boissonnat, 1988).



## 12.2. ROADMAP algoritmi planiranja

- Roadmap algoritmi planiranja putanje spadaju u globalno planiranje kretanja.
- Ovi algoritmi **formiraju slobodni robotski prostor u obliku mreže jednodimenzionalnih krivulja ili linija, koje se zovu mape puteva** (road mape).
- Cilj je reducirati  $N$ -dimenzionalni konfiguracijski prostor u skup jednodimenzionalnih putanja za pretraživanje.
- Kada se konstruira graf putanja tada se koristi algoritam planiranja kretanja kojim se pretražuje niz putanja od početne  $q_i$  do ciljne  $q_f$  robotske konfiguracije.



# ROADMAP algoritmi planiranja

- Mapa puta predstavlja dekompoziciju konfiguracijskog prostora mobilnog robota na temelju specificiranja geometrije prepreka.
- Cilj je konstruirati skup putanja koje omogućuju robotu kretanje bilo gdje unutar slobodnog prostora uz istovremeno minimiziranje ukupnog broja putanja.
- Svojstva roadmap algoritama su:
  - **Pristupnost:** postoji slobodna putanja od startne pozicije robota do mape putanja.
  - **Odstupnost:** postoji slobodna putanja od mape putanje do ciljne pozicije robota.
  - **Povezanost:** postoji slobodna putanja od uzlazne do silazne tačke na mapi putanja.

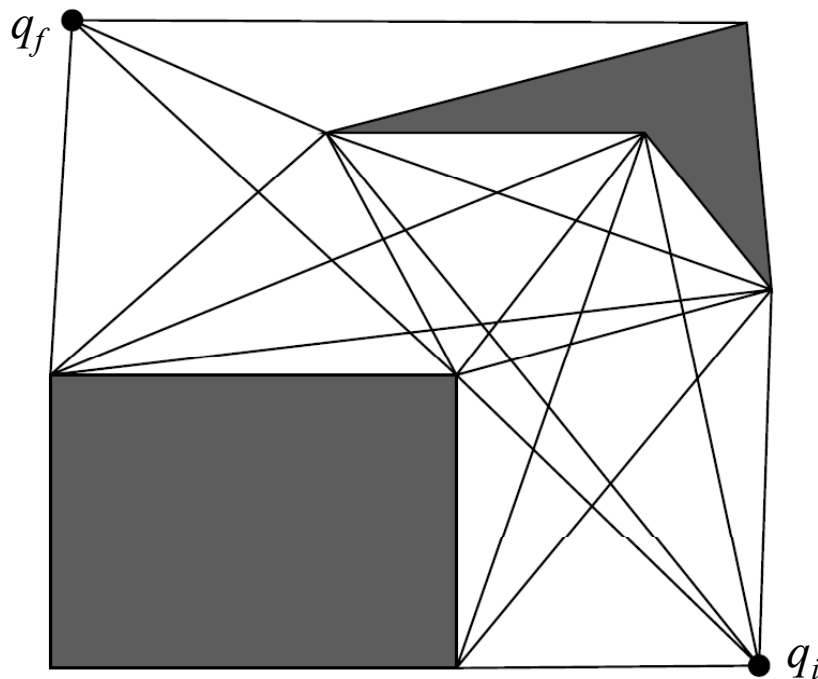


## 12.2.1. Graf vidljivih vrhova

- Graf vidljivih vrhova  $G_v$  se primjenjuje u dvodimenzionalnom konfiguracijskom prostoru sa poligonalnim preprekama.
- **Oblikuje se međusobnim spajanjem svih vidljivih vrhova prepreka  $C$  i početne  $q_i$  i ciljne konfiguracije  $q_f$  robota.**
- Dvije tačke, odnosno vrha, su vidljive ako između njih nema prepreka.
- Također postoje putanje oko rubova prepreka ako su dva susjedna vrha iste prepreke međusobno vidljiva.
- Ako je dobiveni graf putanja unutar mape prostora, tada je svaka spojna linija grafa potencijalni dio putanje mobilnog robota od početne do ciljne pozicije.

# Graf vidljivih vrhova

- Korišćenje grafa vidljivih vrhova:
  - $K_1$ : nacrtaju se spojne linije od početne i ciljne tačke do svih vidljivih vrhova i do uglova prostora.
  - $K_2$ : nacrtaju se spojne linije između svih vidljivih vrhova svih prepreka.
  - $K_3$ : linije uzduž rubova prepreka su također spojne linije.
  - $K_4$ : ponavljati  $K_2$  i  $K_3$  dok se svi vrhovi ne spoje.





## Graf vidljivih vrhova

- Postoje dvije važne poteškoće u primjeni grafova vidljivih vrhova.
- Prva se odnosi na **veličinu prostora i broja vrhova i spojnih linija između njih.**
- Sa povećanjem broja poligonalnih prepreka povećava se veličina konfiguracijskog prostora i broj vrhova i spojnih linija.
- Iako se ovaj algoritam veoma brzo izvršava, on može biti sporiji i neefikasniji u odnosu na druge algoritme ako je konfiguracijski prostor značajno popunjen preprekama.
- Drugi nedostatak je da **rezultantne putanje imaju tendenciju navoditi robota što je moguće bliže preprekama** tokom njegovog kretanja ka cilju.



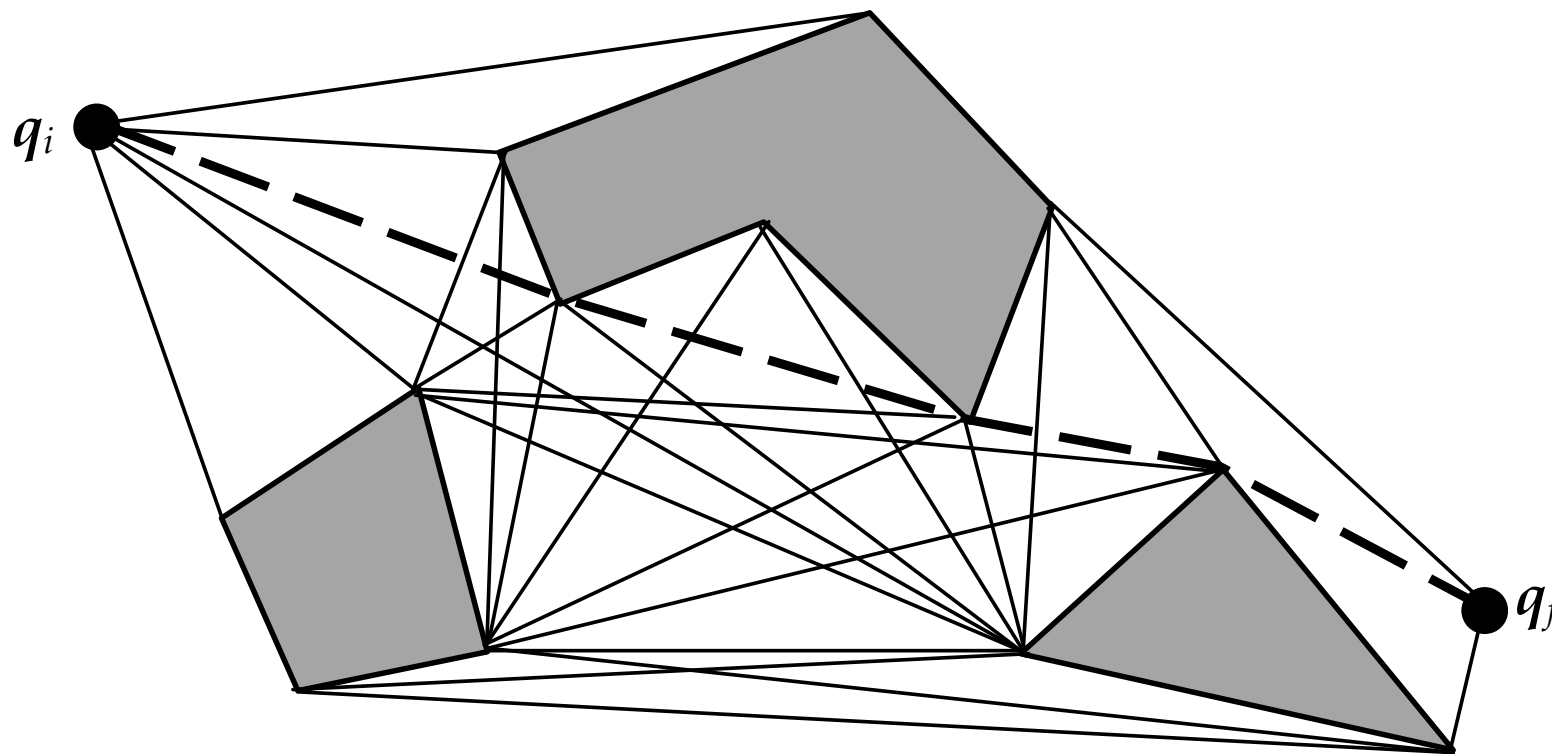
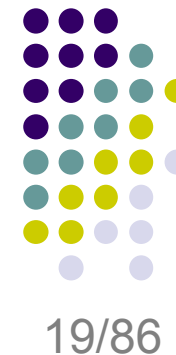
# Graf vidljivih vrhova

- Ovaj metod je ekstremno brz i efikasan u razrijeđenim okolinama, odnosno okolinama sa preprekama između kojih postoje veća rastojanja.
- Planiranje pomoću grafova vidljivih vrhova se može učiniti **optimalnim** u pogledu duljine rezultatne putanje.
- Za optimalnost je usko vezan pojam **sigurnosti**, u smislu držanja robota na razumnoj udaljenosti od prepreka.
- Rješenje koje bi zadovoljilo oba kriterija, optimalnost i sigurnost, sastoji se u povećanju dimenzija prepreka, radije nego u povećanju polumjera robota, ili alternativno, modificirati rezultatnu putanju nakon planiranja putanje na temelju udaljenosti između putanje kretanja robota i prepreka.



# Graf vidljivih vrhova

- Pronalaženja optimalne putanje korištenjem grafa vidljivih vrhova od početne  $q_i$  do ciljine  $q_f$  konfiguracije mobilnog robota.



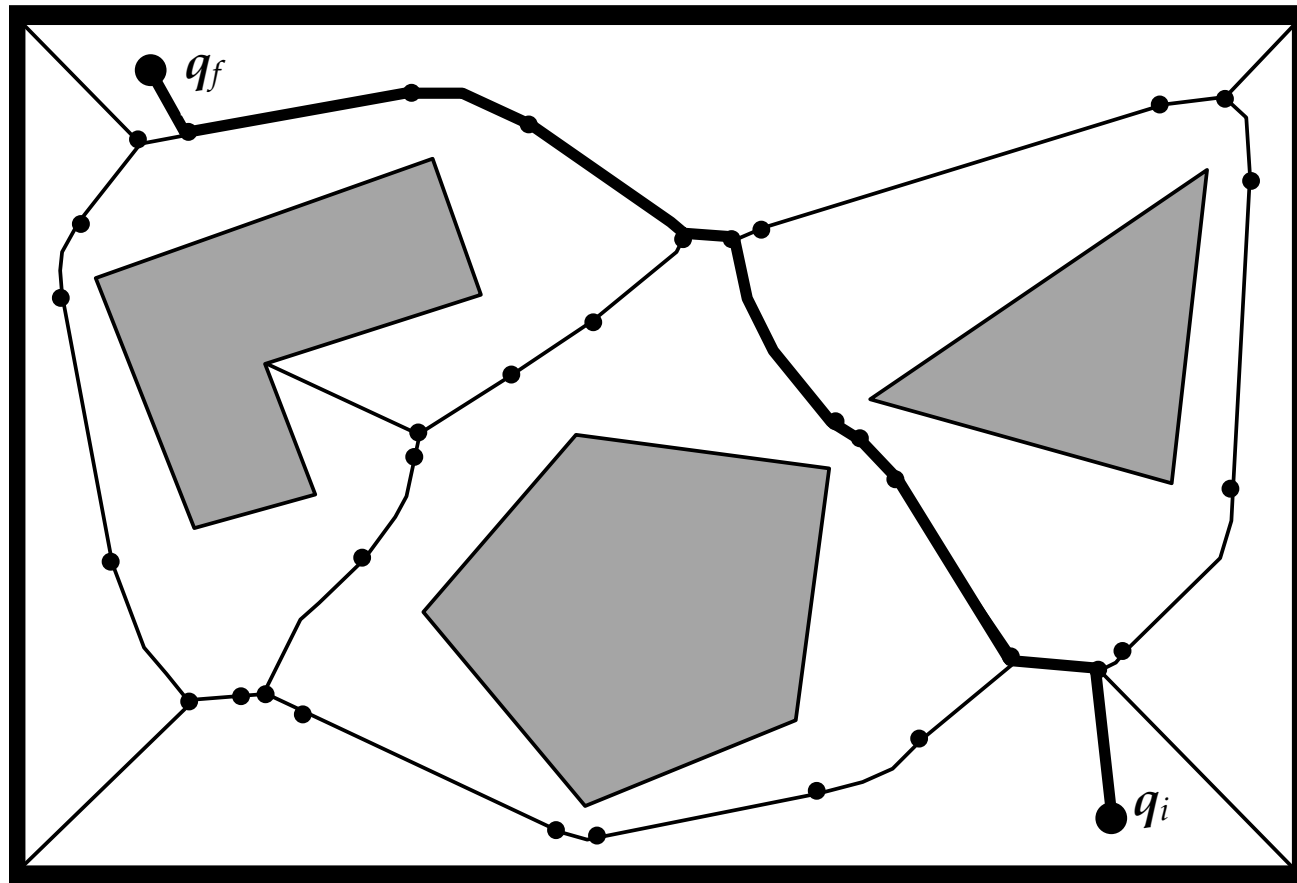
## 12.2.2. Voronoijev dijagram

- Za razliku od grafa vidljivih vrhova, **Voronoijev dijagram formira vrlo sigurnu mapu putanja koje su udaljene od prepreka koliko je maksimalno moguće** (Latombe, 1991).
- U ovom potpoglavlju se razmatra **generalizirani Voronoijev dijagram**.
- **Ovaj dijagram predstavlja mrežu putanja koje su jednako udaljene od susjednih prepreka.**
- Za svaku tačku slobodnog prostora računa se njena udaljenost do najbliže prepreke (slika na sljedećem slajdu).
- Veličina dijagrama se povećava što se robot više udaljava od prepreke.



# Voronoijev dijagram

- Veličina dijagrama se povećava što se robot više udaljava od prepreke.



## Voronoijev dijagram

- Za zatvoreni slobodni prostor sa kompaktnim preprekama, Voronoijev dijagram se sastoji od kontinuiranih linija.
- Tako se u bilo kojoj tački Voronoijevog dijagrama udaljenost do najbližih prepreka ne može povećati bez obzira na bilo koje lokalno kretanje po Voronoijevom dijagramu.
- Ako robot slijedi putanju definiranu Voronoijevim dijagramom, on će osim izbjegavanja prepreka maksimizirati (lokalno) svoju udaljenost do njih.
- U tačkama koje su na jednakom odstojanju od dvije ili više prepreka formiraju se oštri vrhovi. Voronoijev dijagram sadrži vrhove dobivene sa ovim oštrim rubnim tačkama.



## Voronoijev dijagram

- Voronoijev dijagram ima važan nedostatak u slučaju ograničenog područja lokalizacijskih senzora, odnosno senzora na temelju čijih mjerenja se obavlja lokalizacija.
- Budući da ovaj algoritam planiranja putanje maksimizira udaljenost između robota i objekata u okolini, bilo koji senzor kratkog dometa na robotu će prouzročiti pogrešnu percepciju okoline.
- Ako se takvi senzori koriste za lokalizaciju tada će odabrana putanja biti prilično loša sa stajališta lokalizacije.
- S druge strane, metod grafa vidljivih vrhova se može dizajnirati na način da zadrži robota po želji blizu objekata (prepreka) u mapi.

## Voronoijev dijagram

- Međutim, postoji i jedva primjetna, važna prednost metode Voronoijevih dijagrama u odnosu na druge metode planiranja kretanja: ***izvršivost***.
- Za zadanu, pojedinačnu planiranu putanju Voronoijevog dijagrama, robot sa sensorima udaljenosti, kao što su laserski skener ili ultrazvučni senzori, može slijediti Voronoijeve vrhove korištenjem jednostavnog upravljačkog algoritma.
- Ovaj upravljački sistem će prirodno držati robota na vrhovima Voronoijevog grafa, tako da kretanje zasnovano na Voronoijevom dijagramu može ublažiti pogreške usljed netačnosti mjerenja enkodera.







## Voronoijev dijagram

- Ovo interesantno svojstvo Voronoijevog dijagrama može se iskoristiti za obavljanje automatskog mapiranja okoline pronalaženjem nepoznatih vrhova Voronoijevih dijagrama i kretanjem po njima, te nakon toga konstruirati Voronoijevu mapu okoline [Choset i saradnici, 2000].
- Dva najčešće korištena načina mjerenja udaljenosti kod Voronoijevog dijagrama su  $L_1$  i  $L_2$  metrike:

$$L_1 \rightarrow \{(x, y) : |x| + |y| = \text{konst.}\}$$

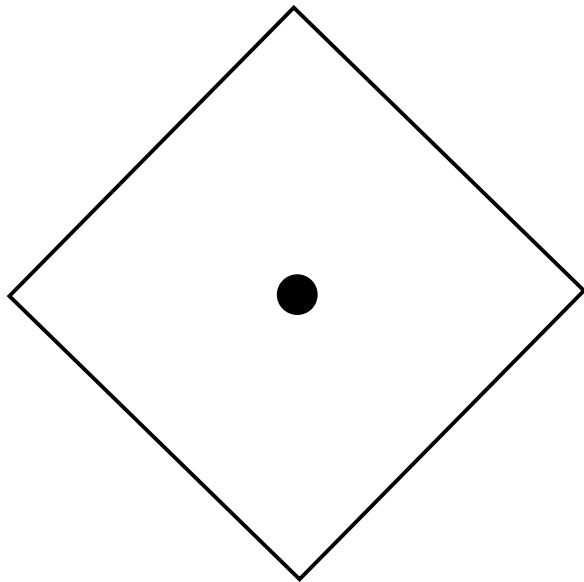
$$L_2 \rightarrow \{(x, y) : x^2 + y^2 = \text{konst.}\}$$

# Voronoijev dijagram

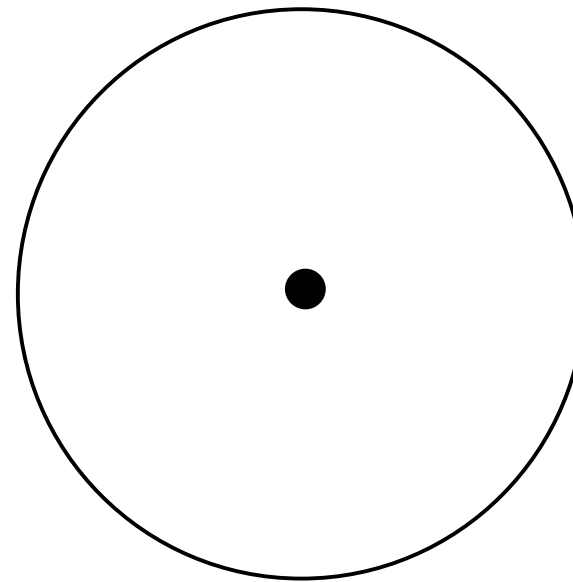
- Grafičke interpretacije navedenih metrika su prikazane na sljedećim slikama.



26/86



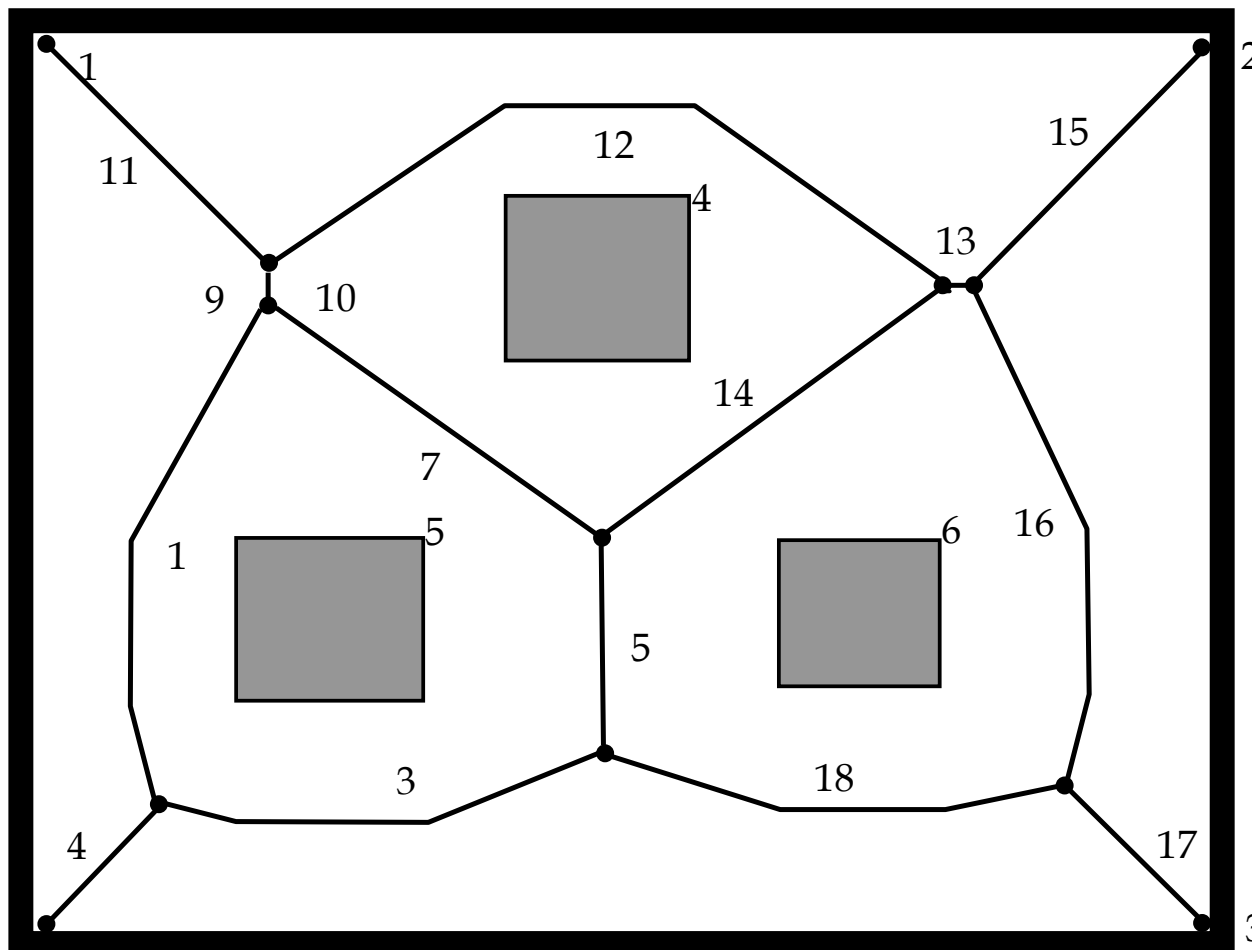
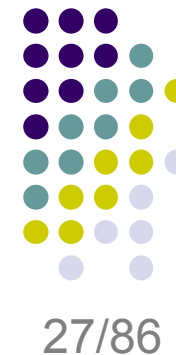
$$\{(x, y) : |x| + |y| = \text{konst.}\}$$



$$\{(x, y) : x^2 + y^2 = \text{konst.}\}$$

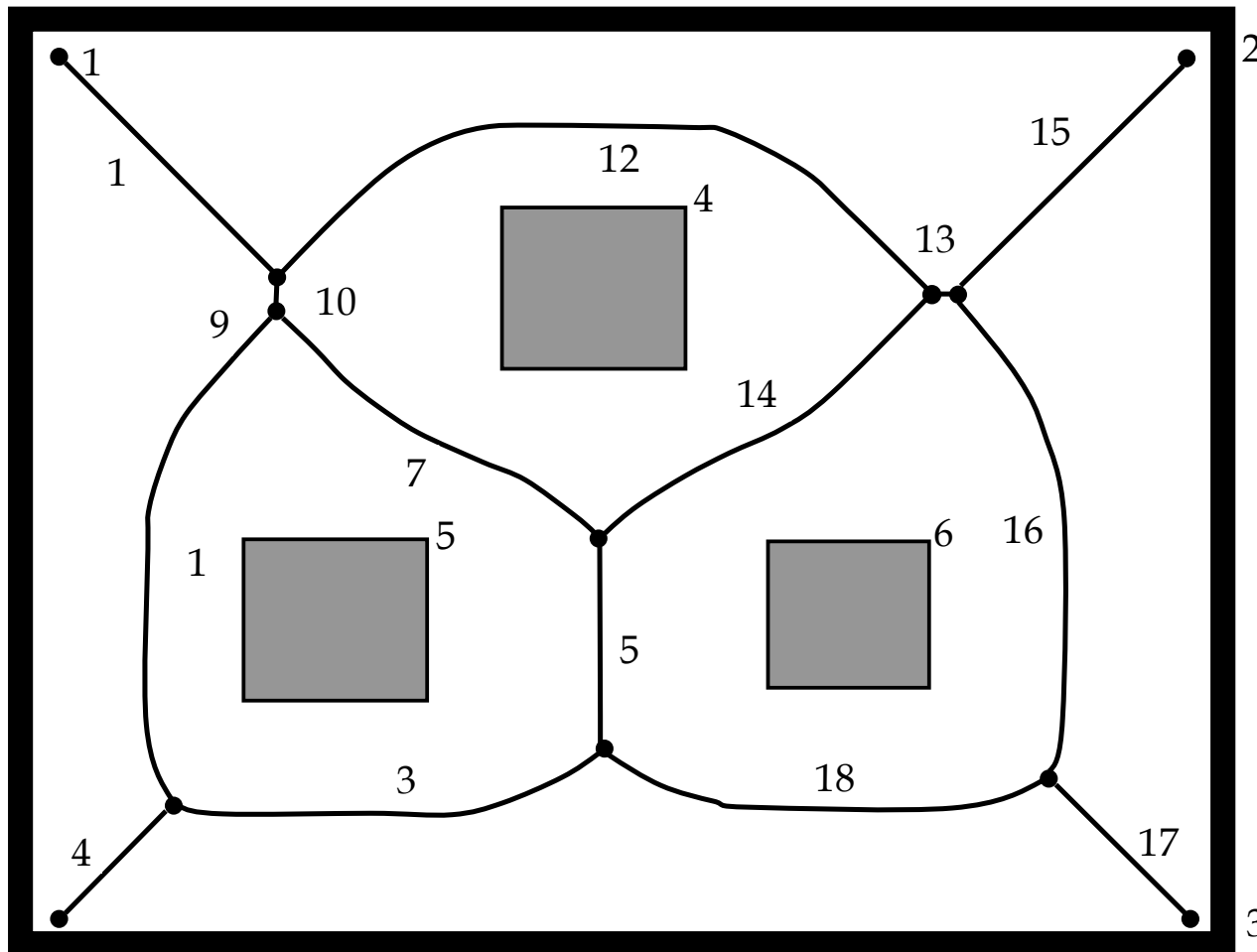
# Voronoijev dijagram

- Primjer kreiranja grafa  $L_1$  metrikom (oštri rubovi).



# Voronoijev dijagram

- Primjer kreiranja grafa sa  $L_2$  metrikom (zaobljeni rubovi).



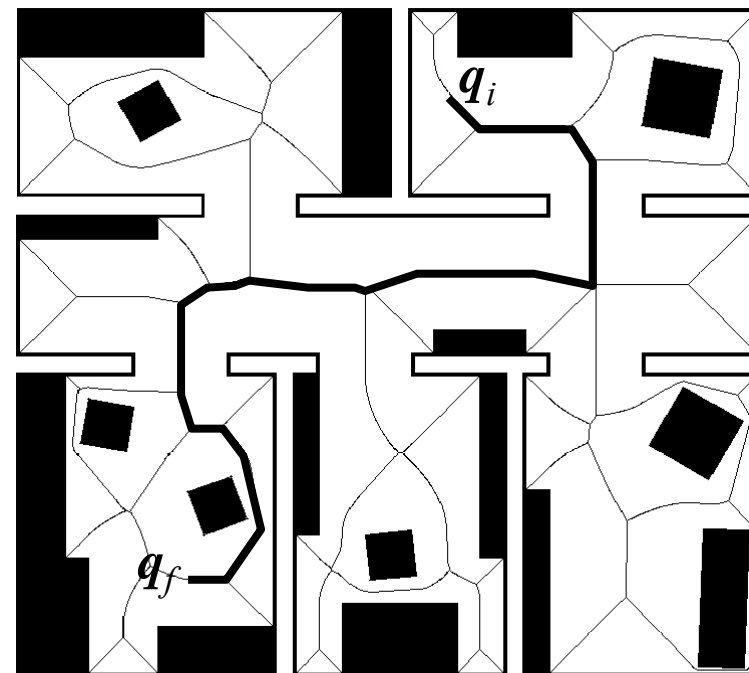
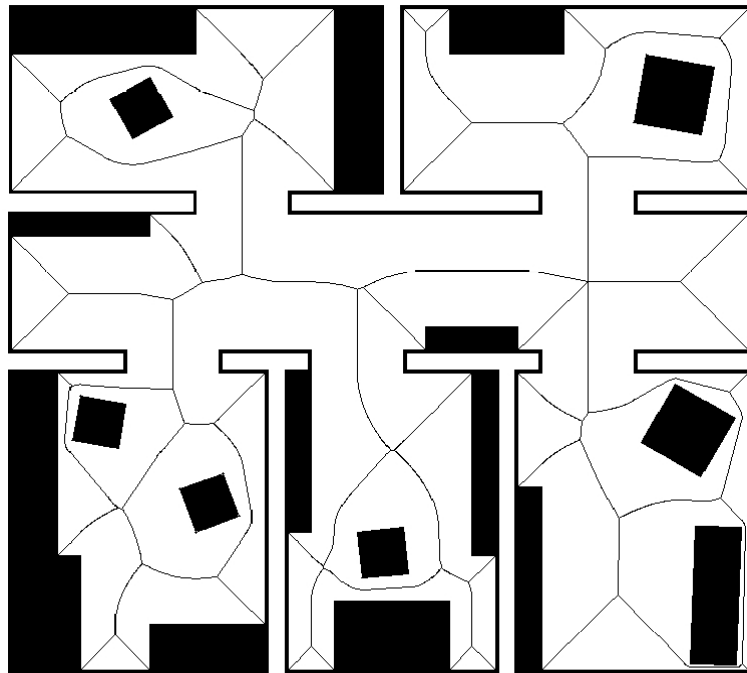
## Voronoijev dijagram

- Kada su zadane početna i ciljna tačka, robot se može kretati na dovoljno velikom rastojanju od najbliže prepreke dok se kreće prema tački koja leži na Voronoijevom dijagramu.
- Nakon toga robot slijedi skup paraboličnih i pravolinijskih segmenata (spojnih linija) koji sačinjavaju Voronoijev dijagram dok ne dostigne lokaciju iz koje se može direktno usmjeriti ka ciljnoj lokaciji uz održavanje maksimalne udaljenosti od prepreka u okolini.



# Voronoijev dijagram

- Planiranje kretanja mobilnog robota korištenjem generaliziranog Voronoijevog dijagrama.



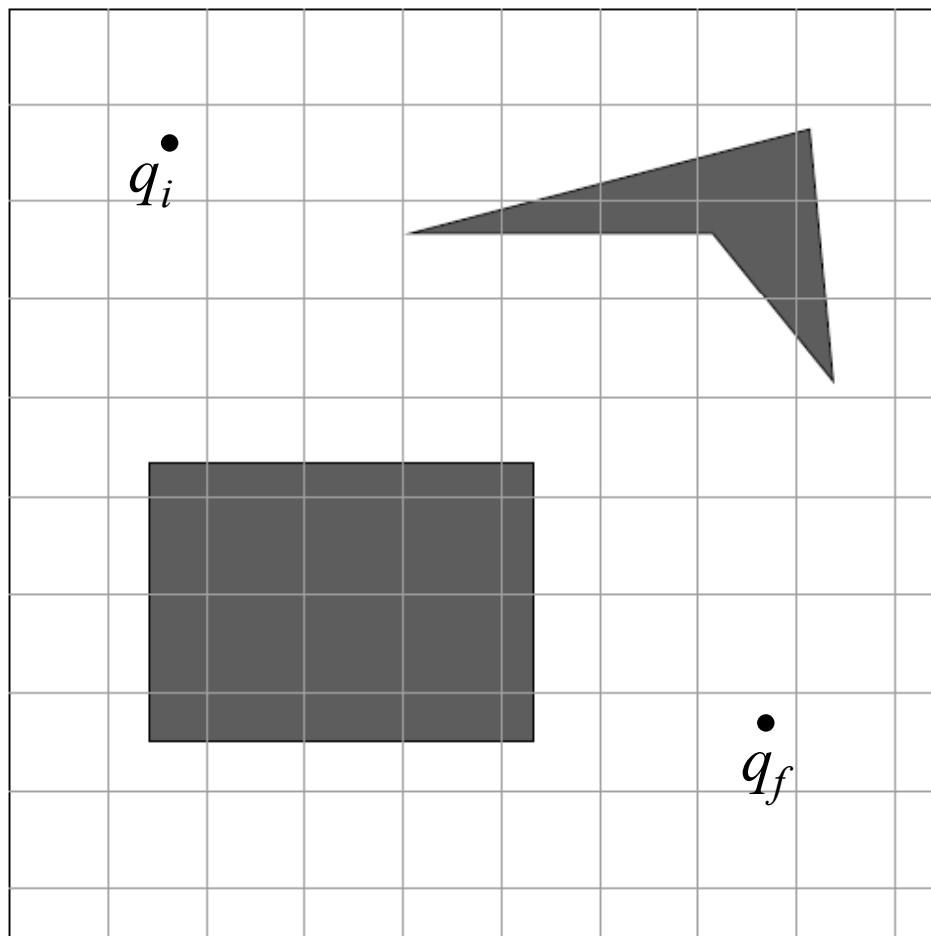
## 12.3. Dekompozicija prostora (segmentata)

- Dekompozicija prostora dijeli slobodni robotski prostor u jednostavne regione, koji se nazivaju ***segmenti (ćelije)***.
- Segmenti su obično konveksne površine pa je potrebno konstantno vrijeme za računanje putanje između bilo koje dvije konfiguracije unutar segmenta.
- Kada se konstruira graf  $G_{seg}$  i uspostavi slobodan prostor unutar njega  $F$  tada se može primijeniti neki od roadmap algoritama. Čvorovi grafa  $G_{seg}$  su segmenti.
- Postoji rub (spojna linija) između dva čvora ako su korespondentni segmenti susjedni jedan u odnosu na drugog.



# Dekompozicija prostora (segmentata)

- **Jednostavna segmentna dekompozicija predstavlja mrežu sa fiksnom rezolucijom.**





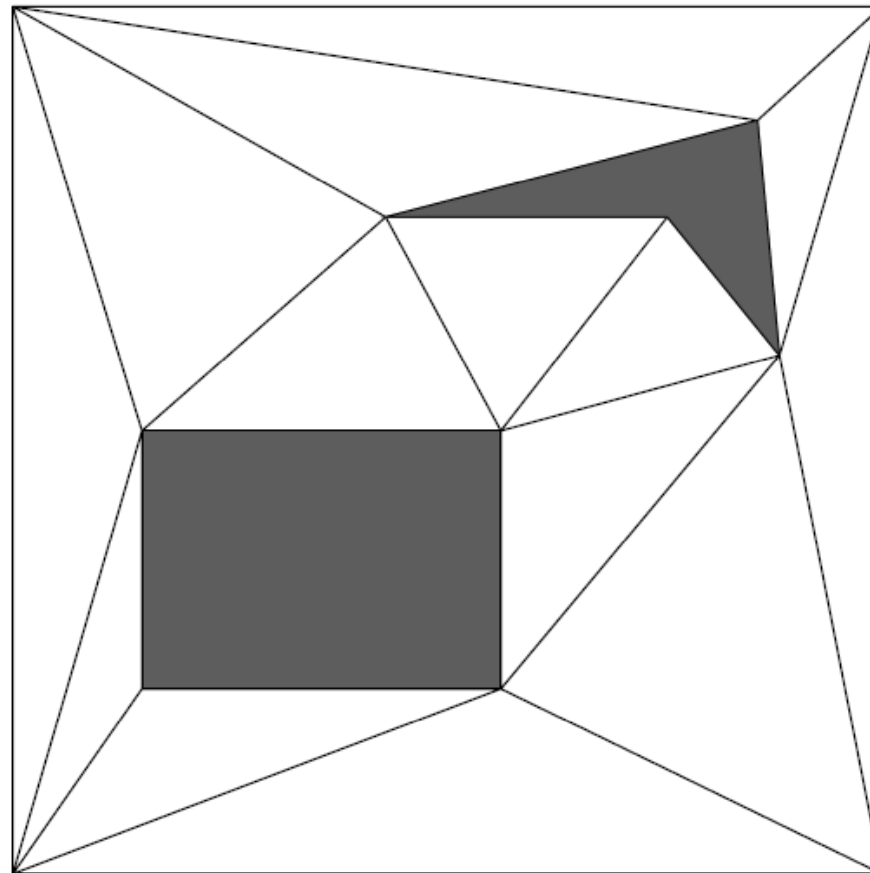
## Dekompozicija prostora (segmenata)

- Da bi se pronašla putanja između početne  $q_i$  i ciljne tačke  $q_f$ , prvo se lociraju dva segmenta unutar kojih se nalaze ove tačke.
- Nakon toga se traži putanja u grafu  $G_{seg}$  između dva korespondentna čvora.
- Rezultat je sekvenca susjednih slobodnih segmenata koji formiraju prolaz u slobodnom prostoru između  $q_i$  i  $q_f$ .
- Glavna prednost ovog algoritma je lagana implementacija, što ga čini široko primijenjenim u planiranju kretanja mobilnih robota.
- Algoritam pronalazi putanju kada ona postoji samo ako je rezolucija mreže dovoljno dobra. Zato se kaže da je ovaj algoritam samo **rezolucijski kompletan**.



## Dekompozicija prostora (segmentata)

- Za prevazilaženje ovog problema koristi se **algoritam triangulacije** (de Berg i saradnici, 2000) koji dijeli slobodan prostor u trokute korištenjem vrhova poligonalnih prepreka.



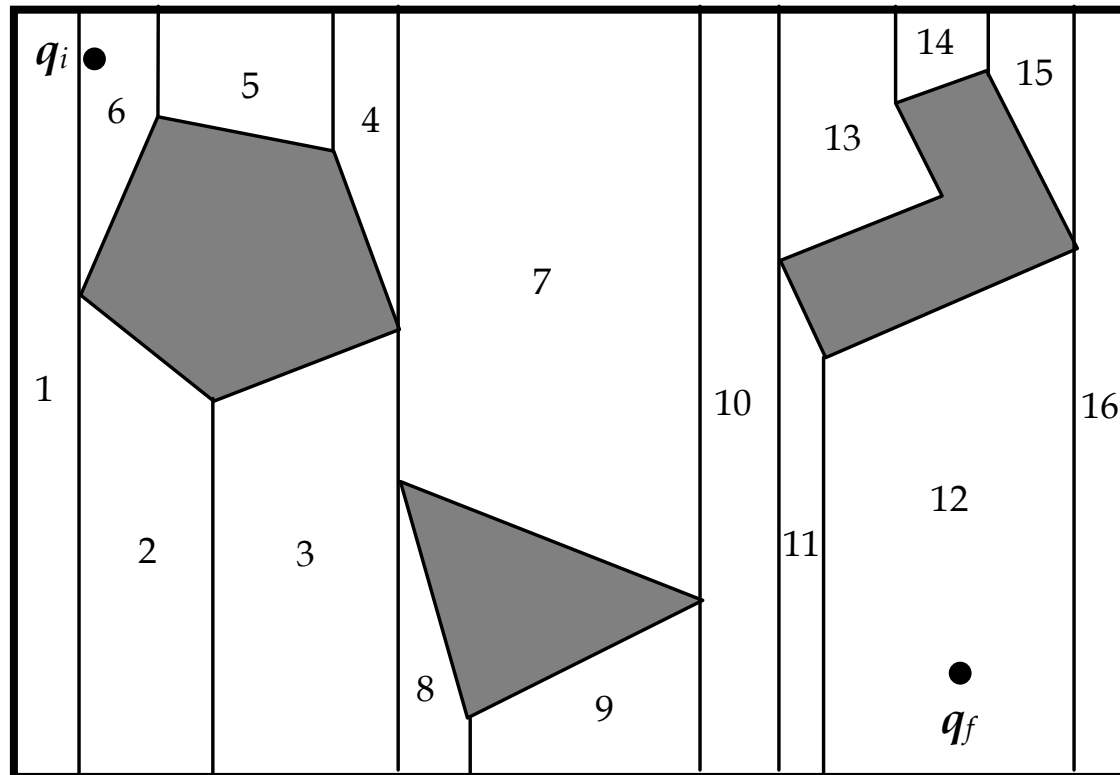
## Dekompozicija prostora (segmentata)

- Važan aspekt metoda dekompozicije prostora je postavljanje granica između segmentata.
- Ako su granice postavljene korištenjem funkcije strukture okoline tada se govori o **egzaktnoj dekompoziciji** (Latombe, 1991).
- Pod strukturom okoline se podrazumijevaju pojedinačne prepreke i slobodni prostor.
- Rezultantni segmenti su u cijelosti zauzeti ili su u cijelosti prazni.
- Pojedinačna pozicija robota unutar svakog slobodnog segmenta nije od značaja, već je važna mogućnost prelaska robota iz pojedinačnog slobodnog segmenta u njegove susjedne slobodne segmente.



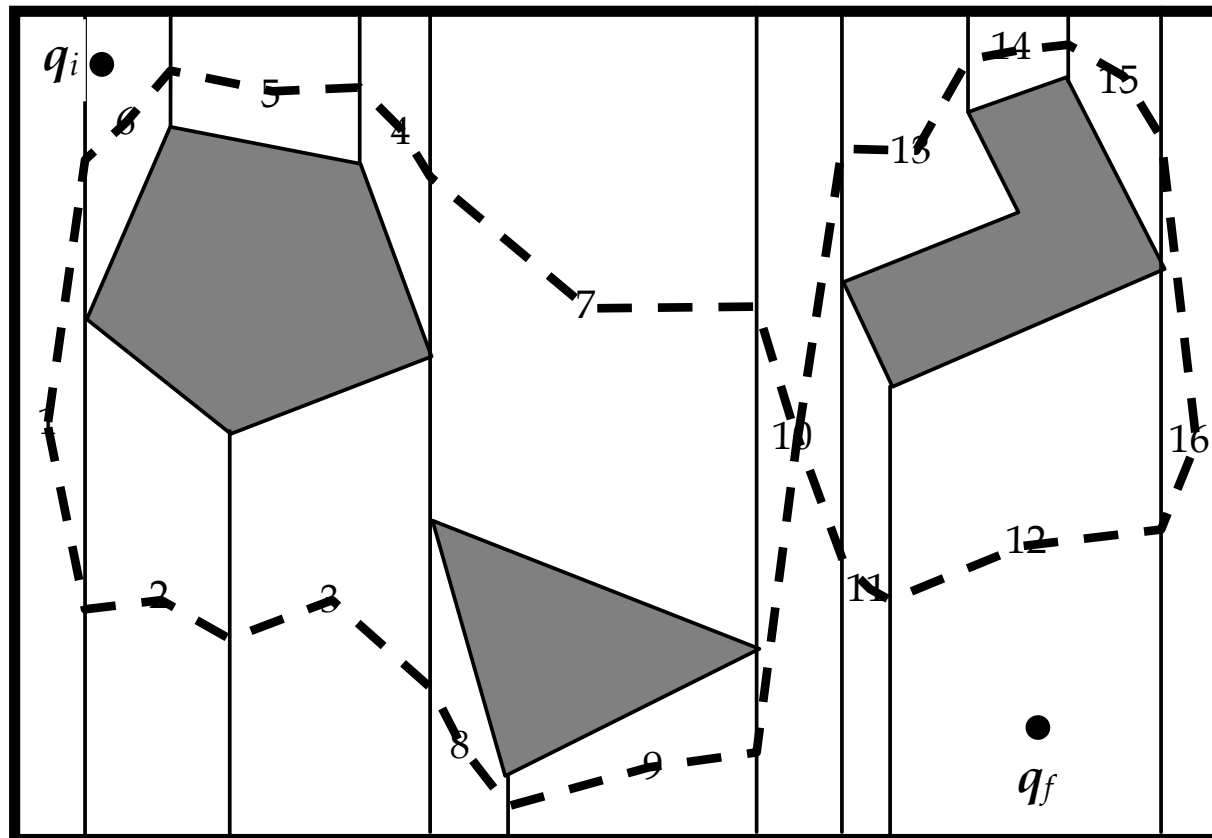
# Dekompozicija prostora (segmentata)

- **Proces planiranja putanje egzaktnom dekompozicijom segmentata:**
- Prvo se slobodni prostor podijeli u segmente.
- Zatim se u svakom segmentu pronađe centroid, odnosno centar područja i označi se brojem.



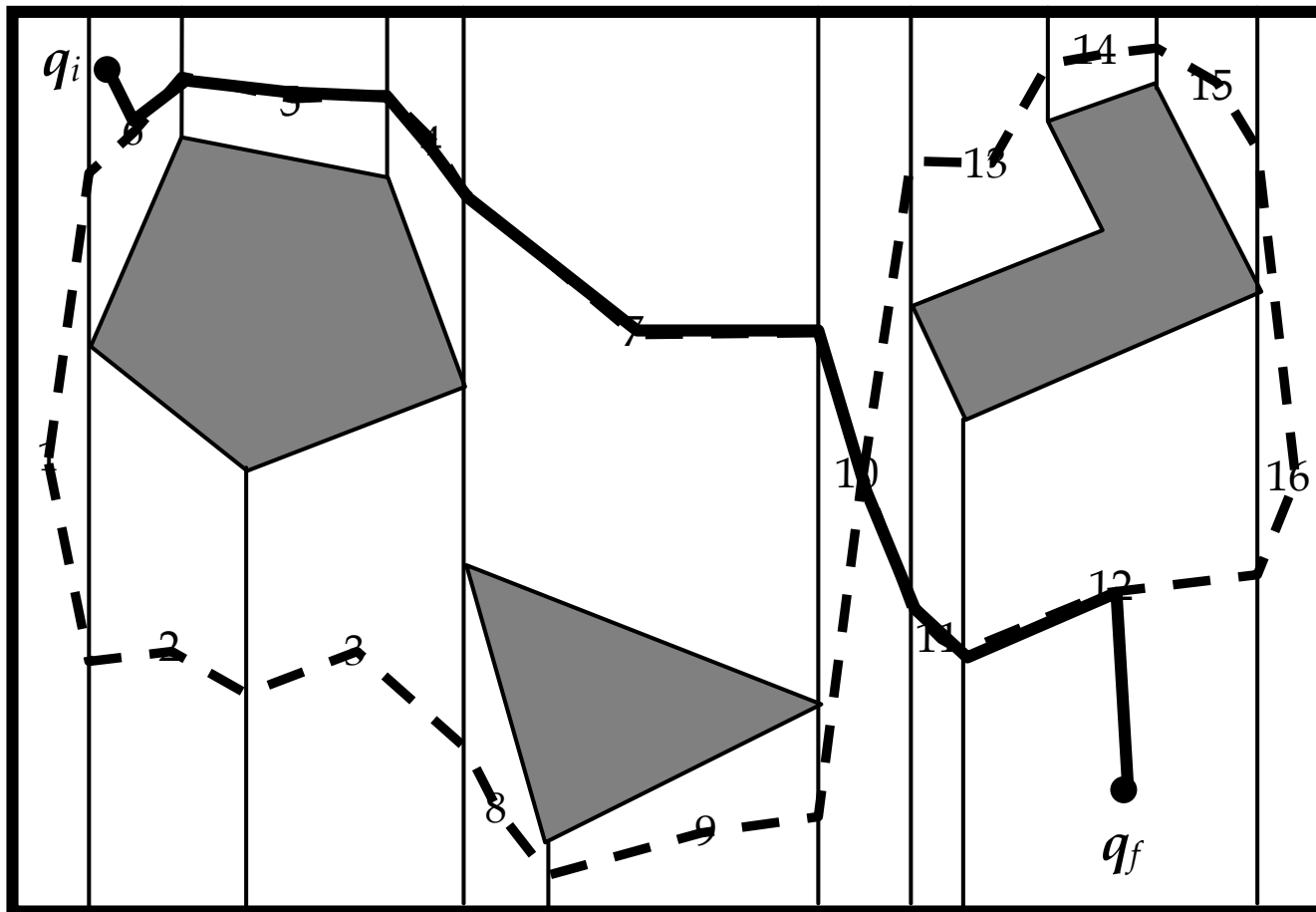
# Dekompozicija prostora (segmenata)

- Ovi centriodi predstavljaju čvorove, odnosno vrhove povezanog grafa.
- Povezivanje dva susjedna vrha grafa se obavlja preko tačke na sredini njihove granice.



# Dekompozicija prostora (segmenata)

- Najkraća pronađena putanja je označena punom linijom.



## Dekompozicija prostora (segmenata)

- **Ključni nedostatak egzaktne dekompozicije segmenata je da broj segmenata i računarska efikasnost cjelokupnog procesa planiranja putanje ovise o gustoći i složenosti prepreka u okolini.**
- U slučaju da je koncentracija prepreke u okolini značajno smanjena, tada će biti malo segmenata čak i ako su dimenzije okoline veoma velike.
- Zbog složenosti u implementaciji ova se dekompozicija relativno rijetko koristi mobilnoj robotici.



## Dekompozicija prostora (segmenata)

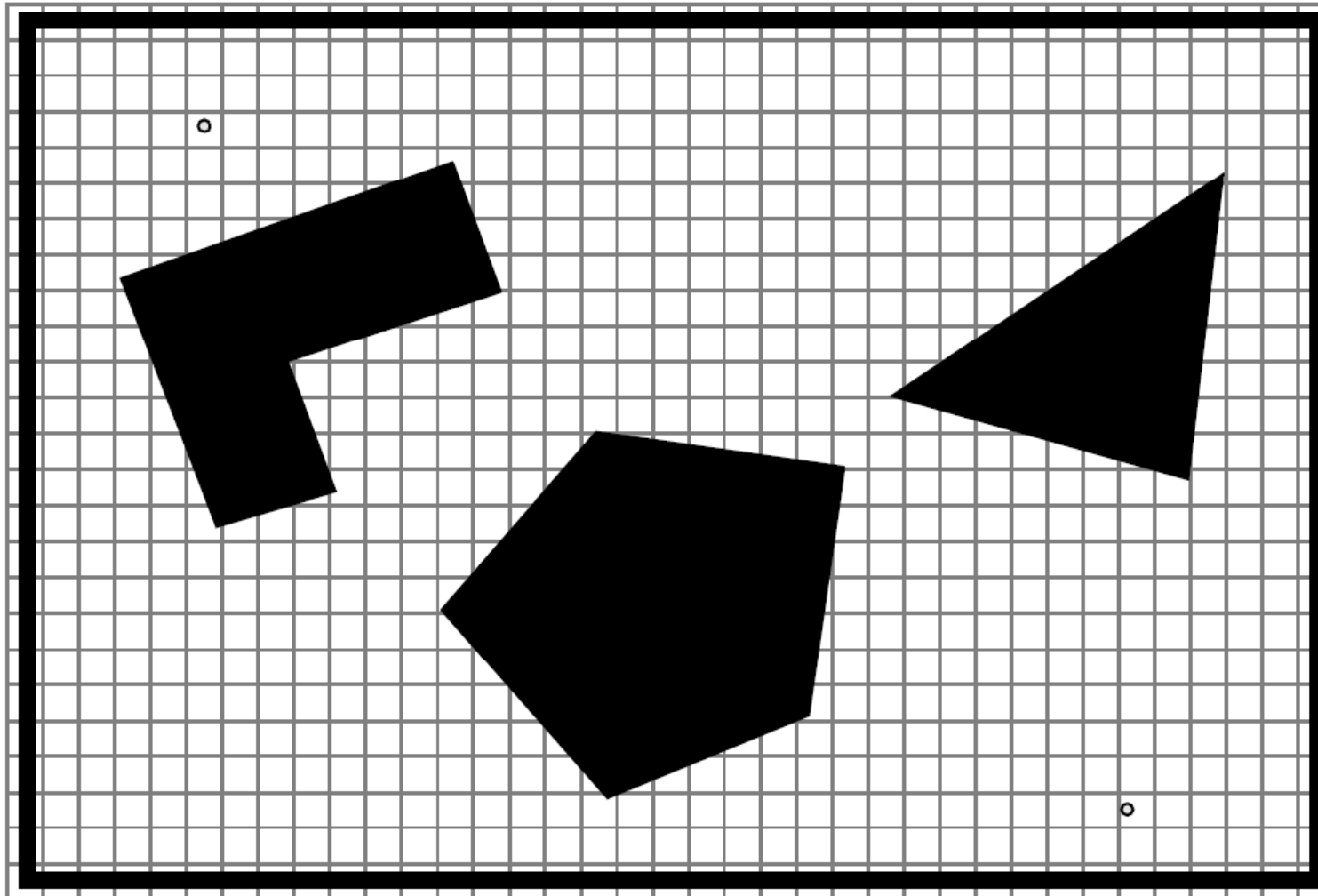
- Kada dekompozicija rezultira aproksimacijom stvarne mape tada se govori o **aproksimativnoj dekompoziciji**, koja je jedna od najpopularnijih tehnika planiranja putanje mobilnih robota.
- Najvažniji razlog za ovo je mrežasti prikaz robotske okoline.
- Najviše korištena tehnika aproksimativne dekompozicije je **fiksna dekompozicija** koja formira diskretnu mrežu segmenata fiksnih dimenzija (slika na sljedećoj slici).





# Dekompozicija prostora (segmenata)

- Aproksimativna fiksna dekompozicija prostora



## Dekompozicija prostora (segmenata)

- Dimenzije segmenata u fiksnoj dekompoziciji prostora ne ovise o pojedinačnim preprekama u okolini.
- Mora se paziti da segment ne bude previše mali, jer će u slučaju okoline velikih dimenzija biti mnogo segmenata što će zahtijevati znatne memorijske resurse za pohranu informacija o pojedinom segmentu, bez obzira na gustoću i broj prepreka u okolini.
- Glavna prednost fiksne dekompozicije je niska računarska složenost planiranja putanje.

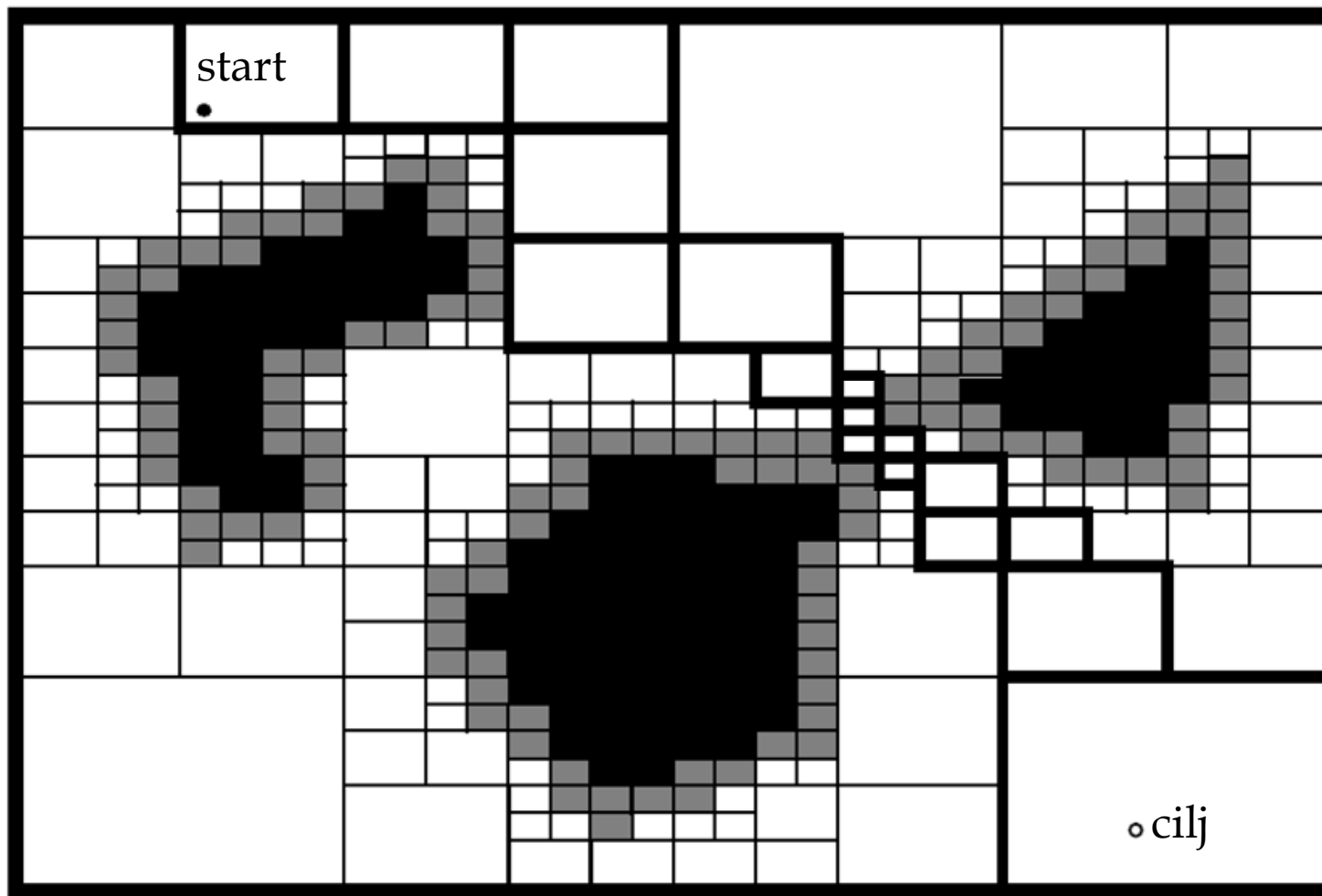
## Dekompozicija prostora (segmenata)

- Osim fiksne dekompozicije, postoji i **varijabilna dekompozicija prostora** (adaptivna dekompozicija).
- Na sljedećoj slici slobodni prostor je izvana ograničen pravokutnikom, a iznutra sa tri poligona.
- Pravokutnik je rekurzivno dekomponiran (razložen) u veći broj manjih pravokutnika.
- Svaka dekompozicija dijeli svaki djelomično zauzet/djelomično slobodan segment prostora u četiri identična podsegmenta.
- Na svakoj razini rezolucije samo segmenti koji leže isključivo unutar slobodnog prostora se koriste za konstrukciju povezivog grafa.



# Dekompozicija prostora (segmenata)

- Varijabilna dekompozicija prostora



## Dekompozicija prostora (segmenata)

- Planiranje putanje u ovakvoj adaptivnoj dekompoziciji se može hijerarhijski izvršavati.
- Planiranje započinje sa grubom rezolucijom, koja se u svakom sljedećem koraku reducira dok se ne identificira rješenje ili se ne postigne limit na rezoluciju.
- Kada se uspoređuju dekompozicije sa fiksnim i varijabilnim dimenzijama segmenata, tada se ova druga lakše adaptira složenijim okolinama, pa će stoga okoline sa manjom gustoćom i koncentracijom prepreka sadržavati manji broj segmenata, što za sobom povlači značajno manje memorije za pohranu.



## 12.4. Metoda potencijalnih polja

- **Osnovni koncept metode potencijalnih polja je „napuniti“ radni prostor robota umjetnim potencijalnim poljem u kome će ciljna pozicija privlačiti robota (privlačna sila), a prepreke ga odbijati [Khatib, 1986].**
- Drugim riječima, robot i prepreke imaju isti polaritet, a cilj suprotan polaritet, pri čemu cilj predstavlja minimum u prostoru, a prepreke maksimume (predstavljene šiljcima u potencijalnom polju).
- **U metodi potencijalnih polja robot se razmatra kao tačka u umjetnom potencijalnom polju koji se kreće u smjeru gradijenta polja prema ciljnoj lokaciji.**
- Ova metoda je posebno atraktivna zbog mogućnosti njene elegantne matematičke analize i pojednostavljenja.



## Metoda potencijalnih polja

- Konvencionalno, aktivni potencijal je definiran kao funkcija relativne udaljenosti robota i cilja, pri čemu je cilj fiksna tačka u prostoru.
- Umjetno potencijalno polje  $U(\mathbf{q})$  se formira od komponente pridružene cilju  $U_{cilj}(\mathbf{q})$  i komponenti pridruženih preprekama  $U_{prep}(\mathbf{q})$ .
- Rezultantno umjetno potencijalno polje predstavlja sumu navedene dvije komponente:

$$U(\mathbf{q}) = U_{cilj}(\mathbf{q}) + \sum U_{prep}(\mathbf{q}) \quad (*)$$

- Funkcija potencijala specificira kretanje robota u bilo kojoj konfiguraciji  $\mathbf{q} \in C$ .



## Metoda potencijalnih polja

- Budući da se robot promatra kao tačka  $(x, y)$  to se njegova orijentacija može zanemariti i rezultatno potencijalno polje je predstavljeno u dvodimenzionalnom prostoru (ravnini).
- Na temelju izraza (\*) može se izračunati izraz za silu umjetnog polja koja djeluje na poziciju  $\mathbf{q} = (x, y)$ :

$$F = -\nabla U(\mathbf{q}) = - \begin{pmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{pmatrix}$$

gdje  $\nabla U(\mathbf{q})$  označava gradijent vektora  $U(\mathbf{q})$ .



## Metoda potencijalnih polja

- Predznak – u prethodnom izrazu znači da se robot kreće ka cilju (koji je obično globalni minimum od  $U(\mathbf{q})$ ) u smjeru negativnog gradijenta vektora  $U(\mathbf{q})$ .
- Sila  $F$  se također može razdijeliti na privlačnu (cilj) i odbojnu (prepreka) silu:

$$\begin{aligned} F(\mathbf{q}) &= F_{cilj}(\mathbf{q}) - F_{prep}(\mathbf{q}) \\ &= -\nabla U_{cilj}(\mathbf{q}) - \nabla U_{prep}(\mathbf{q}) \end{aligned}$$

- Rezultantna sila se dobiva superpozicijom privlačne sile kojom cilj djeluje na robota i odbojnih sila kojima prepreke djeluju na robota.
- Ova sila, odnosno potencijal, će voditi robota ka cilju uz istovremeno izbjegavanje prepreka koje se nađu na putanji.



## Metoda potencijalnih polja

- Potencijal kojim cilj privlači robota se može opisati paraboličnom funkcijom oblika:

$$U_{cilj}(\mathbf{q}) = \frac{1}{2} \alpha \|\mathbf{q} - \mathbf{q}_{cilj}\|^2$$

gdje je  $\alpha$  pozitivna skalirajuća konstanta i  $\|\mathbf{q} - \mathbf{q}_{cilj}\|$  Euklidska udaljenost između vektora pozicije robota i cilja.

- Ovaj potencijal je diferencijabilna funkcija i na temelju njega se može izračunati sila kojom cilj privlači robota:

$$\begin{aligned} F_{cilj}(\mathbf{q}) &= -\nabla U_{cilj}(\mathbf{q}) = -\alpha \|\mathbf{q} - \mathbf{q}_{cilj}\| \nabla \left( \|\mathbf{q} - \mathbf{q}_{cilj}\| \right) \\ &= -\alpha (\mathbf{q} - \mathbf{q}_{cilj}) \end{aligned}$$



## Metoda potencijalnih polja

- Ova sila konvergira linearno ka nuli kako se robot približava cilju.
- Odbojni potencijal je veoma jak kada je robot blizu prepreke i slabi kako se robot udaljava od nje.
- Jedan od načina izražavanja ovog potencijala je:

$$U_{prep}(\mathbf{q}) = \frac{\beta}{\|\mathbf{q} - \mathbf{q}_{prep}\|}$$

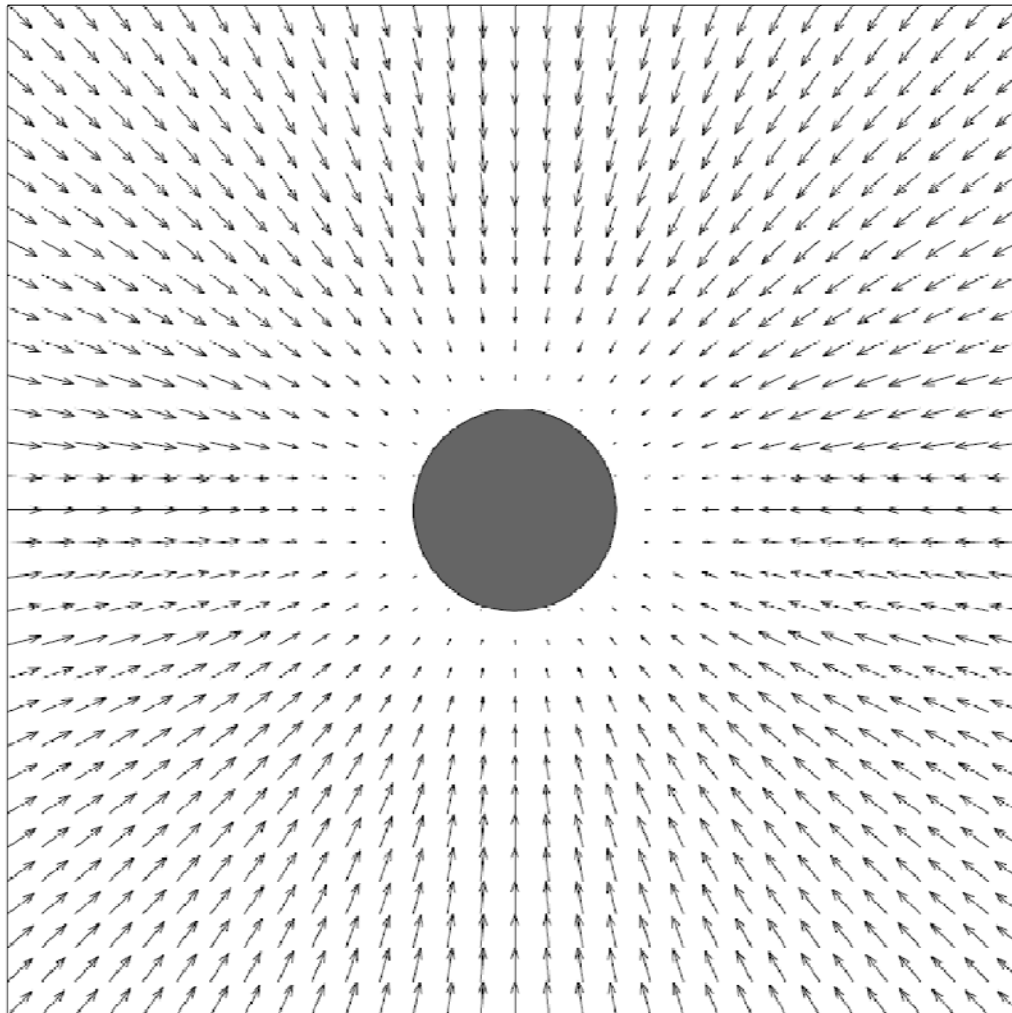
gdje je  $\mathbf{q}_{prep}$  pozicija prepreke.

- Korištenjem prethodnog izraza računa se sila kojom prepreka odbija robota na sljedeći način:

$$\mathbf{F}_{prep}(\mathbf{q}) = -\nabla U_{prep}(\mathbf{q}) = \frac{\beta}{\|\mathbf{q} - \mathbf{q}_{prep}\|^2}$$

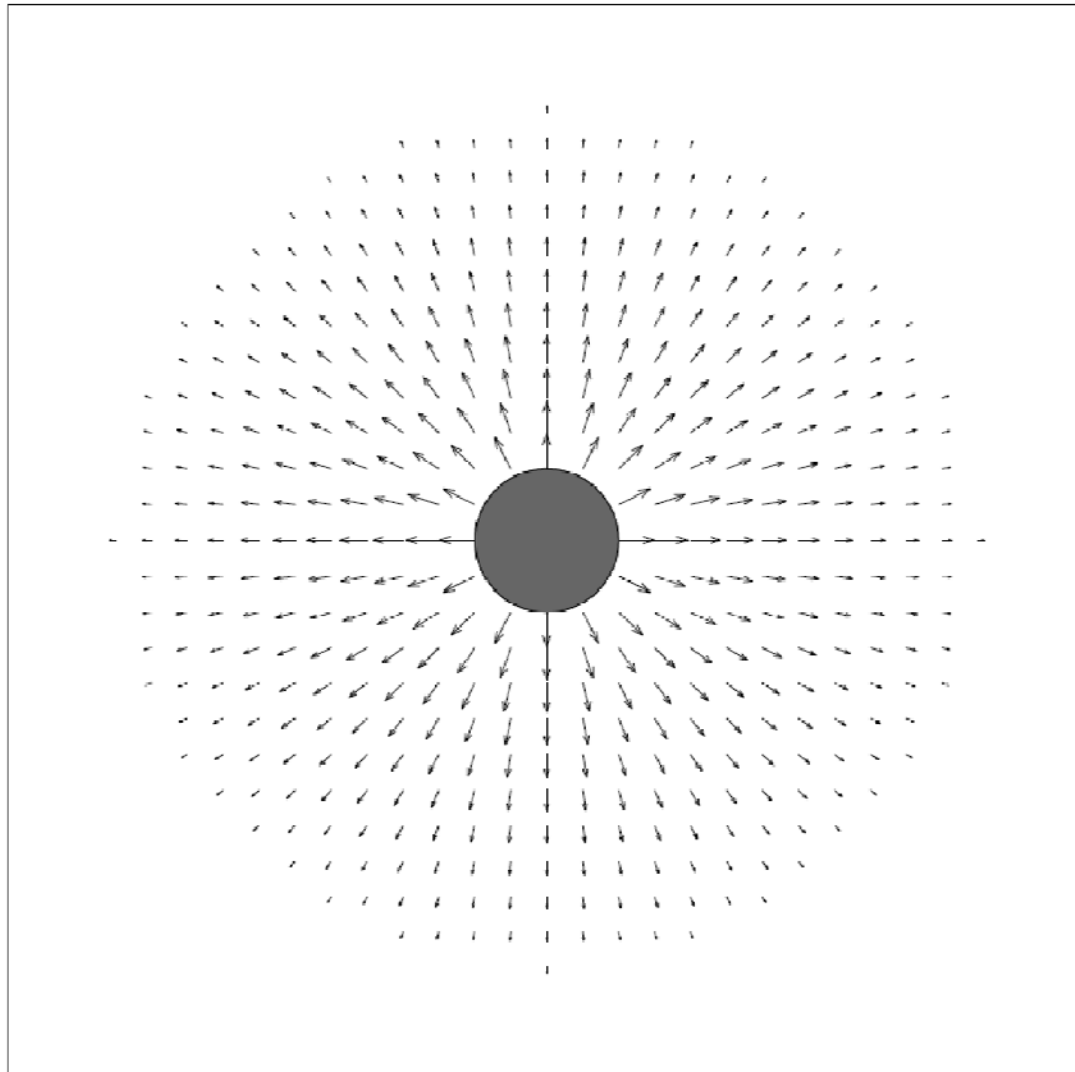
# Metoda potencijalnih polja

- Privlačno potencijalno polje cilja



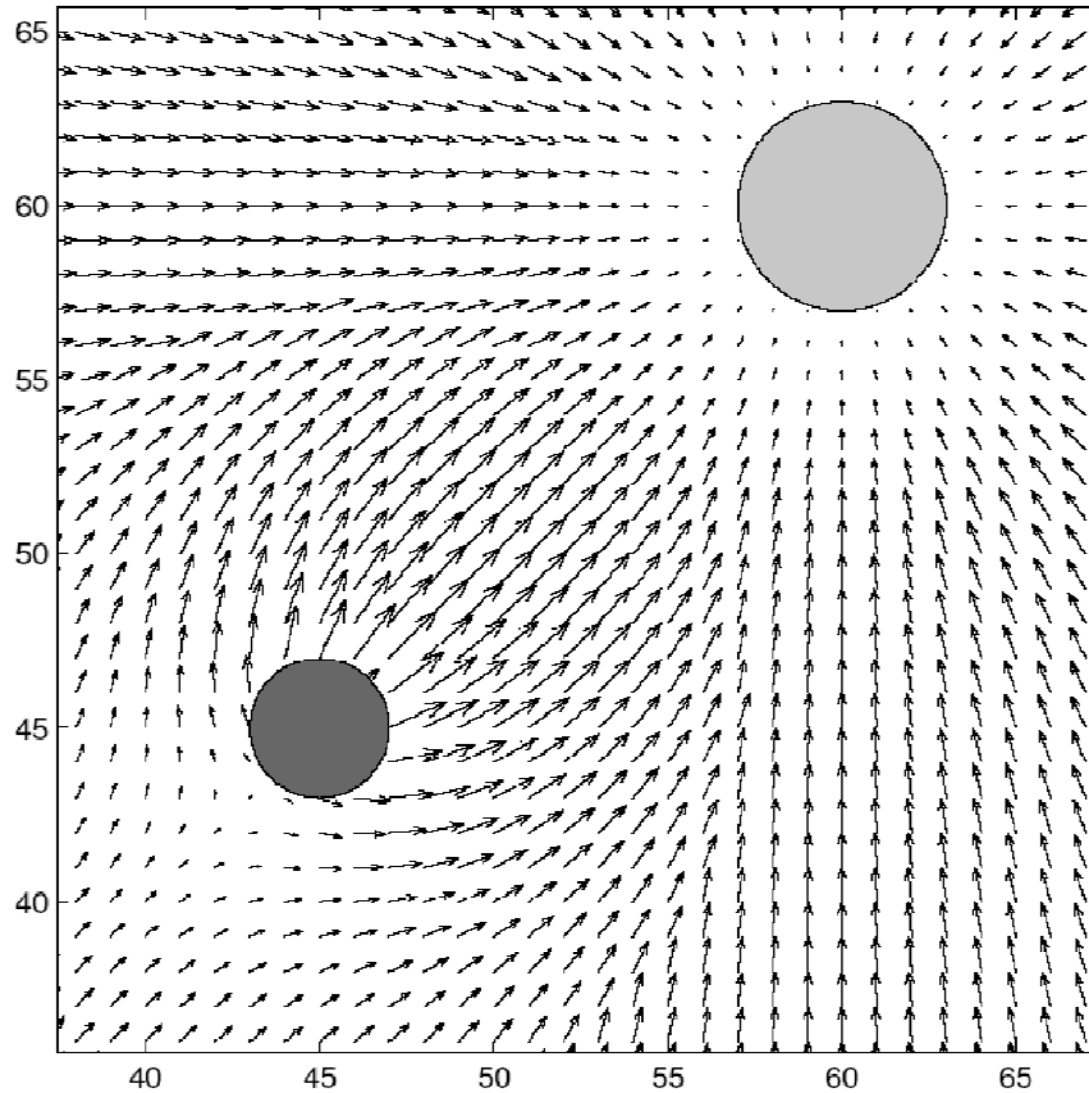
# Metoda potencijalnih polja

- Odbojno potencijalno polje prepreke



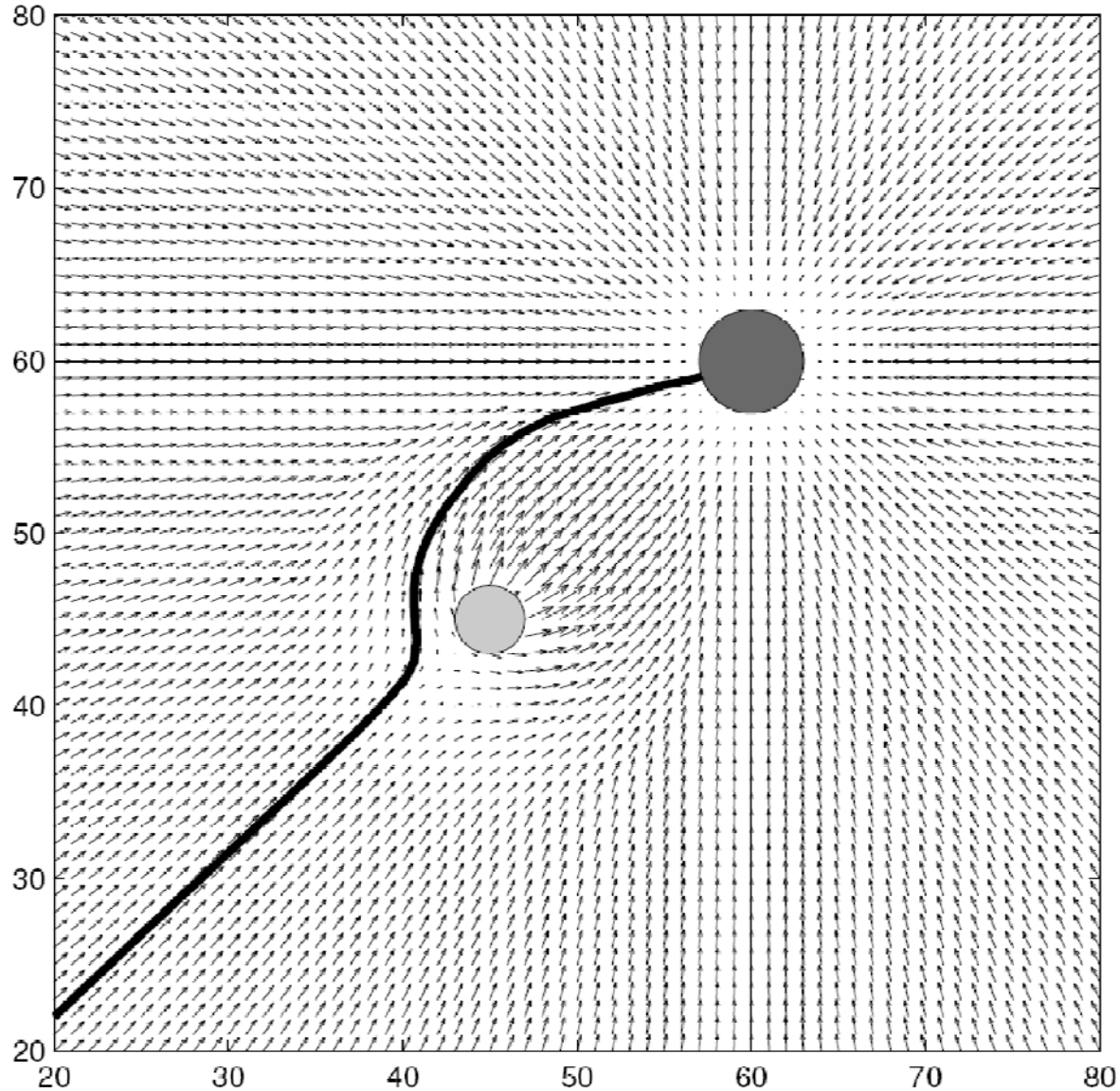
# Metoda potencijalnih polja

- Rezultantno potencijalno polje



# Metoda potencijalnih polja

- Rezultantno potencijalno polje



## Metoda potencijalnih polja

- Metoda potencijalnih polja je prilično efikasna i robusna u odnosu na upravljačke i senzorske (mjerne) pogreške.
- Međutim, ova metoda, koja spada u skupinu metoda optimizacije najbržim spustom, nije efikasna u slučaju problema **lokalnog minimuma**.
- U slučaju lokalnog minimuma robot može upasti u zamku i ne dostići globalni minimum (cilj).
- Za rješavanje problema lokalnog minimuma ne postoji općeniti postupak već se metodi potencijalnih polja dodaju stohastička pretraživanja koja omogućuju izlazak robota iz zamke [Latombe, 1991].





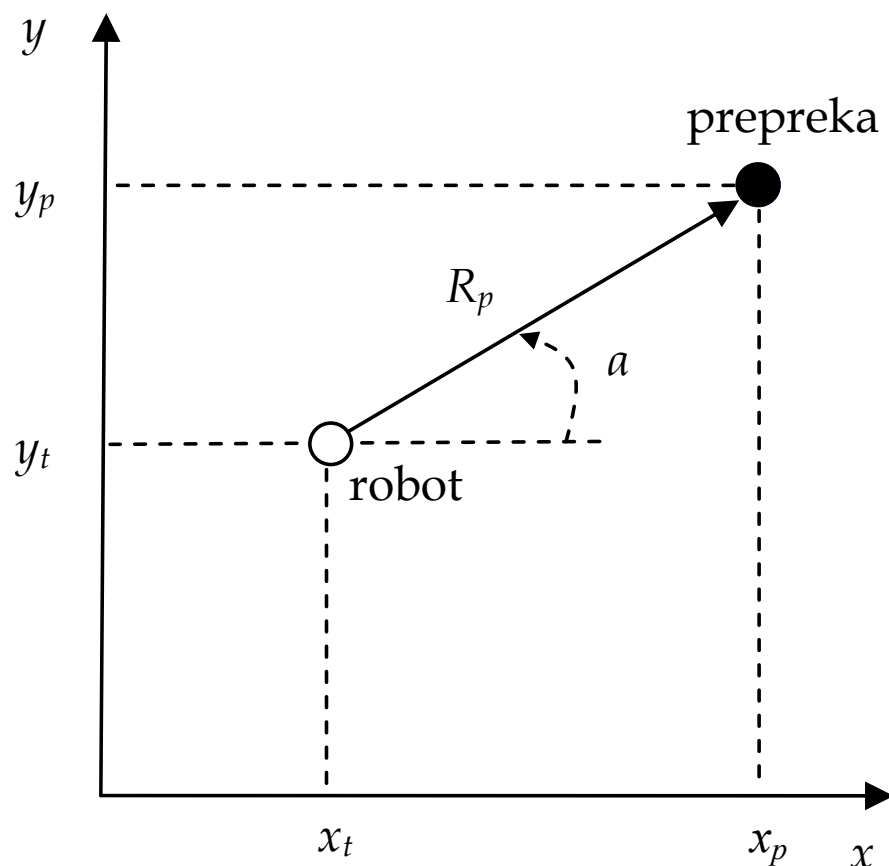
## 12.5. Metoda privlačnih i odbojnih sila

- U nastavku će se prezentirati algoritam **planiranja putanje zasnovan na principu djelovanja privlačnih i odbojnih sila, koji reducira broj prepreka koje imaju dominantan utjecaj na kretanje** robota u svakom pojedinom vremenskom trenutku [Velagić, Lačević i Peruničić, 2005; 2006; Velagić, Lačević i Osmić, 2006].
- Reduciranje broja prepreka se obavlja pomoću neizrazitih pravila.
- Kako je već navedeno privlačna sila se pridružuje cilju, a odbojne sile preprekama. Sve sile se predstavljaju pravcima i rezultantna sila predstavlja smjer kretanja robota u datom trenutku.
- Robot nema unaprijed podatke o okolini, koje dobiva posredstvom ultrazvučnih mjerenja okoline.



## Metoda privlačnih i odbojnih sila

- Objasnimo šta se dešava kada robot  $(x_t, y_t)$  na svom putu prema ciljnoj tački detektira prepreku definiranu sa  $(x_p, y_p)$



$$\Delta x_p = x_p - x_t$$

$$\Delta y_p = y_p - y_t$$

$$R_p^2 = \Delta x_p^2 + \Delta y_p^2$$

Odbojna sila:

$$F_r = \frac{1}{R_p^2}$$

# Metoda privlačnih i odbojnih sila

- Projekcije odbojne sile na pojedine osi:

$$F_{rx} = F_r \cdot \cos(\alpha)$$

$$F_{ry} = F_r \cdot \sin(\alpha)$$

- gdje je ugao  $\alpha$  jednak:

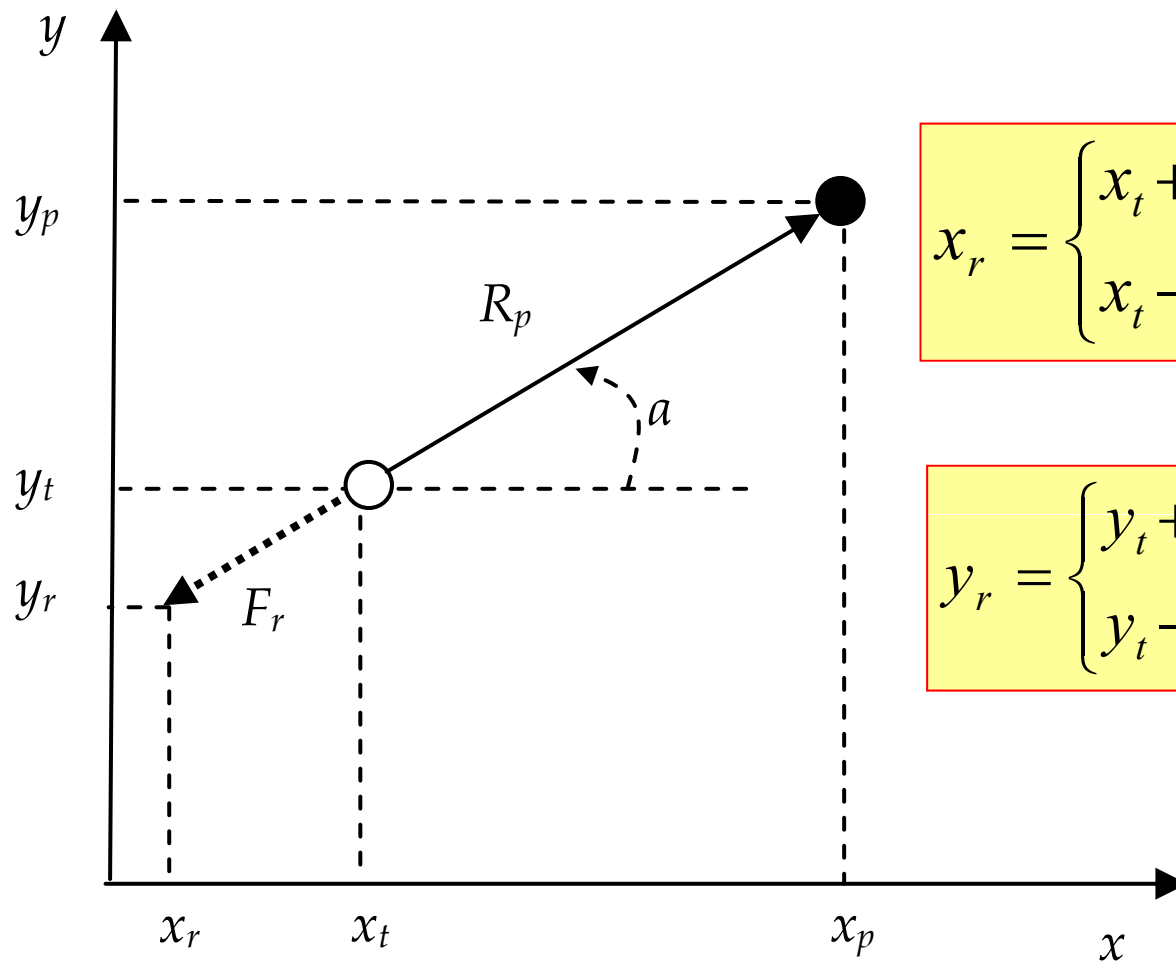
$$\alpha = \arctan \frac{\Delta y_p}{\Delta x_p}$$

- Koordinate vrha vektora odbojne sile u odnosu na koordinate trenutne pozicije robota se računaju prema izrazima na sljedećem slajdu.



# Metoda privlačnih i odbojnih sila

- Pridruživanje odbojne (reaktivne) sile prepreci.

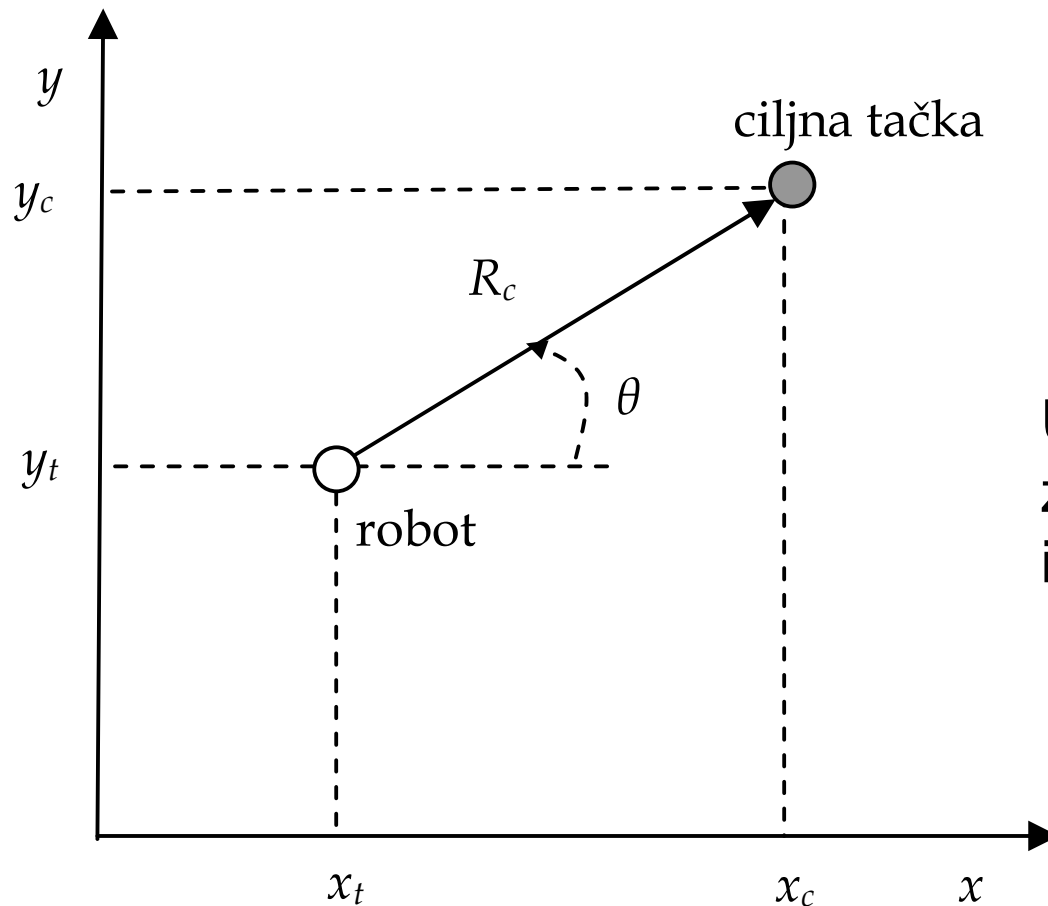


$$x_r = \begin{cases} x_t + |F_{rx}|, & \text{za } \Delta x_p < 0 \\ x_t - |F_{rx}|, & \text{za } \Delta x_p > 0 \end{cases}$$

$$y_r = \begin{cases} y_t + |F_{ry}|, & \text{za } \Delta y_p < 0 \\ y_t - |F_{ry}|, & \text{za } \Delta y_p > 0 \end{cases}$$

# Metoda privlačnih i odbojnih sila

- Ciljnoj tački se pridružuje sila kojom ona privlači robota.



$$\Delta x_c = x_c - x_t$$
$$\Delta y_c = y_c - y_t$$
$$R_c = \sqrt{\Delta x_c^2 + \Delta y_c^2}$$

U predloženom algoritmu zakon po kome se mijenja iznos privlačne sile je:

$$F_a = \begin{cases} \kappa |R_c|, & \text{za } |R_c| \leq 1 \text{ m} \\ |R_c|, & \text{za } |R_c| > 1 \text{ m} \end{cases}$$

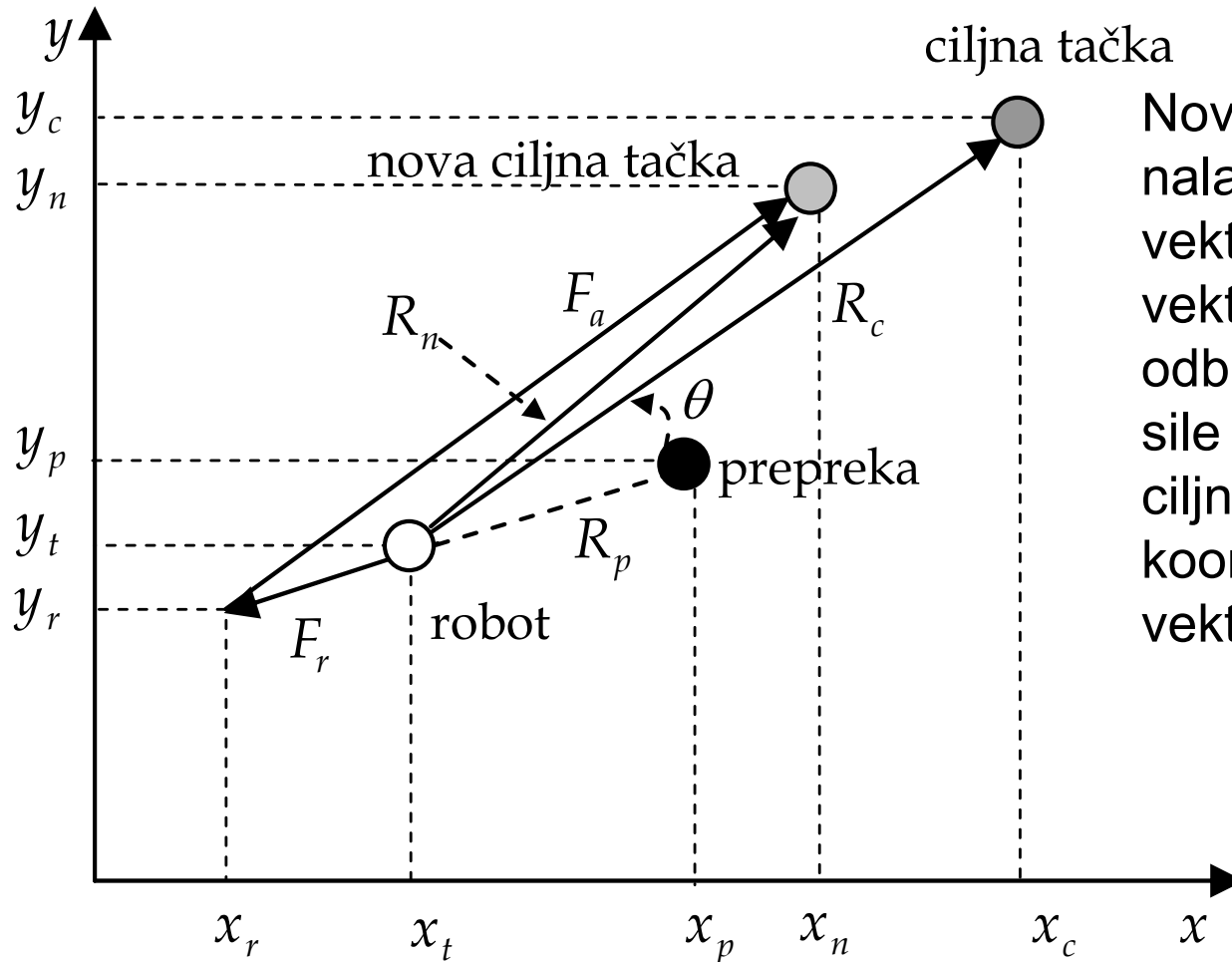
## Metoda privlačnih i odbojnih sila

- U ovom algoritmu je odabran  $\kappa=5$ .
- Velik iznos ovog parametra je odabran zbog toga što u krugu radijusa 1 m oko ciljine tačke relevantne prepreke koje utječu na kretanje robota (kasnije će se razviti pravila za određivanje koja prepreka je relevantna) bi proizvele jake odbojne sile i odbacile robota unatrag, tako da on nikad ne bi stigao u ciljnu tačku.
- Iznos ovog parametra je povezan sa područjem čitanja sonara (u našem slučaju  $R=4$  m).
- U slučaju većeg  $R$ -a, morao bi se izabrati veći  $\kappa$ .



# Metoda privlačnih i odbojnih sila

- Određivanje nove ciljne tačke.

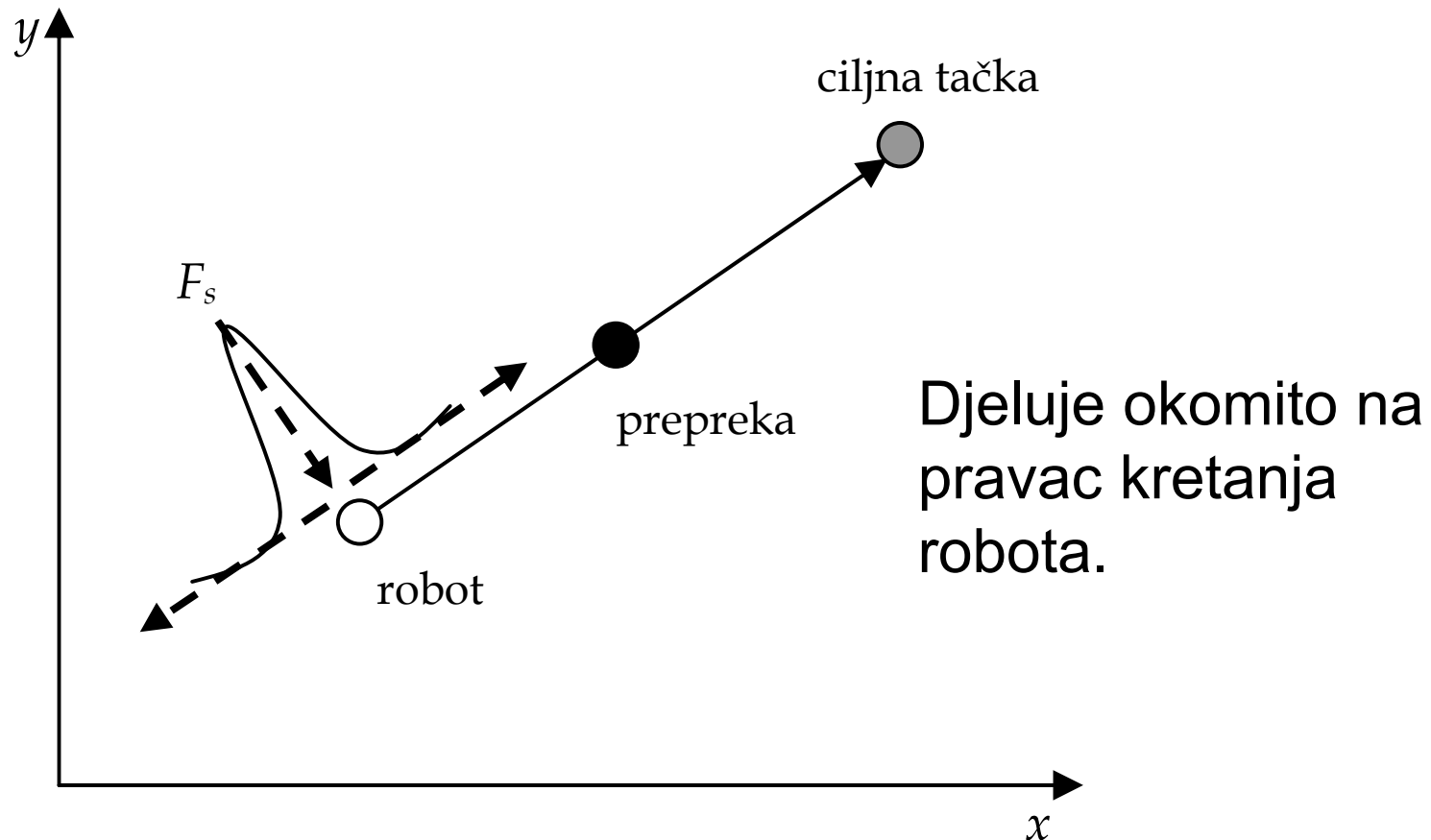


Nova ciljna tačka se nalazi na poziciji vrha vektora koji se dobije vektorskim zbrojem sila odbijanja od prepreke i sile privlačenja od strane ciljne tačke, odnosno koordinate vrha ovog vektora se dobivaju kao:

$$x_n = x_r + \Delta x_c$$
$$y_n = y_r + \Delta y_c$$

# Metoda privlačnih i odbojnih sila

- Uvodi se i *stohastička sila*.
- Ova sila ima mali intenzitet i jedino u slučaju kada dolazi do poništavanja privlačne i odbojne sile njen iznos naglo raste i ona tada djeluje na robota.





# Metoda privlačnih i odbojnih sila

- Za stohastičku silu vrijedi sljedeće pravilo:

AKO  $|F_a| = |F_r|$  I ugao između  $F_a$  i  $F_r$  je manji od 0.01 rad  
ONDA  $F_s$  velika INAČE  $F_s$  mala

- U nastavku se opisuju pravila planiranja kretanja kojima se određuju relevantne prepreke za daljnje kretanje robota u trenutnoj situaciji.
- S tim u vezi uvodimo pojmove **vidno polje**, odnosno vidokrug mobilnog robota i **ciljni krug**.
- Ovdje se koristi mobilni robot opremljen sa 16 sonarnih senzora, koji su kružno raspoređeni.
- Jasna slika, odnosno granica pouzdanosti očitavanja ovakvih podataka je 4 m od centra platforme ( $R_v$ ), odnosno vidno polje robota je krug sa polumjerom 4 metra sa centrom u centru platforme.



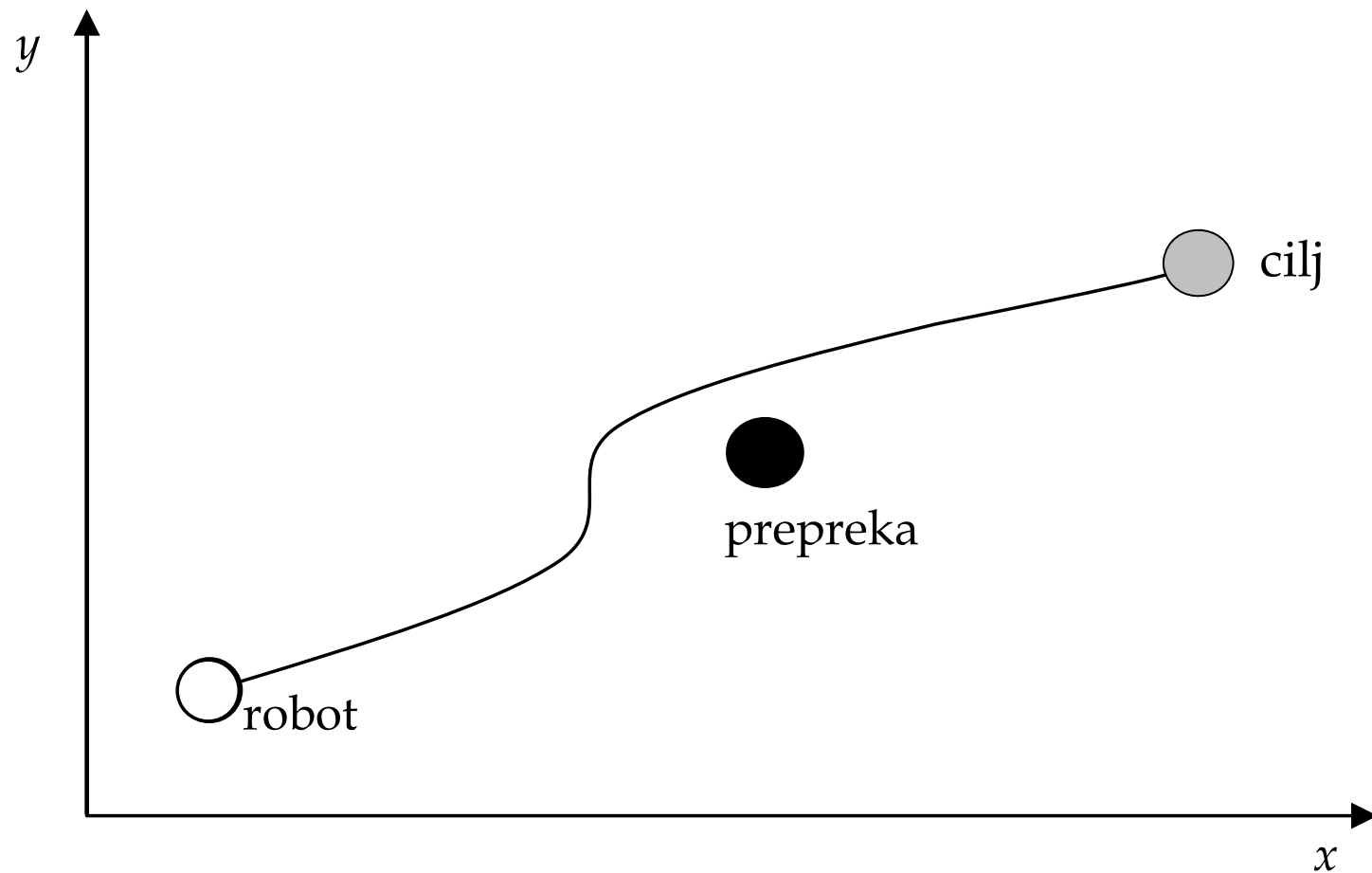
## Metoda privlačnih i odbojnih sila

- Ako u prostoru postoji samo ciljna tačka bez ikakvih prepreka, tada komponenta -  $R_p$  ima vrijednost jednaku nuli i jedina sila koja tada djeluje na robota je sila privlačenja koju uzrokuje ciljna tačka.
- U slučaju da se detektirana prepreka nalazi negdje na kraju vidnog polja robota, algoritam će pridružiti silu kojom ova prepreka odbija robota. Sila kojom prepreka odbija robota će biti relativno mala, tako da će izvršiti skoro neprimjetno skretanje robota sa njegove pravolinijske putanje.
- Kada se robot pak "previše" približi prepreci tada kvadratni član u ukupnoj sili preovladava i faktički je jedina sila koja djeluje na robota, sila odbijanja i robot se trudi da se što više udalji od prepreke.



# Metoda privlačnih i odbojnih sila

- Situacija izbjegavanja prepreke



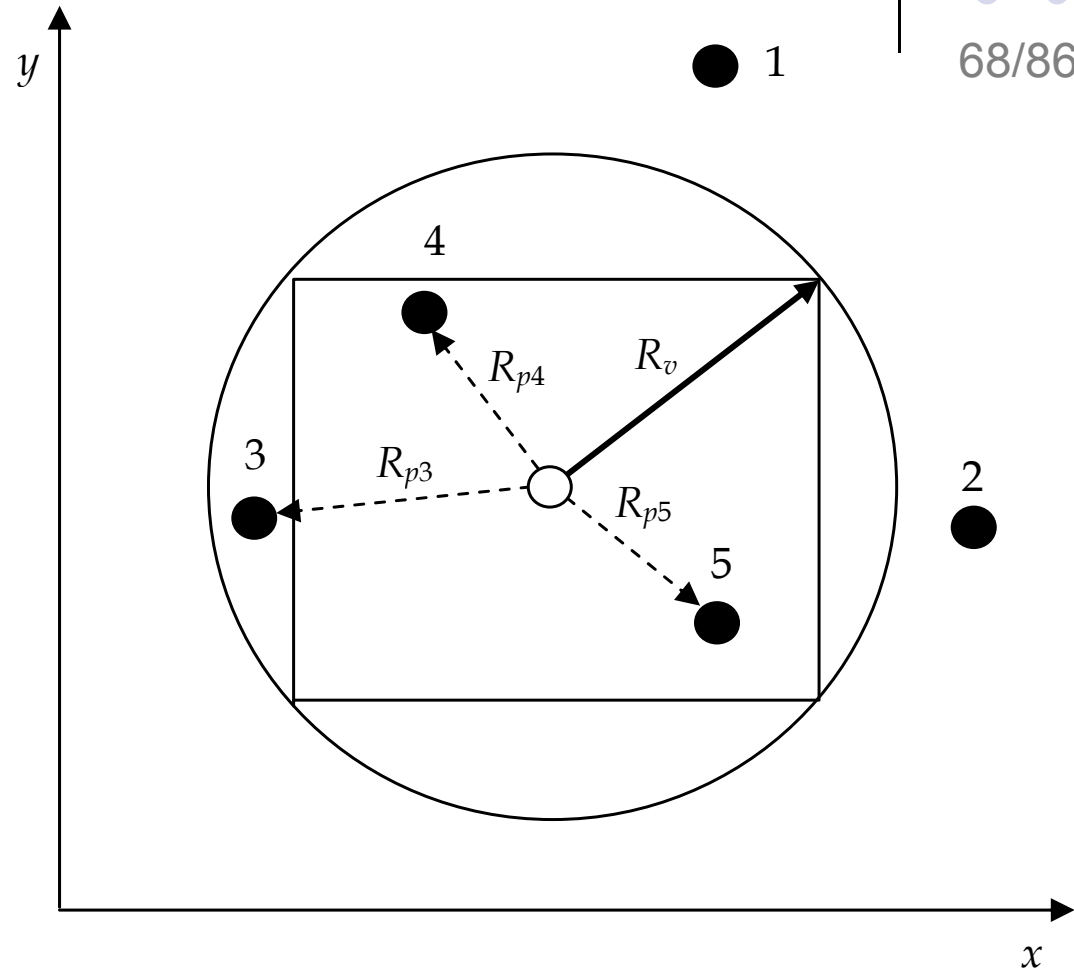
# Metoda privlačnih i odbojnih sila

- Da bi prepreka bila vidljiva mora zadovoljavati:

$$R_{pi} < R_v,$$

odnosno mora biti u vidnom polju robota, gdje je  $R_{pi}$  udaljenost  $i$ -te prepreke do robota.

- Ovdje će se koristiti prikaz vidnog polja pomoću matrice.

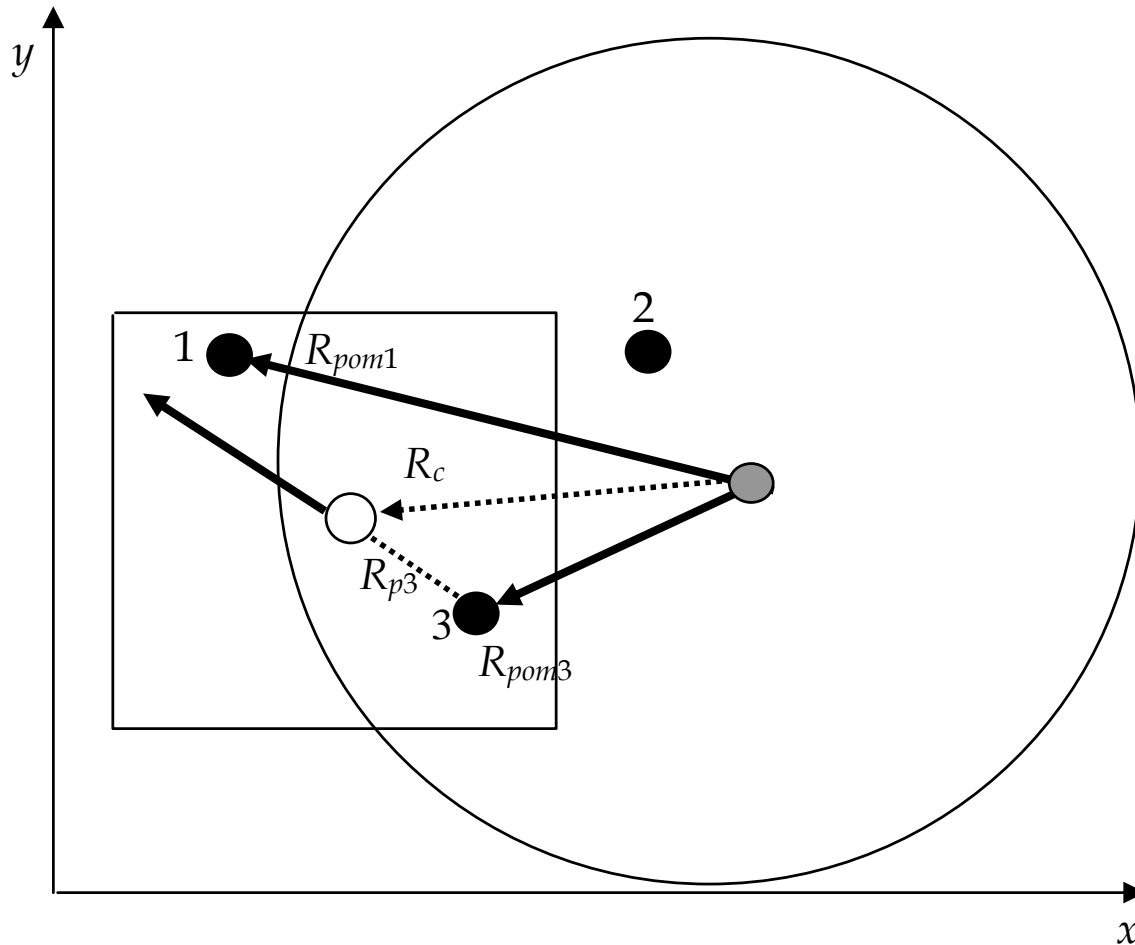


Dimenzije matrice:  $n \times n$  ( $4 \text{ m} \times 4 \text{ m}$ ),  
Dimenzije ćelije:  $5 \text{ cm} \times 5 \text{ cm}$ .



# Metoda privlačnih i odbojnih sila

- Prikaz sila koje utječu na kretanje robota na njegovom putu ka ciljnoj tački.



U ovom pristupu računa se trenutna udaljenost robota do ciljne tačke, parametar  $R_c$ , i nakon toga se opisuje kružnica sa centrom u ciljnoj tački i polumjerom koji je jednak nešto uvećanoj vrijednosti udaljenosti između robota i ciljne tačke, ( $R_c + 0.1$ , gdje duljina 0.1 predstavlja zonu sigurnosti koja se opisuje oko prepreke).



## Metoda privlačnih i odbojnih sila

- Udaljenost između prepreke i ciljne tačke:

$$R_{pom} = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2}$$

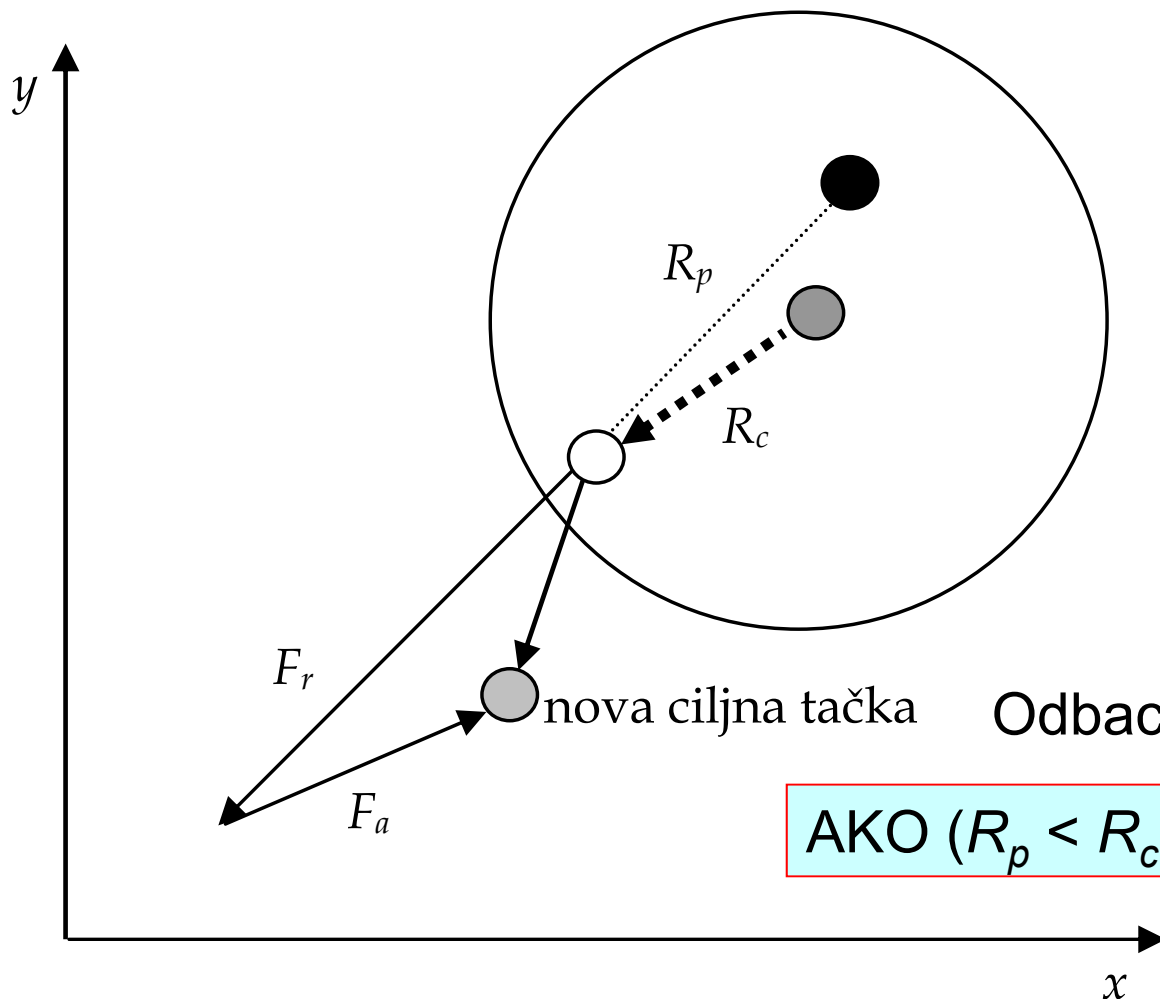
- Uvjet vidljivosti prepreke:

AKO  $((R_c + 0.1) > R_{pom}) \wedge (R_p < R_v)$  ONDA *prepreka vidljiva i utječe na kretanje.*

- Na ovaj način, sve tačke koje nisu na predviđenoj putanji, ne opterećuju nepotrebno robota sa dodatnom silom.

# Metoda privlačnih i odbojnih sila

- Prikaz situacije kada se prepreka nalazi unutar ciljnog kruga iza ciljne tačke.



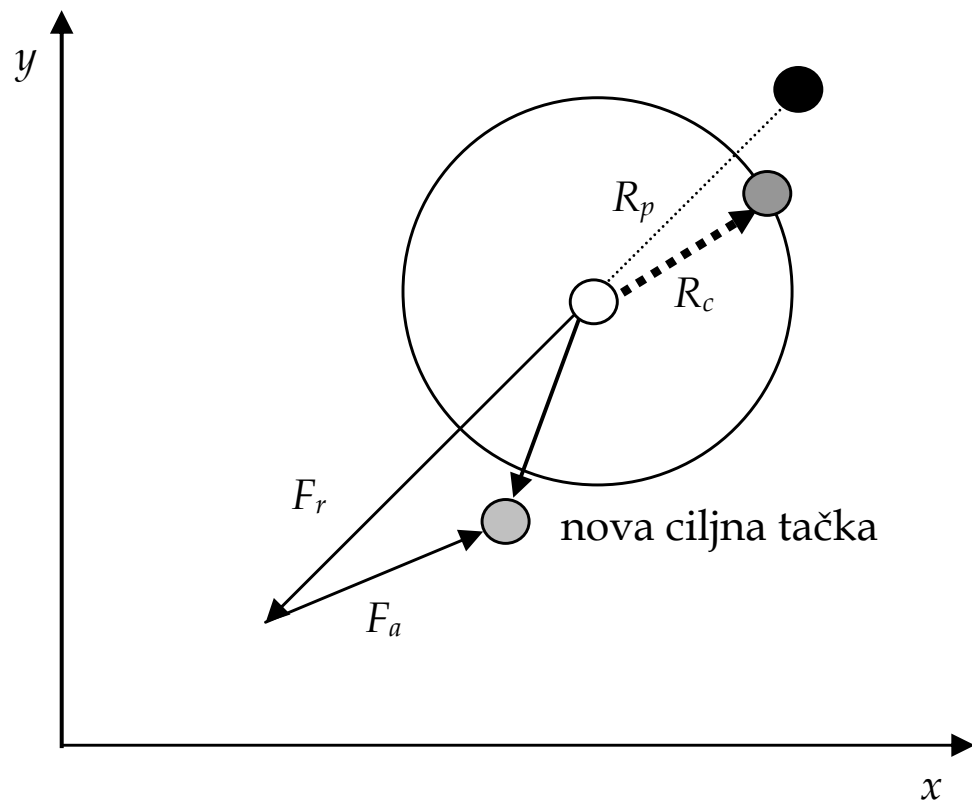
Problem može nastati ukoliko se prepreka nalazi unutar ciljnog kruga, u dijametru ( $R_c + 0.1$ ), a iza ciljne tačke i robot joj se približava te ona pada u njegovo vidno polje.

Odbacivanje prepreke iza robota:

**AKO ( $R_p < R_c$ ) ONDA prepreka vidljiva.**

# Metoda privlačnih i odbojnih sila

- Situacija kada robot ne osjeća prepreku koja se nalazi iza ciljne tačke.



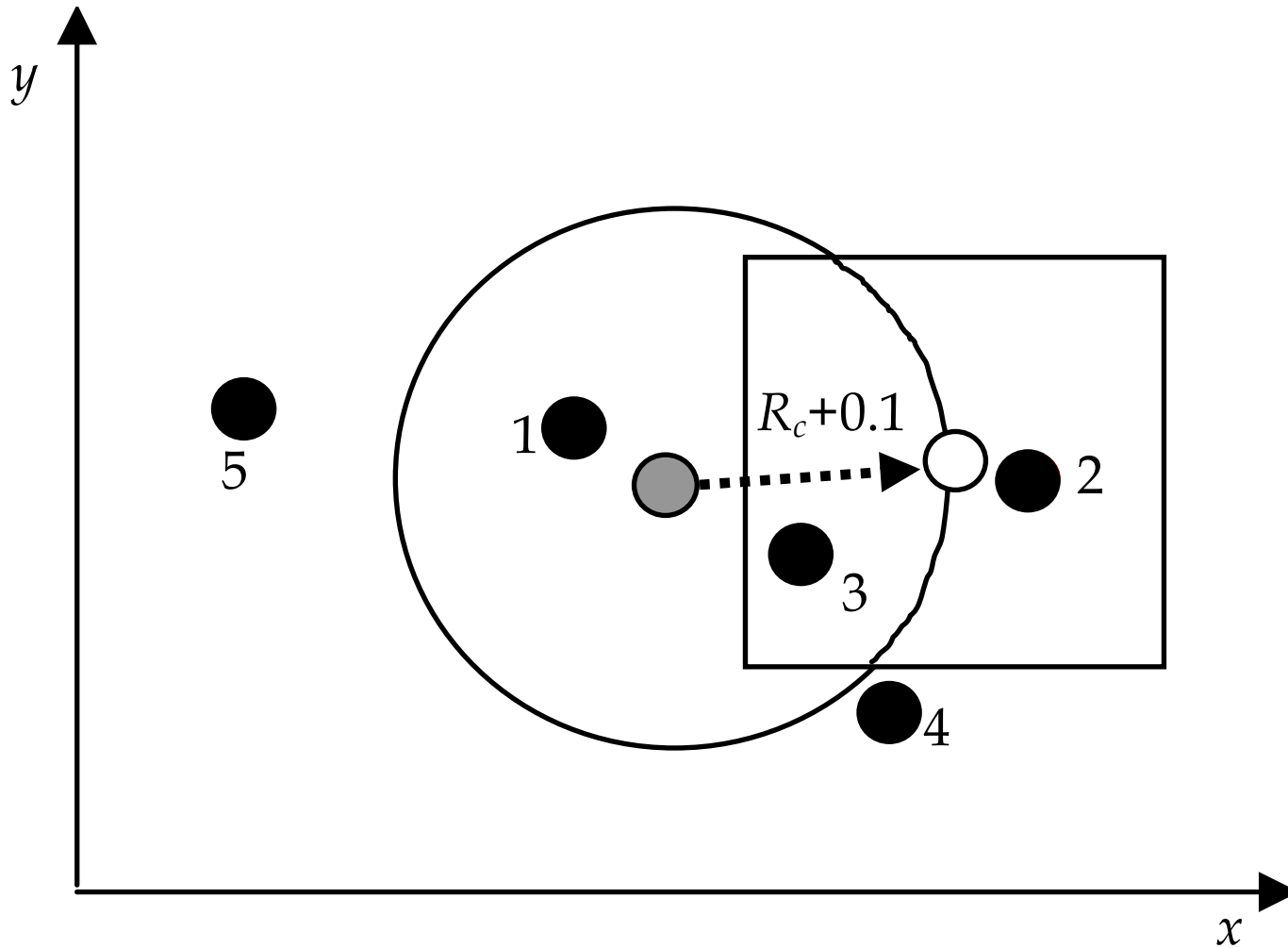
Puni uvjet za vidno polje robota glasi:

AKO  $((R_c + 0.1) > R_{pom}) \vee (R_p < R_v) \vee (R_p < R_c)$   
ONDA *prepreka vidljiva i utječe na kretanje robota*



# Metoda privlačnih i odbojnih sila

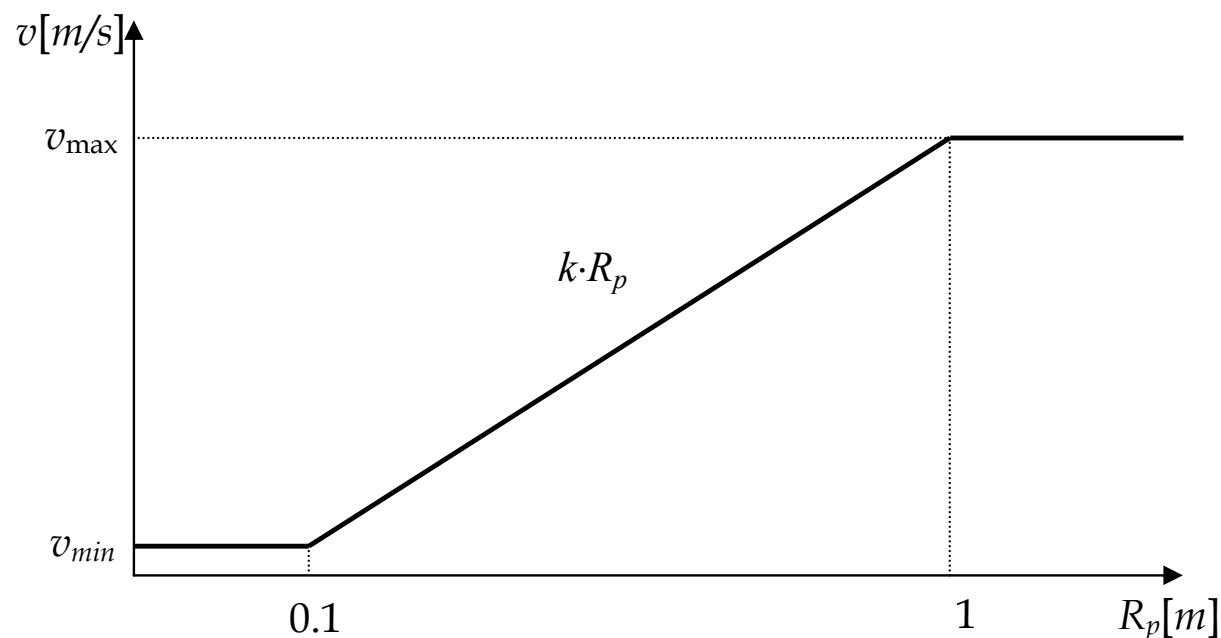
- Uvjeti globalnog pkretanja



**Samo je prepreka 3. relevantna za kretanje.**

## Metoda privlačnih i odbojnih sila

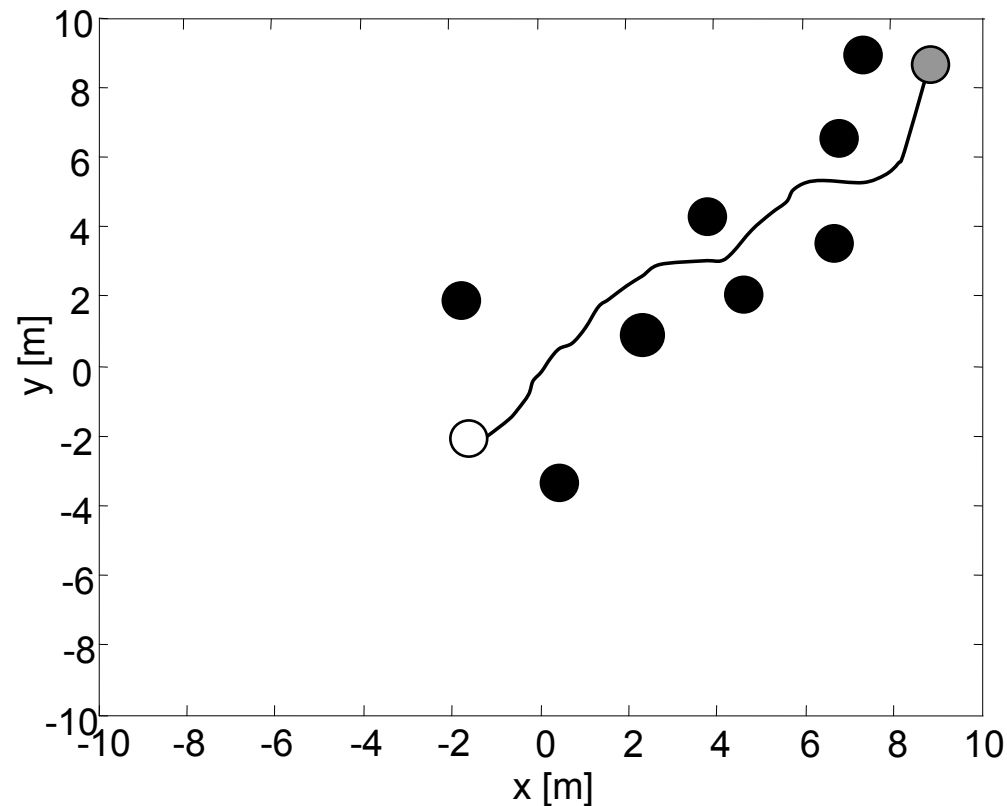
- Kada se robot kreće prema cilju i na svom putu nema prepreka koje bi mu remetile kretanja, on se tada kreće maksimalnom brzinom  $v_{max}$ .
- U slučaju da postoji relevantna prepreka na udaljenosti manjoj od 1 m od robota tada će se brzina robota korigirati koeficijentom  $k$ , tako da će se brzina robota mijenjati ovisno o njegovoj udaljenosti do prepreke, a što je prikazano na slici



# Metoda privlačnih i odbojnih sila

## Simulacijski rezultati

- Predloženi algoritam omogućuje kretanje robotu ka ciljnoj tački uz istovremeno zaobilazanje eventualnih prepreka koje mu se mogu naći na tom putu.

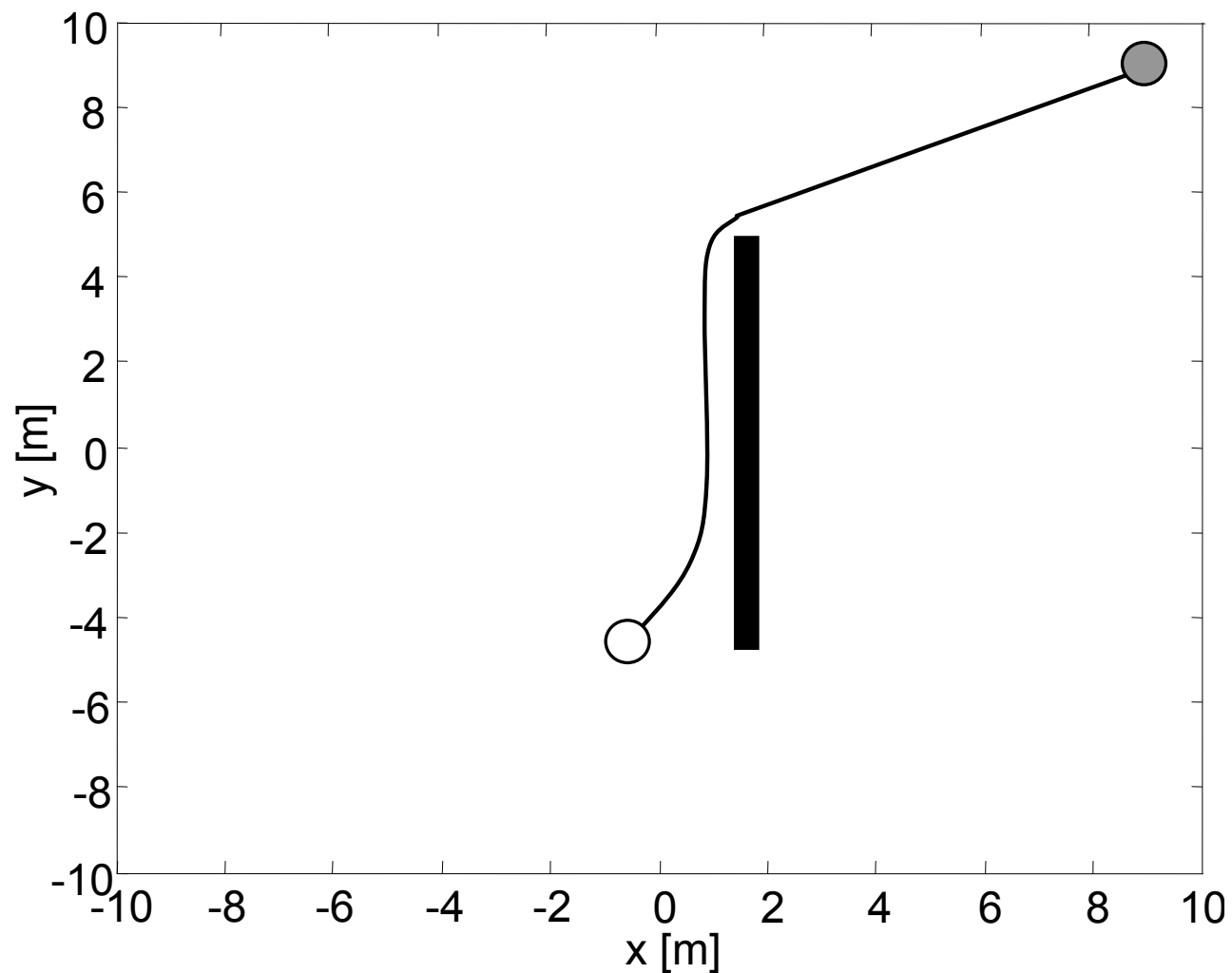


# Metoda privlačnih i odbojnih sila

## Simulacijski rezultati

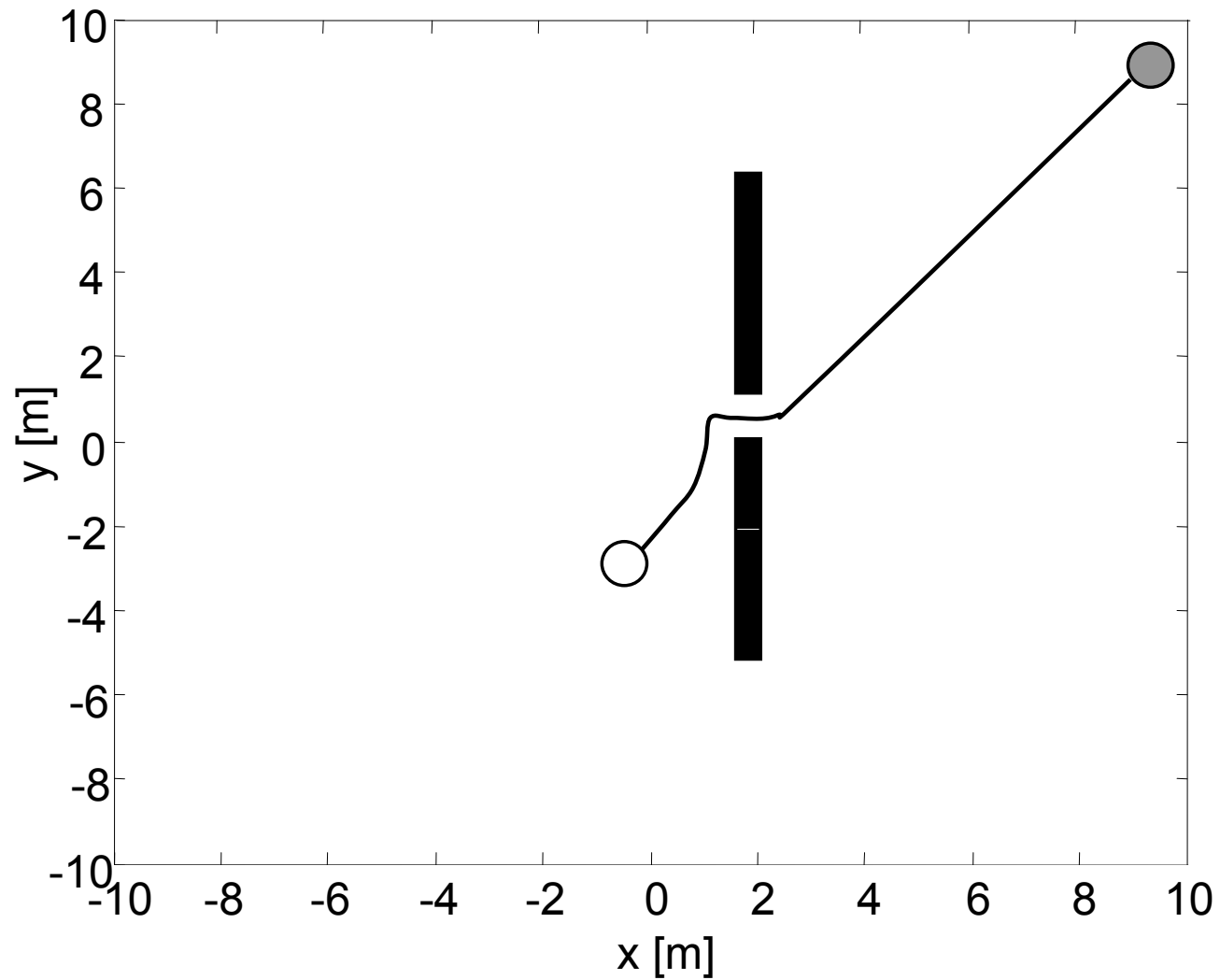


76/86



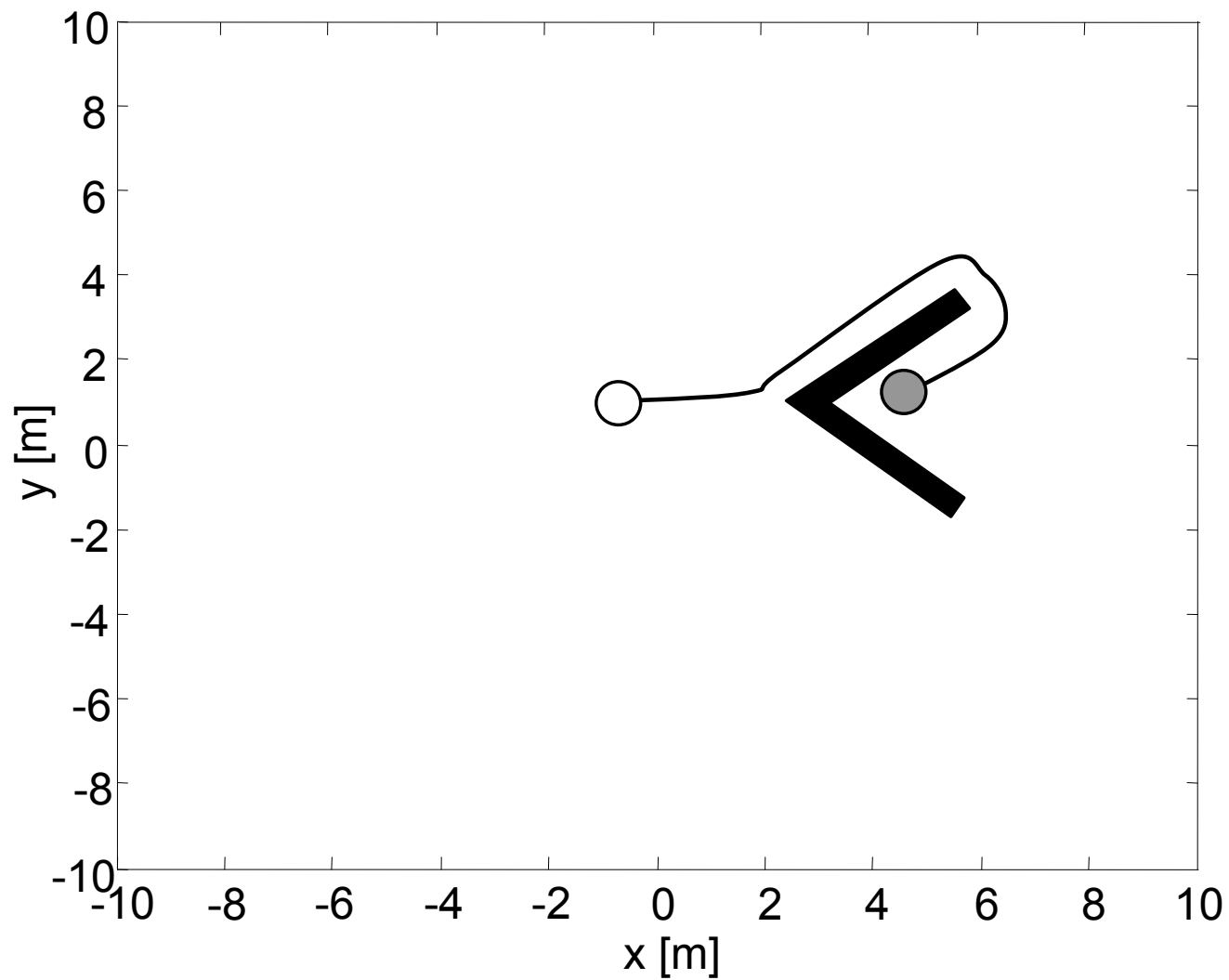
# Metoda privlačnih i odbojnih sila

## Simulacijski rezultati



# Metoda privlačnih i odbojnih sila

## Simulacijski rezultati

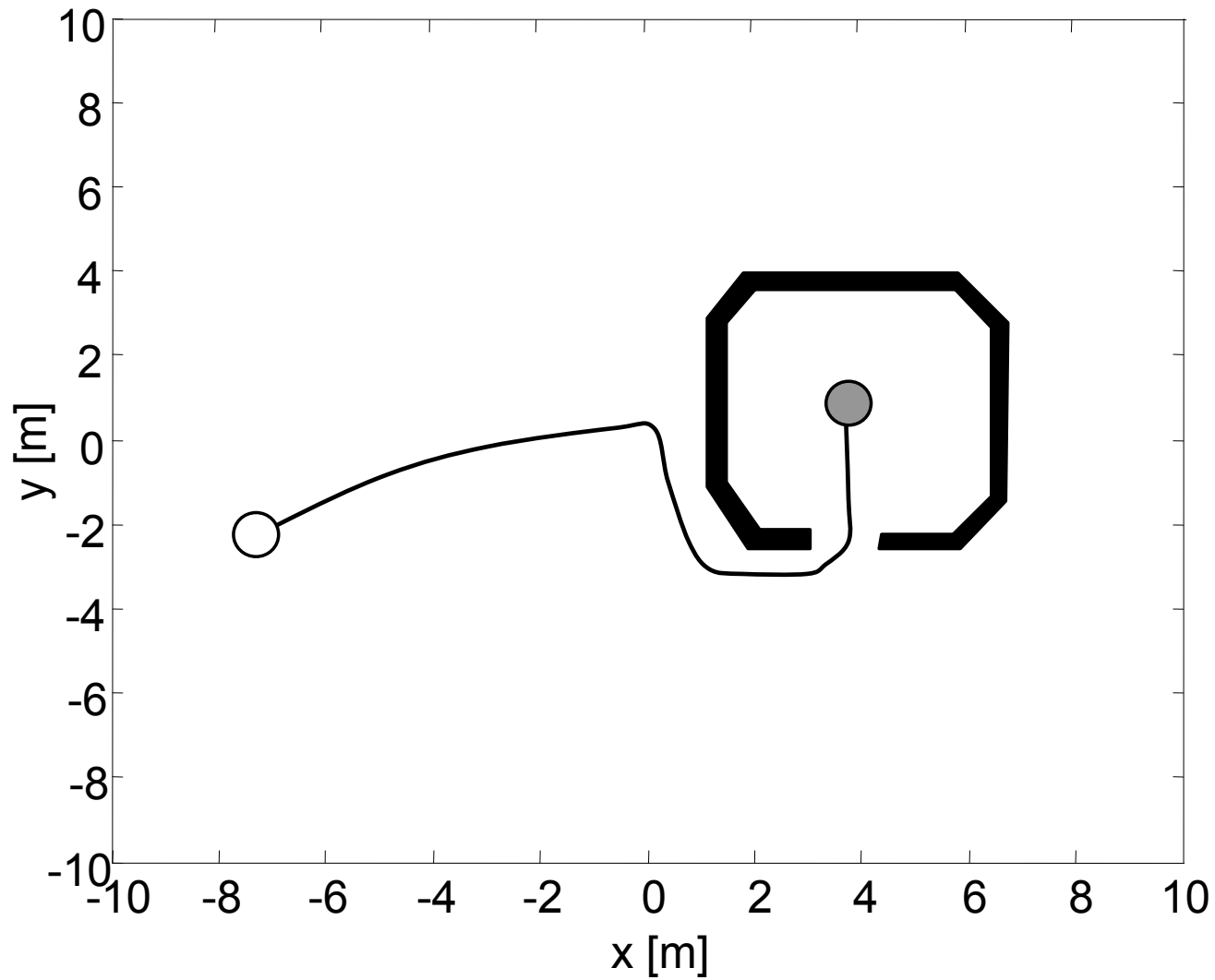


# Metoda privlačnih i odbojnih sila

## Simulacijski rezultati



79/86

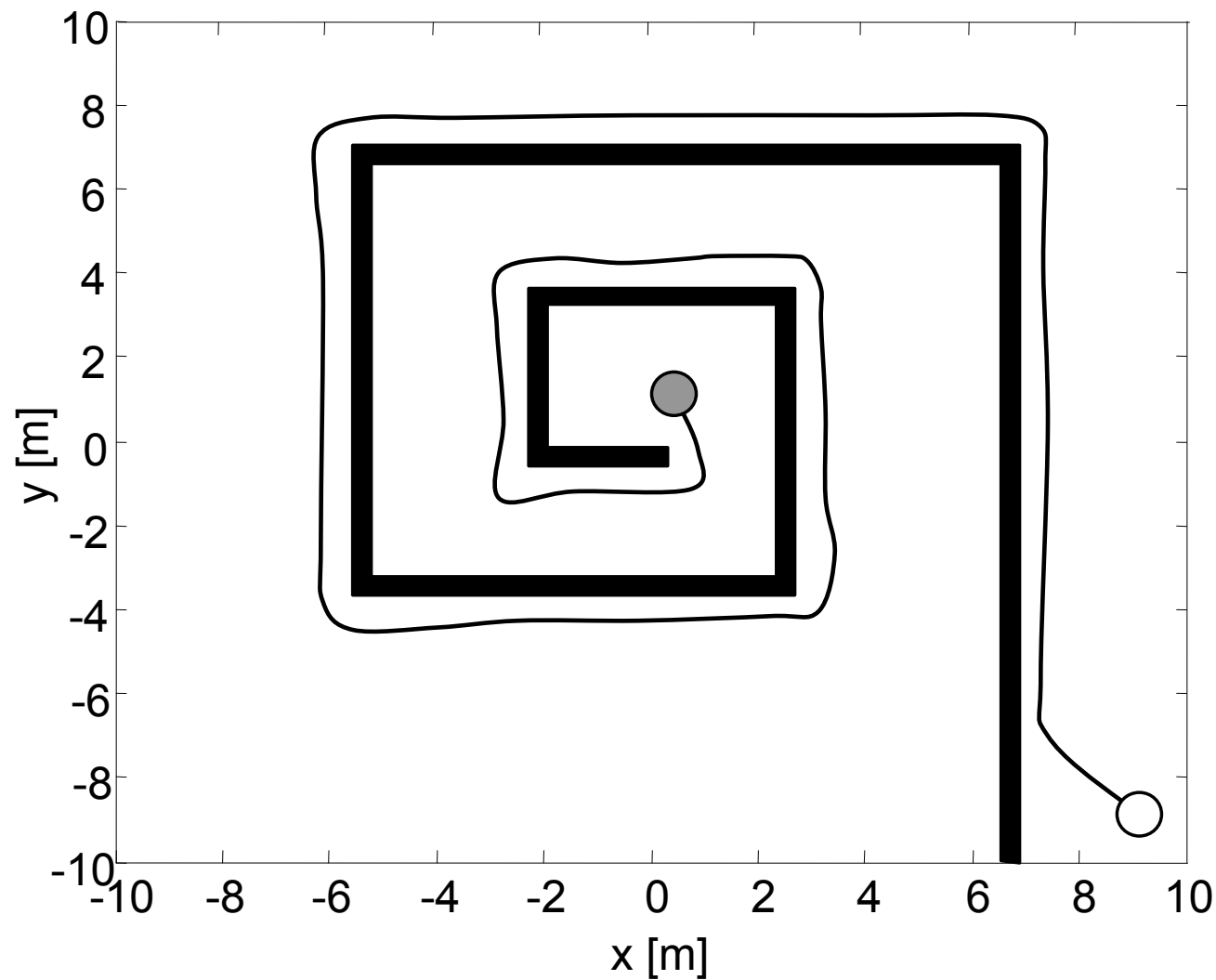


# Metoda privlačnih i odbojnih sila

## Simulacijski rezultati



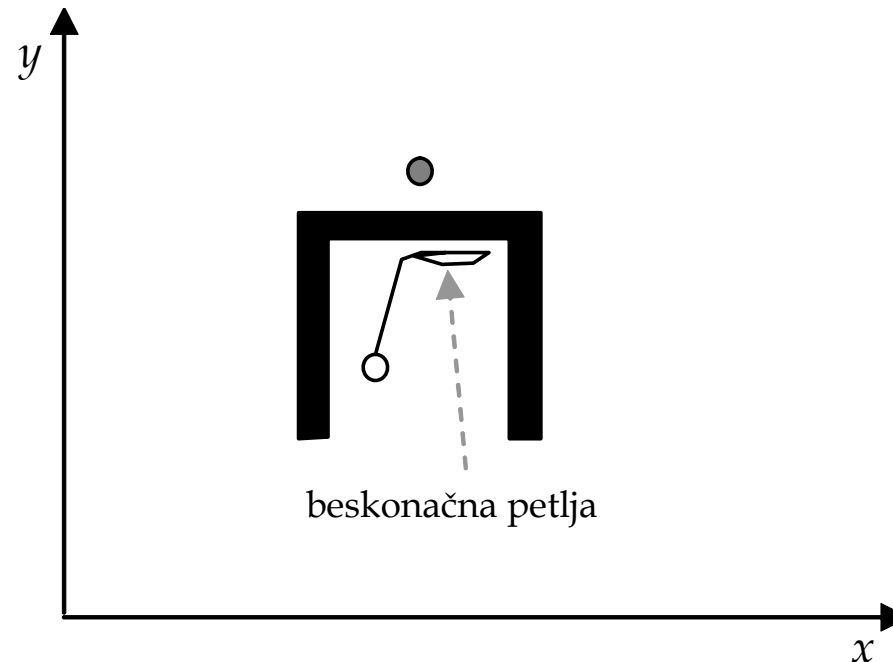
80/86





## 12.6. Problem lokalnog minimuma

- Predloženi algoritam je efikasan u slučajvima kada robot ne upada u zamku, odnosno kada ne postoji problem lokalnog minimuma.
- Međutim, kada se robot nađe u lokalnom minimumu ne uspijeva izaći iz zamke (vrti se u beskonačnoj petlji).



# Problem lokalnog minimuma

- Za izbjegavanje problema lokalnog minimuma predlaže se algoritam koji ne zahtijeva memoriranje prethodno viđenih (pređenih) dijelova okoline [Velagić, Lačević i Peruničić, 2006].



**Algoritam** izbjegavanja problema lokalnog minimuma

**Korak 1.:** *Računati privlačne i odbojne sile, te njihovu sumu*

**Korak 2.:** **AKO** *rezultantna sila ima tendenciju gurnuti robota unazad*

**ONDA** *ne uzimati u obzir privlačnu silu*

*I držati odbojnu silu konstantnom i jednakom udaljenosti do najbliže prepreke*

**INAČE** *primijeniti princip privlačnih i odbojnih sila*

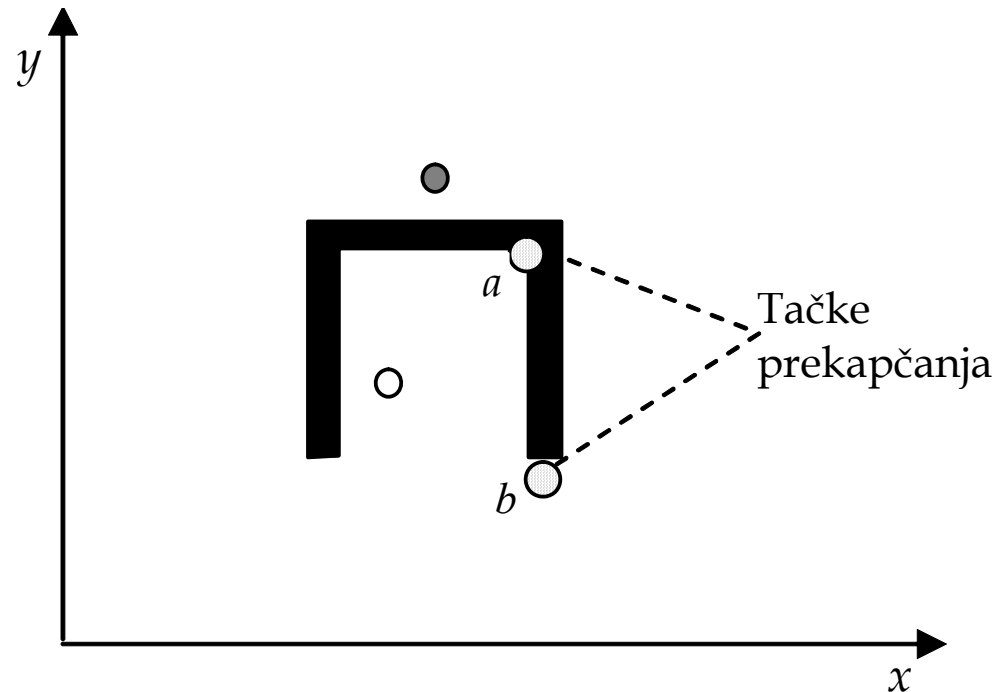
**KRAJ**

**Korak 3.:** **AKO** *robot nije dosegnuo ciljnu tačku*

**ONDA** *idi na Korak 1*

# Problem lokalnog minimuma

- Algoritam tjera robota u desno u **tački prekapčanja  $a$**  (tačka u kojoj robot ima tendenciju vratiti se nazad, odnosno promijeniti smjer kretanja) i vodi ga do druge **tačke prekapčanja  $b$**  u skladu sa opisanim algoritmom izbjegavanja lokalnog minimuma.
- Nakon pristizanja u tačku  $b$  primjenjuje se princip privlačnih i odbojnih sila, opisan ranije.



# Problem lokalnog minimuma

## Simulacijski rezultati



84/84

