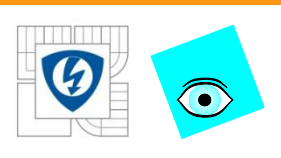


Plánování pohybu mobilního robotu a skupiny mobilních robotů

Dr. rer. nat. Martin Saska (ČVUT v Praze)

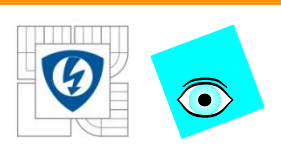
4.března 2011

Tato prezentace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky.



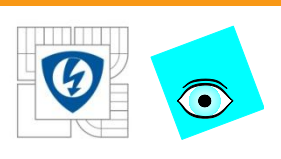
Konfigurační prostor

- Konfigurace robotu v prostoru
 - Zredukovaný popis stavu robotu
- Konfigurační prostor
 - Množina všech možných konfigurací
- Cesta
 - Souvislá posloupnost konfigurací od počátečního do konečného stavu
- Překážka
 - Zakázané konfigurace
- Volný konfigurační prostor
 - Povolené konfigurace



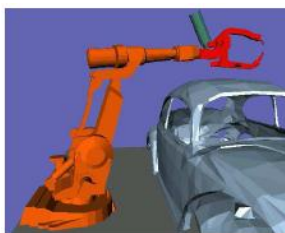
Konfigurační prostor

- Konfigurace robotu
 - Jednoznačně definuje polohu robotu vzhledem k souřadné soustavě
- Příklady
 - x, y - válcový robot pohybující se v rovině - 2 DOF
 - x, y, θ - robotické auto, navigace - 3 DOF
 - x, y, θ, β - traktor s valníkem - 4 DOF
 - $x, y, z, \theta, \beta, \gamma$ létající robot - vrtulník stíhačka, raketoplán 6 DOF
 - α, β manipulátor se dvěma klouby - 2 DOF



Konfigurační prostor

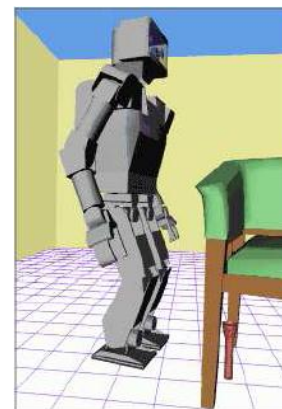
- Konfigurace robotu s různým stupněm volnosti



6 DOF



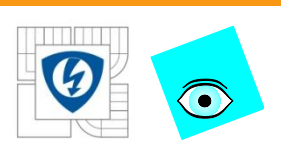
12 DOF



40 DOF

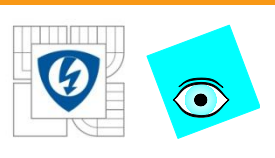


65-120 DOF



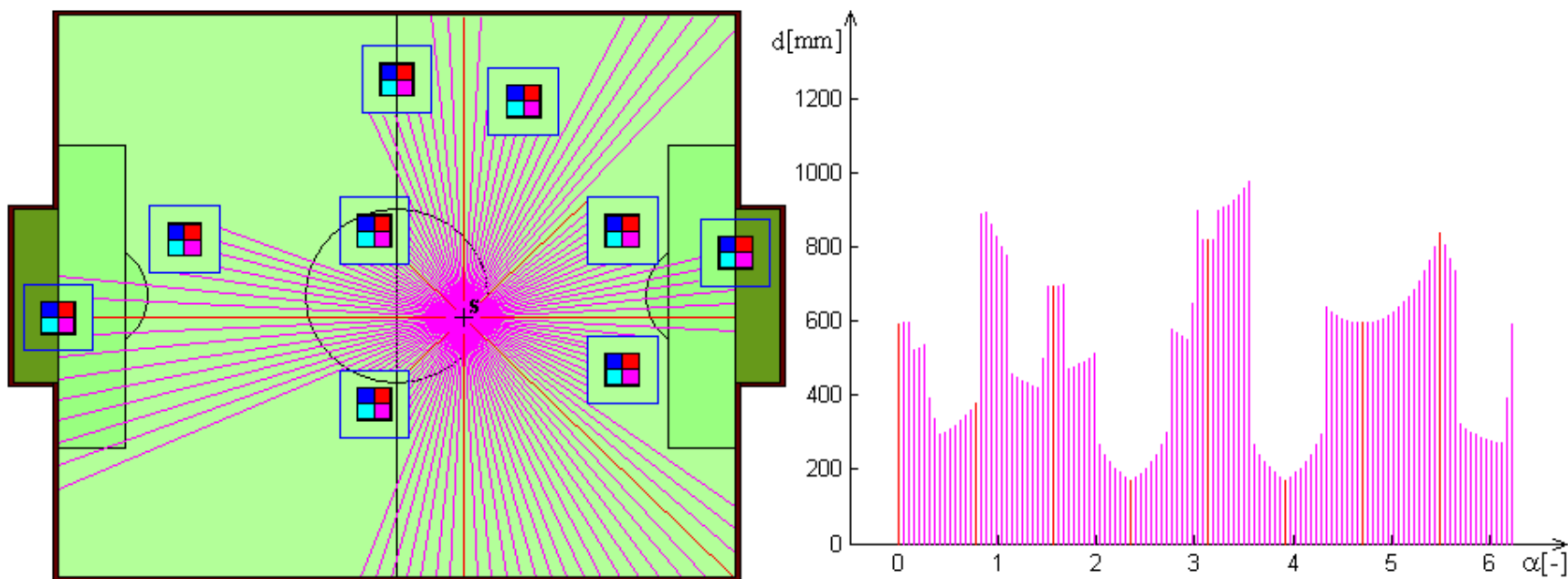
Plánování pohybu mobilního robotu

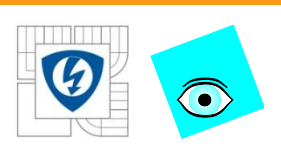
- Lokální neboli reaktivní metody
 - Histogram vektorového pole (Vector field histogram)
 - Potenciálová pole (Potential field)
- Globální metody
 - Mřížka obsazení (Occupancy grid)
 - Graf viditelnosti (Visibility graph)
 - Voroného diagram (Voronoi diagram) } Road-map
 - Buněčná dekompozice (Cell decomposition)
 - Rychle náhodně rostoucí stromy (Rapidly-exploring Random Tree - RRT)



Histogram vektorového pole

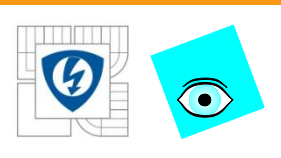
- Vector Field Histogram (VFH)
 - Původně navigační mechanismus pro roboty opatřené sonarem
 - Histogram tvoří vzdálenosti od robotu k překážkám





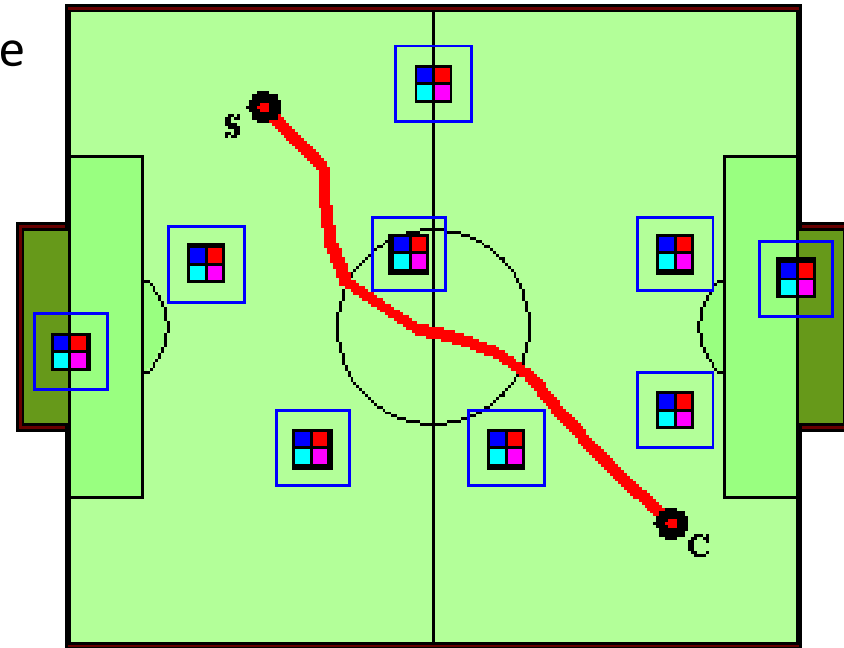
Histogram vektorového pole

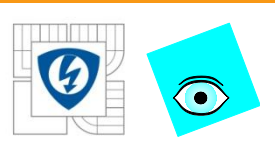
- Nejjednodušší VFH algoritmus:
 1. Opakuj dokud vzdálenost aktuálního stavu q_n a požadovaného stavu q_G je menší než ε :
 - a. Vytvoř histogram H z aktuální pozice robotu
 - b. Vytvoř redukovaný histogram H_{red} z vektorů obsažených v H , jejichž velikost je větších než práh λ
 - c. Vyber vektor w z H_{red} , který svírá s přímkou procházející aktuální a požadovanou polohou robotu nejmenší úhel
 - d. Pohni robotem o definovaný krok ve směru vektoru w
 2. Vygeneruj nový požadovaný cíl pohybu



Histogram vektorového pole

- Výhody a nevýhody histogramu vektorového pole
 - Uvážnutí v lokálním minimu (**U** překážky)
 - Citlivý na nastavení parametrů
 - + Jednoduchý výpočet i implementace
 - + Rychlá odezva na změny v dynamickém prostředí

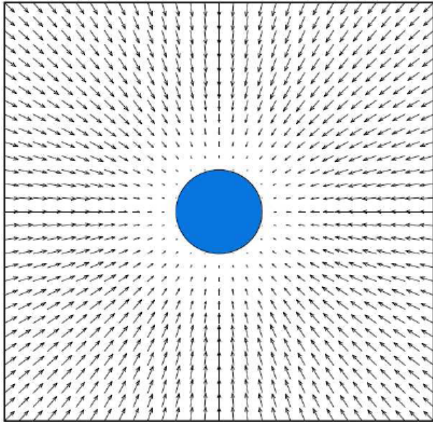




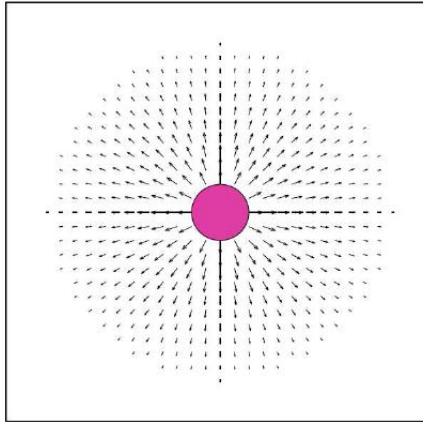
Potenciálové pole

- **Potenciálové pole**

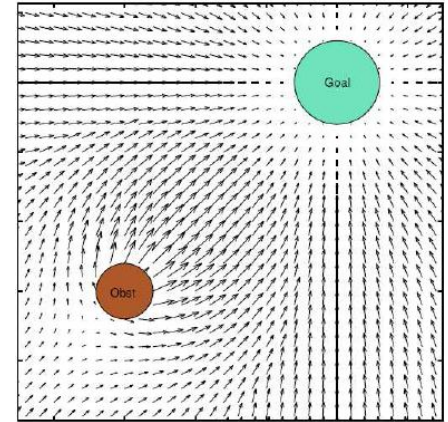
- Konfigurační prostor robotu popsán funkcí jejíž záporný gradient v libovolném bodě určí požadovaný směr pohybu robotu
- Zajištění nekolizního pohybu v dostatečné vzdálenosti od překážek
- Dosažení požadované pozice robotu v pokud možno nejkratším čase



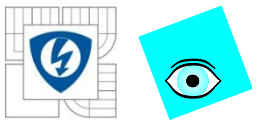
Přitažlivá složka



Odpudivá složka



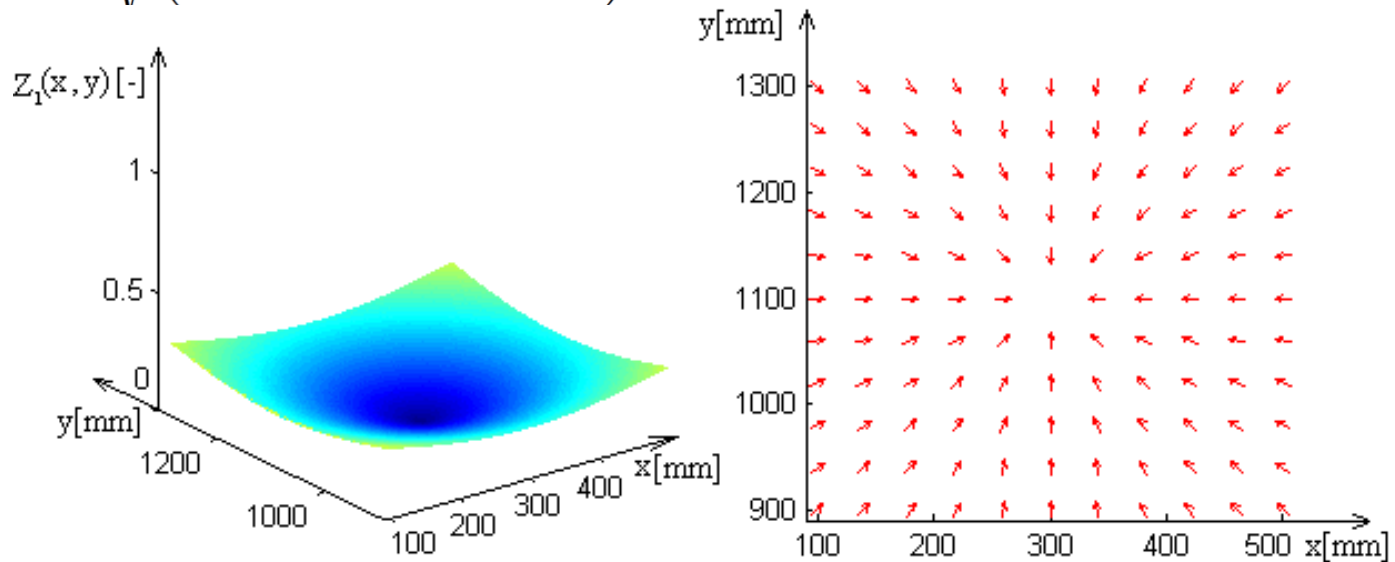
Kombinace obou sil

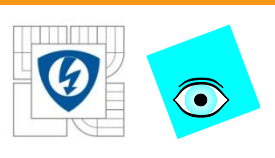


Potenciálové pole

- Atraktivní složka potenciálového pole
 - Přitahuje robot do požadovaného stavu
 - Gradient funkce má konstantní velikost v celém pracovním prostoru

$$Z_1(x, y) = \sqrt{(x - C_x)^2 + (y - C_y)^2}$$

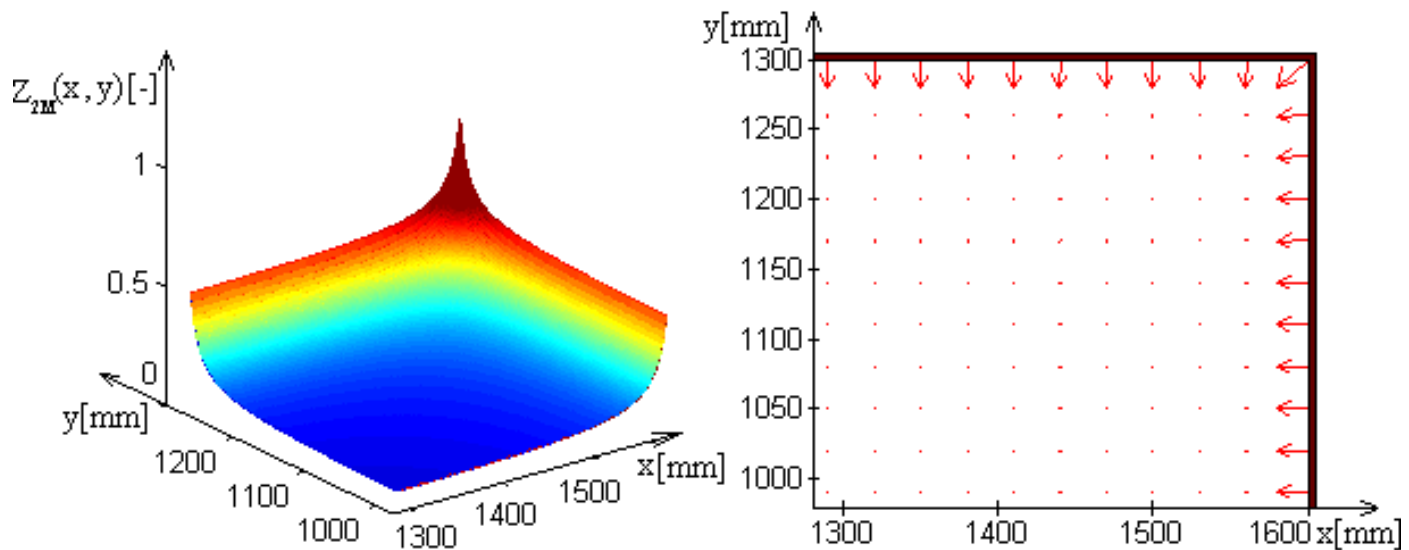


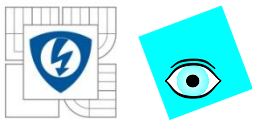


Potenciálové pole

- Repulsivní složka potenciálového pole – mantinel
 - Zabraňuje kolizi robotu s okrajem hrací plochy

$$Z_{2M} = c_1 \left(\frac{1}{x - A_x} - \frac{1}{x - B_x} + \frac{1}{y - A_y} - \frac{1}{y - B_y} \right)$$



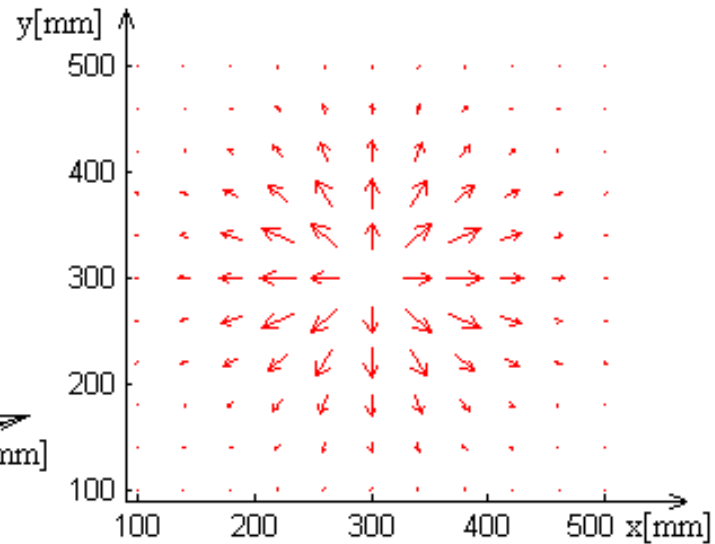
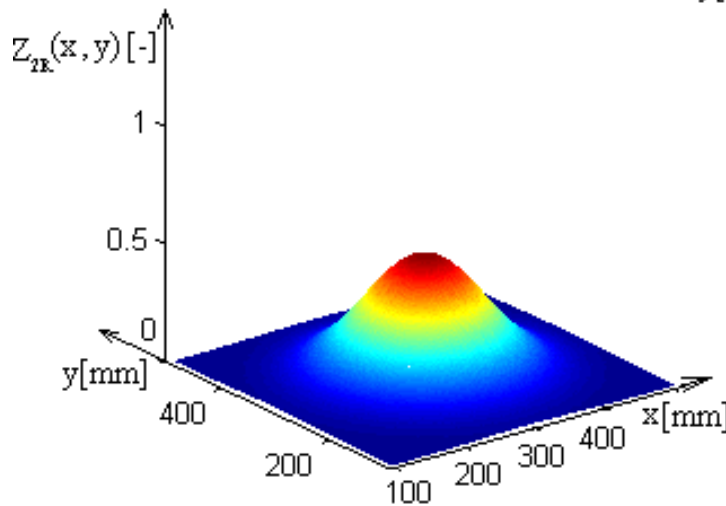


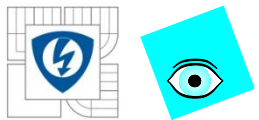
Potenciálové pole

- Repulsivní složka potenciálového pole – roboty
 - Zabraňuje kolizi robotu s ostatními hráči
 - Směr od středu překážek a zanedbatelný vliv pole ve větší vzdálenosti

$$Z_{2R} = \sum_{i=1}^N Z_{2R_j}$$

$$Z_{2R_j} = c_2 \exp \left(-c_3 \left((x - P_x)^2 + (y - P_y)^2 \right) \right)$$

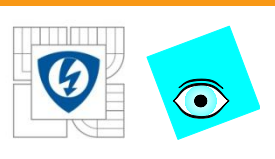




Potenciálové pole

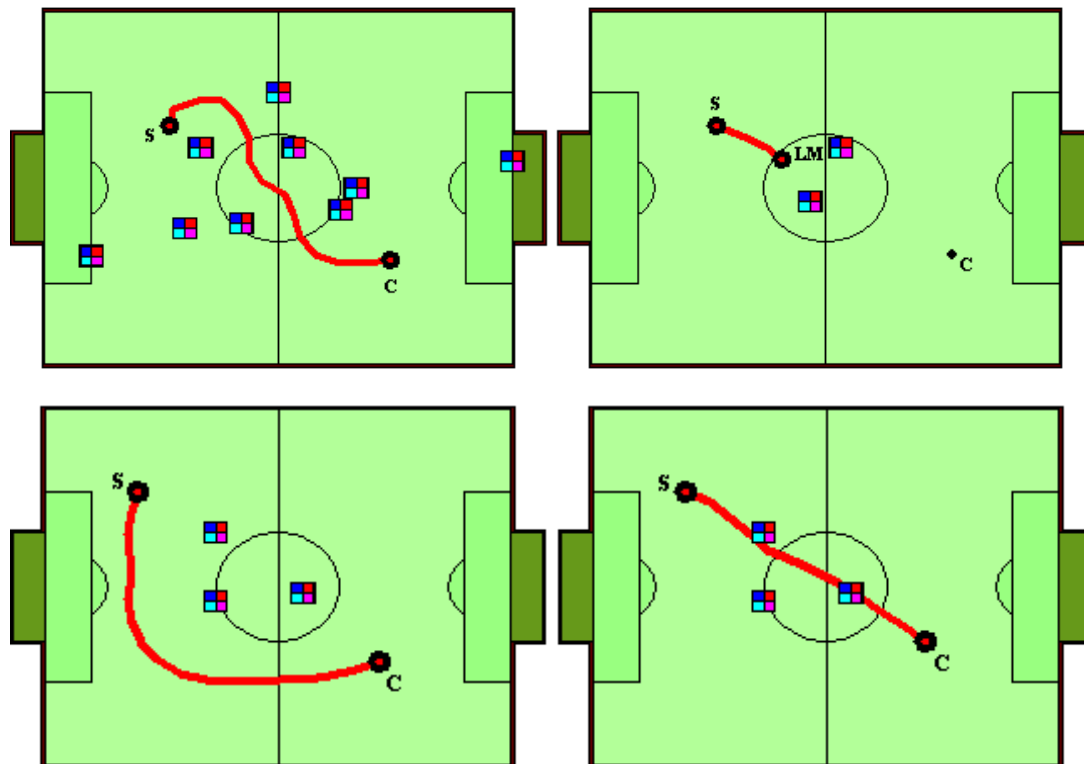
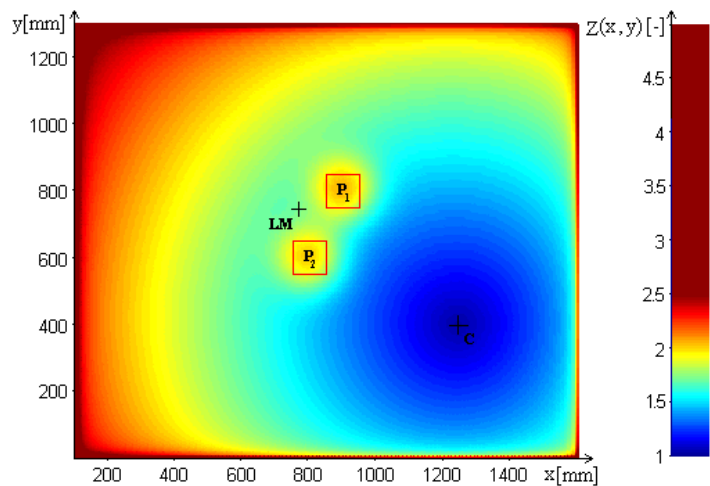
- Kombinace všech složek

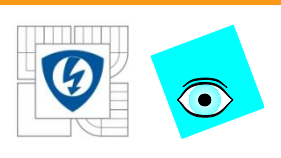
$$\varphi(x, y) = -grad(Z_1(x, y) + Z_2(x, y)) = \frac{(x - C_x; y - C_y)}{\sqrt{((x - C_x)^2 + (y - C_y)^2)}} +$$
$$c_1 \left(-\frac{1}{(x - A_x)^2} + \frac{1}{(x - B_x)^2} - \frac{1}{(y - A_y)^2} + \frac{1}{(y - B_y)^2} \right) -$$
$$2c_2c_3 \sum_{j=1}^N \exp \left(-c_3 \left((x - P_x)^2 + (y - P_y)^2 \right) \right) (x - P_x; y - P_y)$$



Potenciálové pole

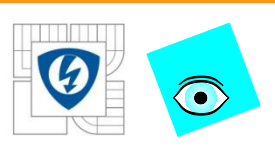
- Kombinace všech složek





Potenciálové pole

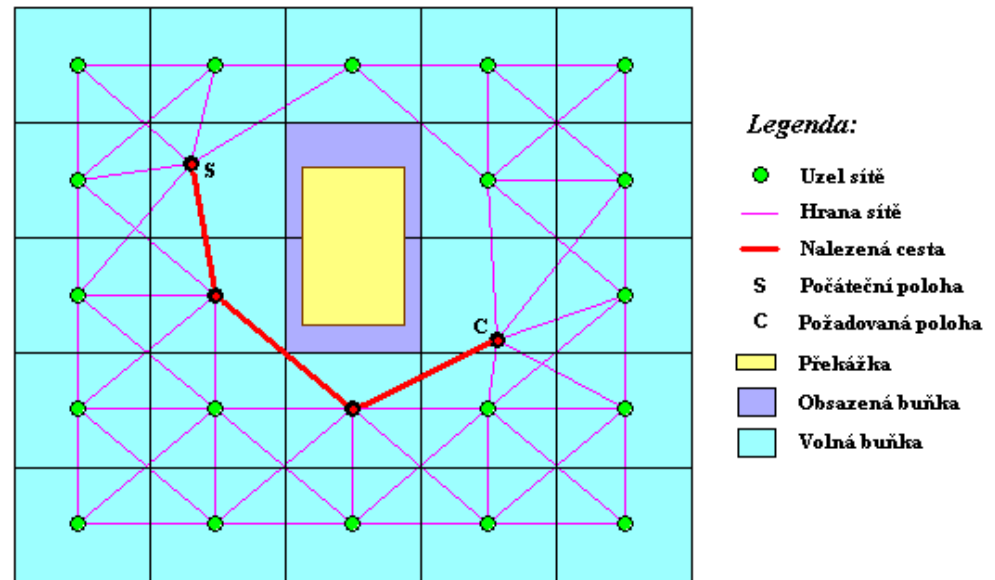
- Výhody a nevýhody potenciálového pole
 - Uvážnutí v lokálním minimu
 - Citlivé na nastavení parametrů
 - + Jednoduchý výpočet
 - + Rychlá odezva na změny v dynamickém prostředí
- Řešení problému lokálních minim
 - Harmonický potenciál
 - Metody simulovaného žíhání
 - Kombinace s Voroného diagramem

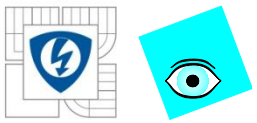


Mřížka obsazení

- Occupancy Grid (OG)

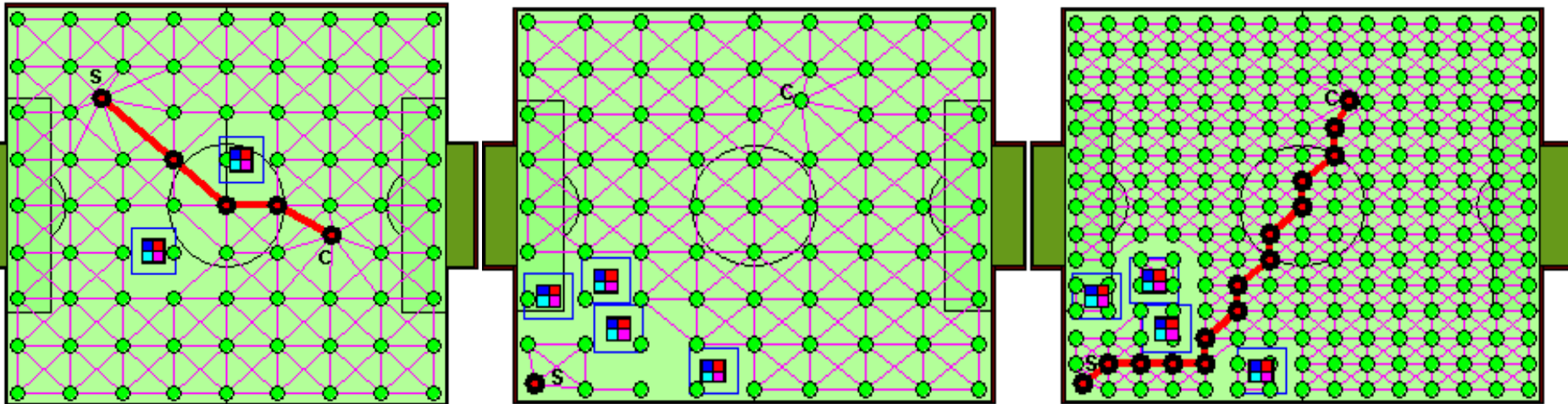
- Celý konfigurační prostor robotu je rozdělený na disjunktní oblasti
- Obsazená buňka: oblast do které zasahuje překážka
- Volná buňka: oblast do které nezasahuje překážka
- Uzly grafu: středy volných buněk
- Hrany grafu: spojují sousedící volné buňky
- Řešení plánování cesty: nejkratší cesta v grafu z počáteční do cílové polohy robotu





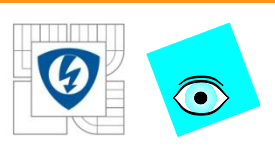
Mřížka obsazení

- Výhody a nevýhody mřížky obsazení
 - Nalezení řešení není zaručeno
 - Citlivé na velikost mřížky
 - + Jednoduchá implementace
 - + Nezávislé na typu mapy či senzorových dat



4.3.2011

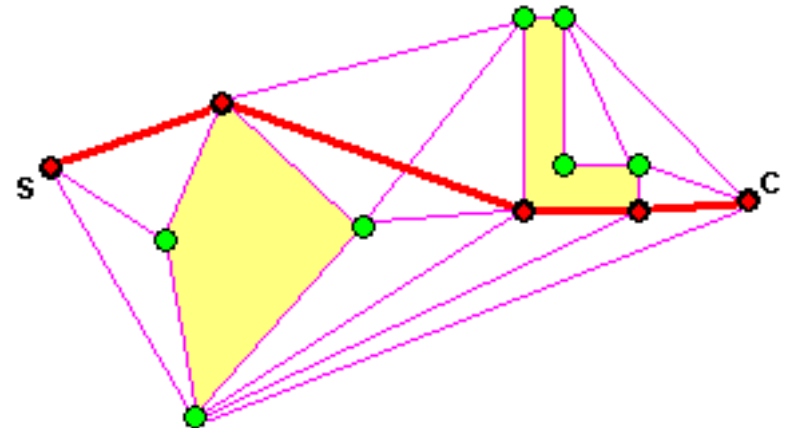
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

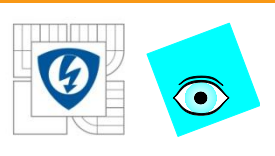


Graf viditelnosti

- Visibility Graph (VG)

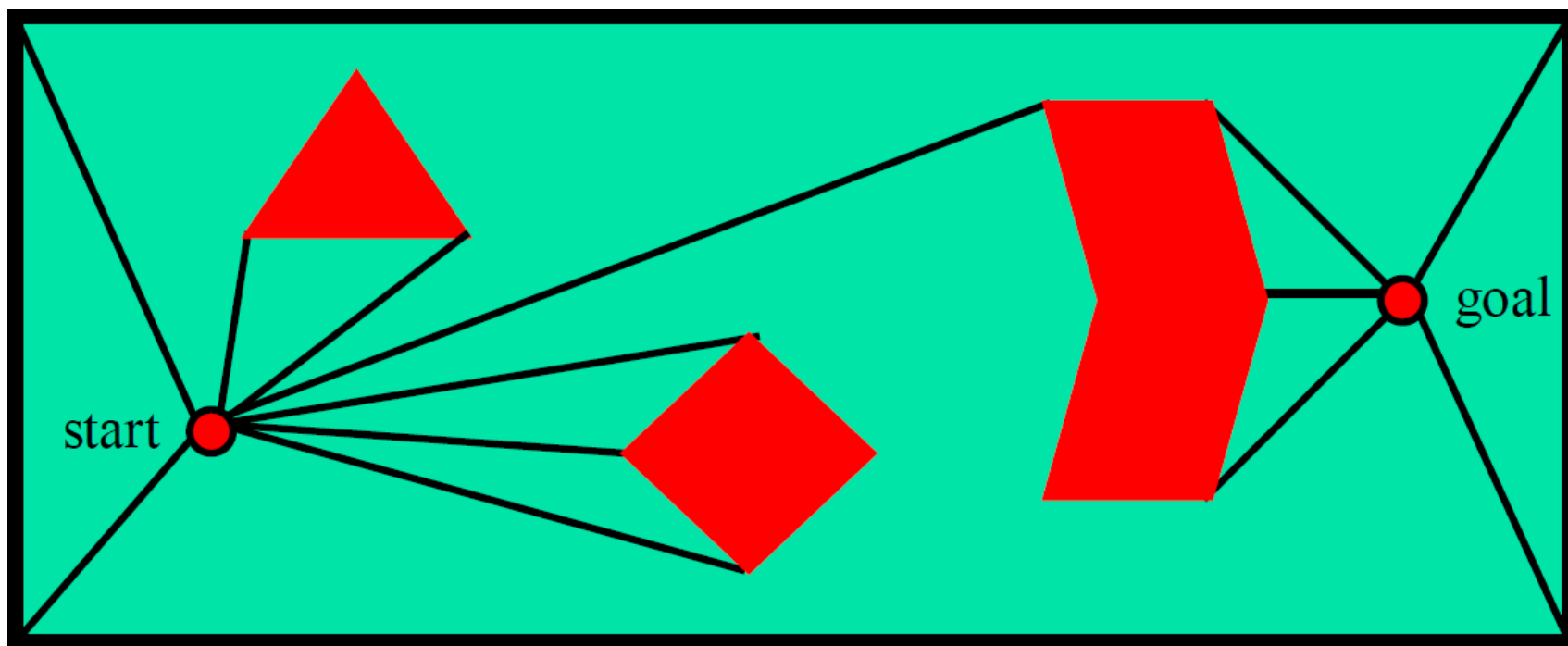
- Založený na hledání nejkratší cesty v grafu
- Uzly grafu viditelnosti: body S a C , vrcholy všech polygonů aproximujících jednotlivé překážky.
- Hrany grafu: spojují všechny dvojice uzlů, mezi kterými je přímá viditelnost -> na úsečce mezi nimi neleží překážka
- Nejkratší cesta mezi uzly S a C
 - A^* , Dijkstrův algoritmus...

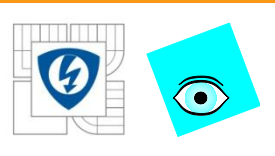




Graf viditelnosti

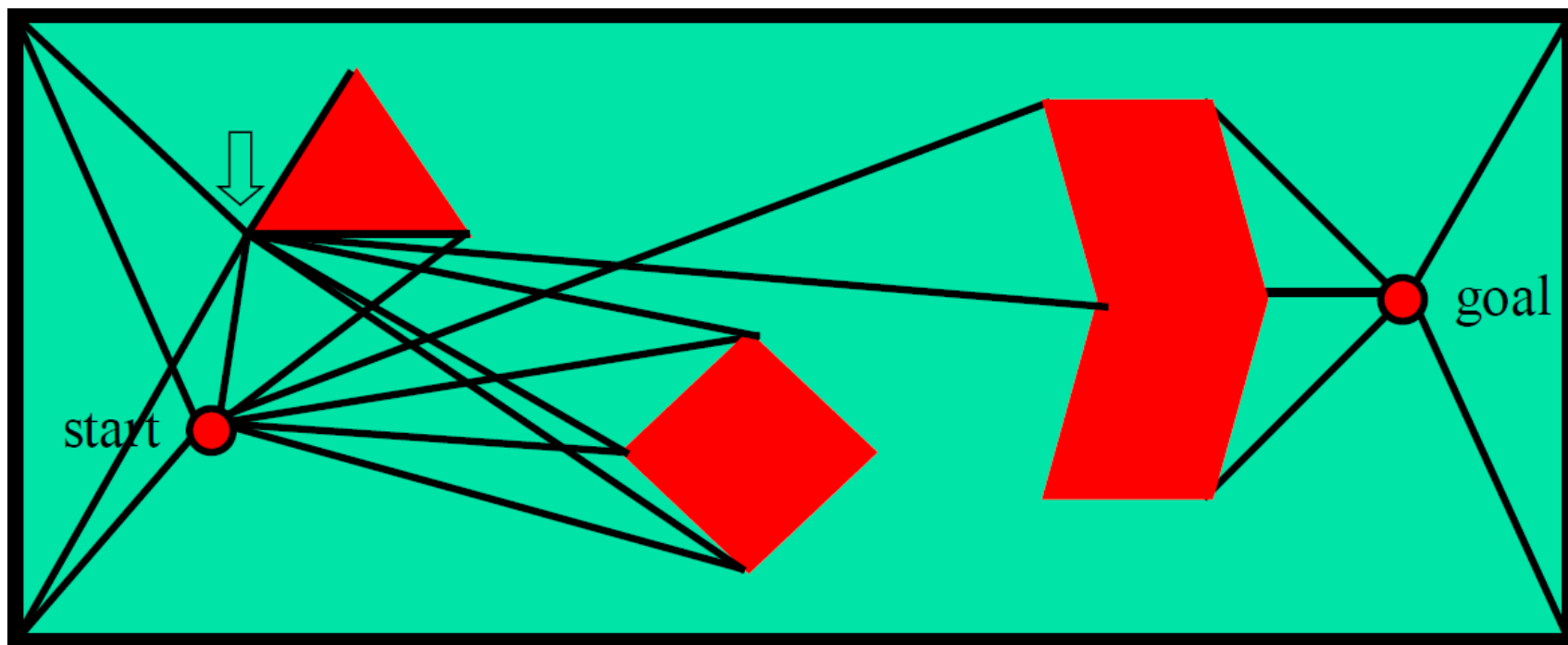
- Příklad konstrukce grafu viditelnosti
 - Nejprve vykreslí hrany viditelnosti z konfigurací S a C

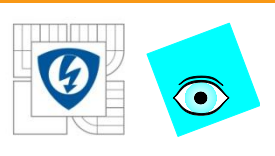




Graf viditelnosti

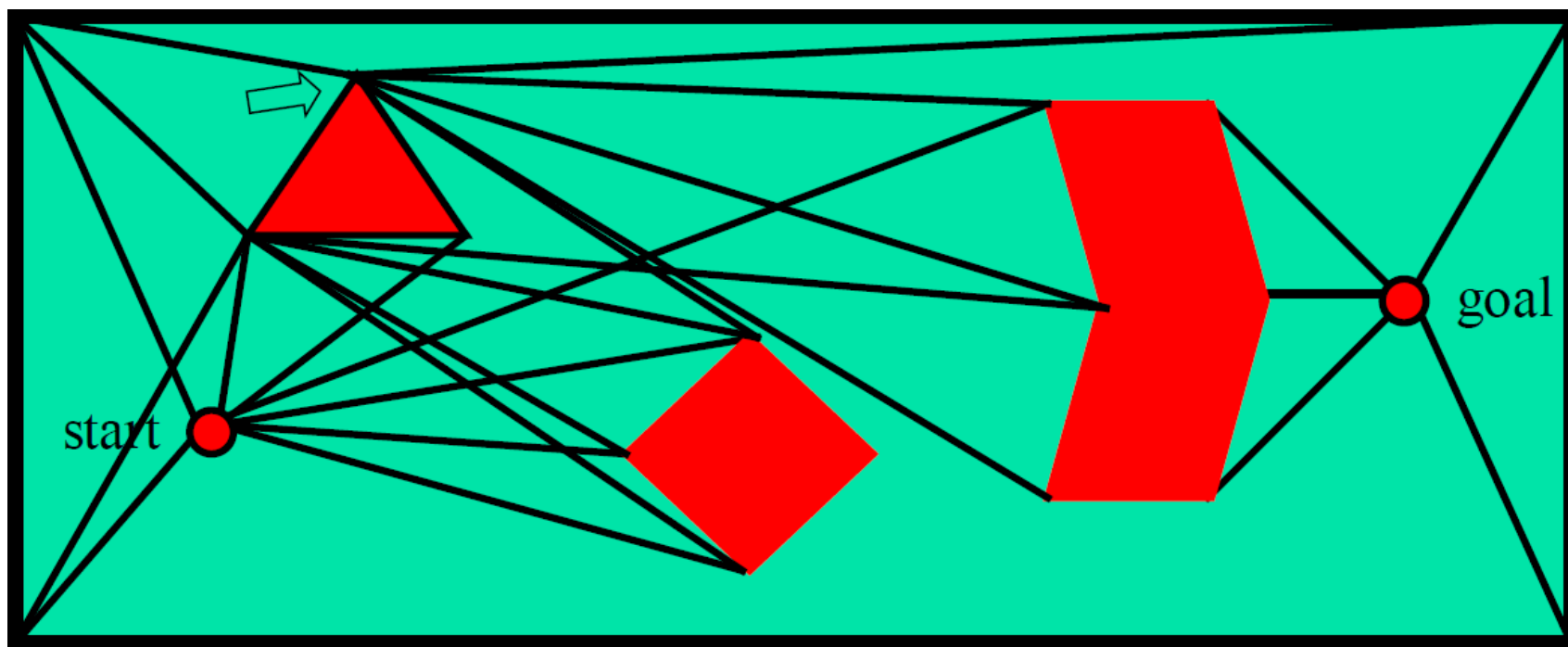
- Příklad konstrukce grafu viditelnosti
 - Přidávej hrany viditelnosti z jednotlivých vrcholů

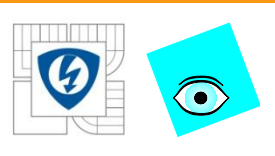




Graf viditelnosti

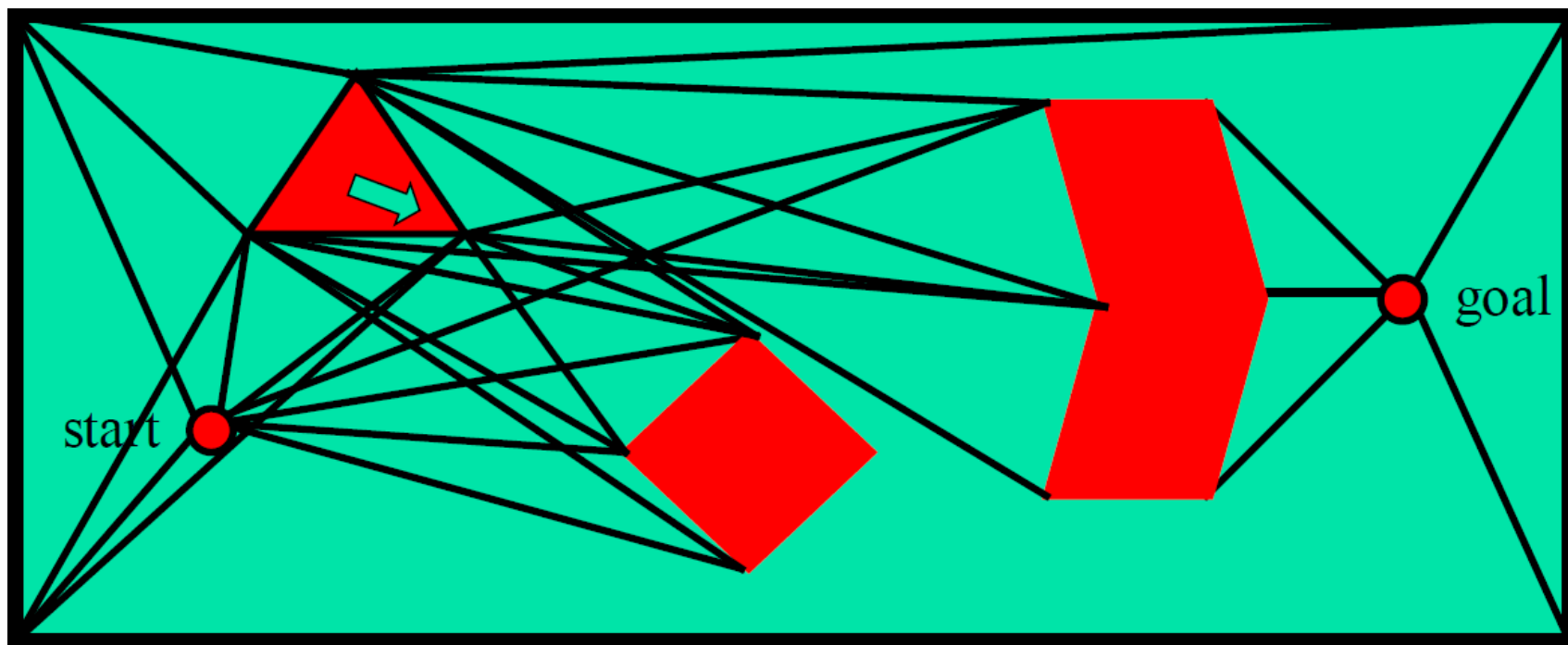
- Příklad konstrukce grafu viditelnosti
 - Přidávej hrany viditelnosti z jednotlivých vrcholů

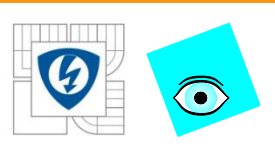




Graf viditelnosti

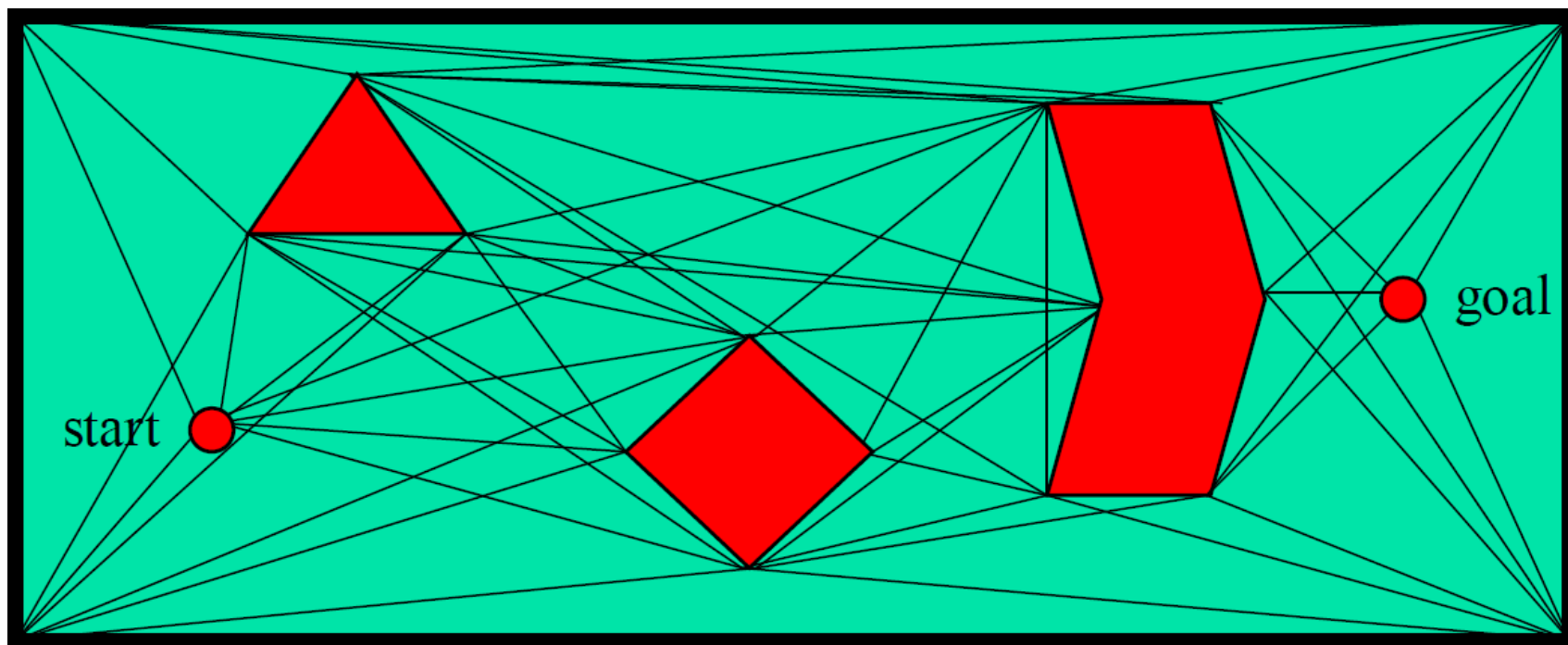
- Příklad konstrukce grafu viditelnosti
 - Nezapomeň přidat i hrany překážek

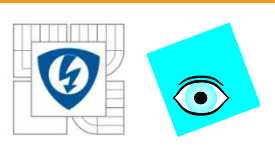




Graf viditelnosti

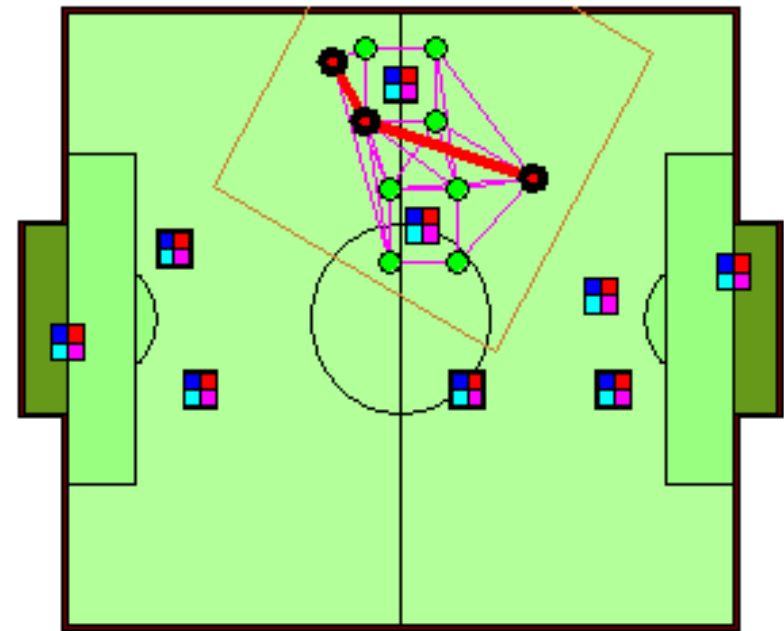
- Příklad konstrukce grafu viditelnosti
 - Opakuj dokud nejsou použity všechny vrcholy

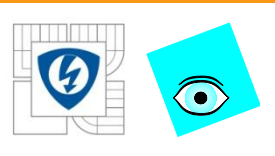




Graf viditelnosti

- Výhody a nevýhody grafu viditelnosti
 - Snaha vést trajektorii co nejbliže k jednotlivým překážkám
 - Nutnost aproximovat překážky polygony
 - Semi-free řešení (dotýká se překážek)
 - + Nalezne řešení vždy pokud existuje
 - + Nalezne nejkratší možnou cestu

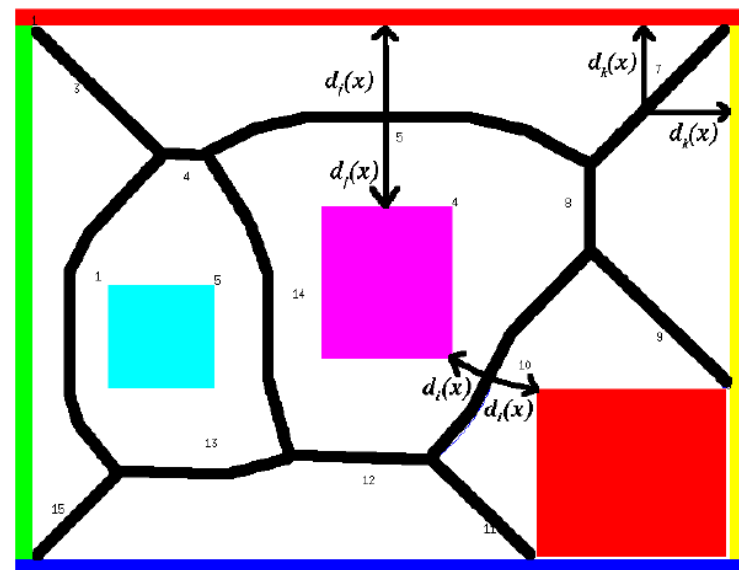


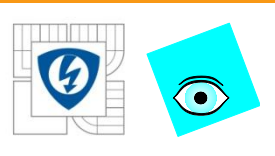


Voroného diagram

- Voronoi diagram (VD)

- Také založen na hledání nejkratší cesty v grafu
- VD: síť, která rozděljuje plochu na regiony vlivu množiny bodů P
- Pro všechny body x takového regionu platí: $V(p_i) = x : |x - p_i| \leq |x - p_j|$ kde p_j jsou všechny body množiny $\{P - p_i\}$
- Uzly grafu: společné body tří sousedních regionů vlivu
- Hrany grafu: hranice sousedních regionů vlivu





Voroného diagram

- Tvorba VD pomocí metody rozděl a panuj

a) Rozděl množinu P na lineárně separabilní podmnožiny

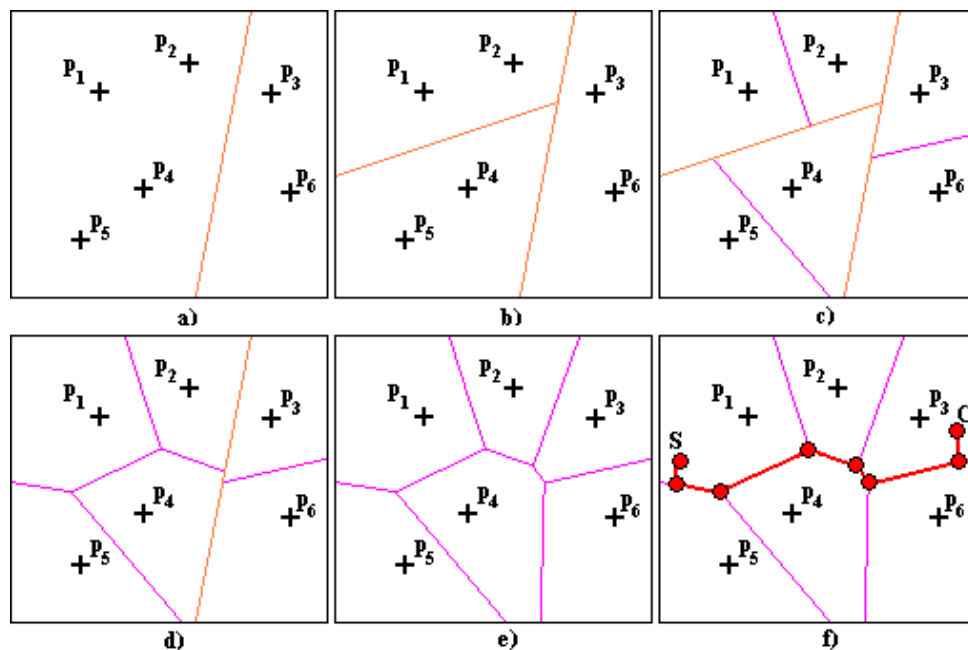
b) Opakuj a) dokud existuje podmnožina s počtem prvků větším než dva

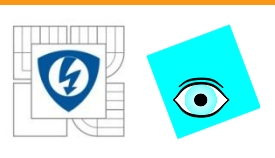
c) Pro tyto podmnožiny sestroj VD, který je redukován na jednu přímku

d) Spoj jednotlivé diagramy pomocí spojnic bodů z obou podmnožin

e) Opakuj d) dokud nebude VD hotov

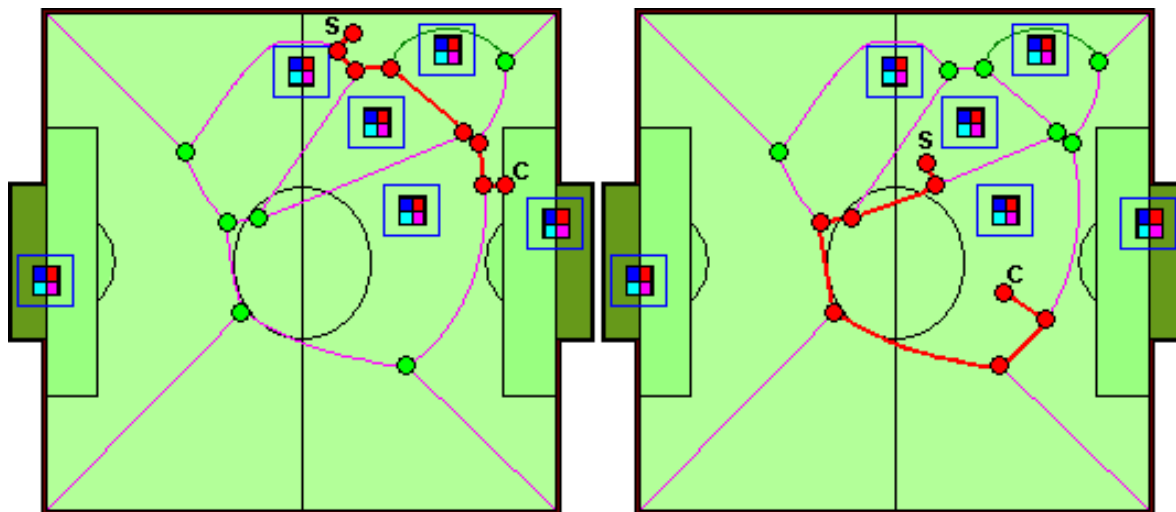
f) Připoj S a C a nalezni nejkratší cestu v grafu

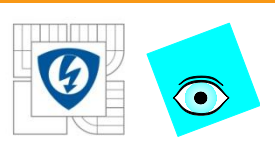




Voroného diagram

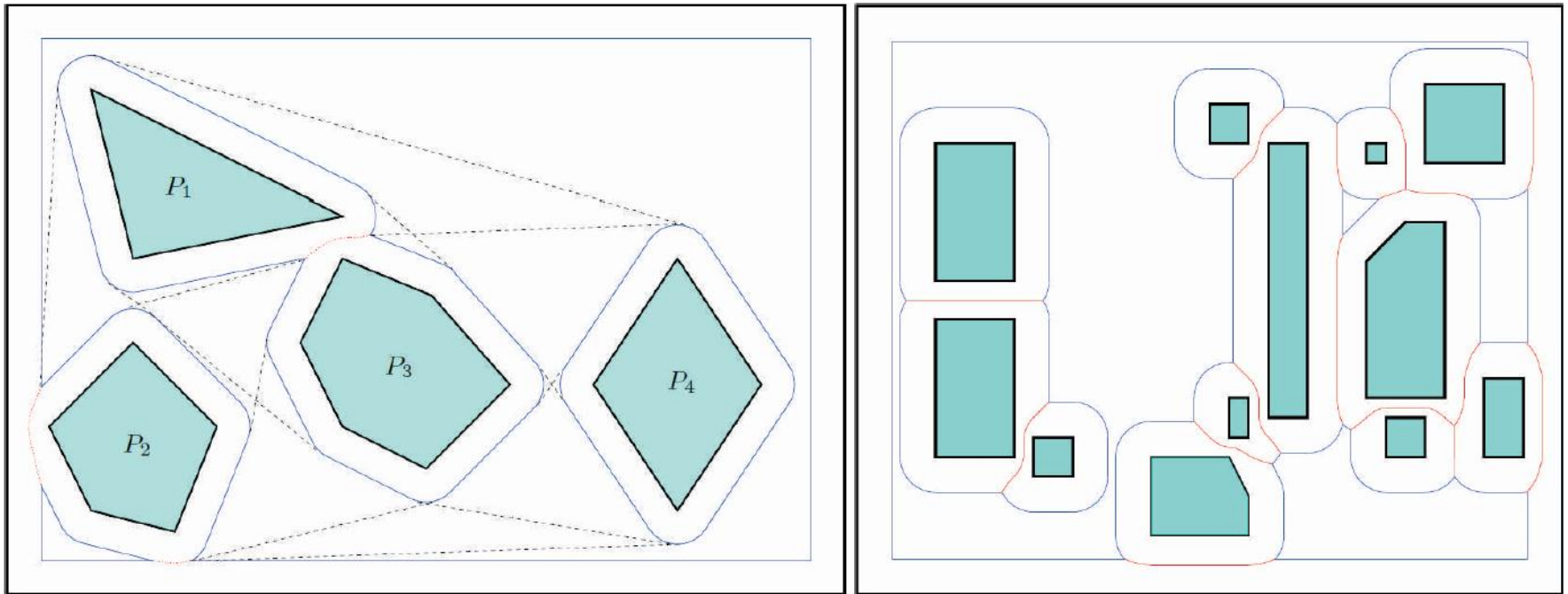
- Výhody a nevýhody Voroného diagramu
 - Složitost konstrukce závisí na typu překážek
 - Delší nalezená řešení
 - + Najde řešení vždy pokud existuje
 - + Bezpečná cesta v dostatečné vzdálenosti od překážek

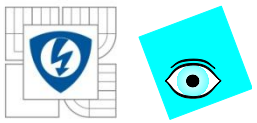




Visibility-Voronoi Complex

- Visibility-Voronoi Complex
 - Kombinuje vlastnosti VD a VG při zachování pozitivních vlastností obou metod

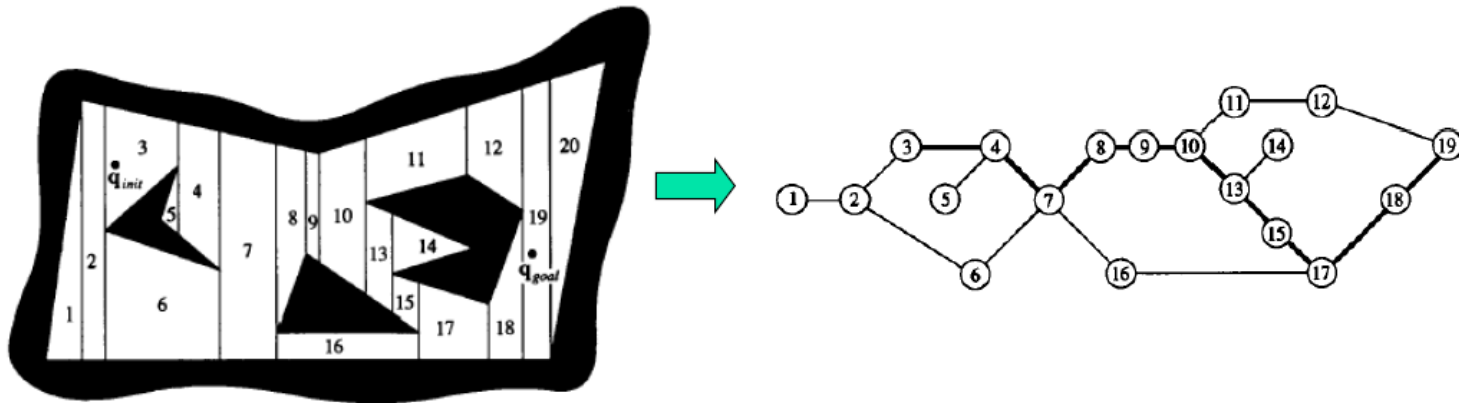


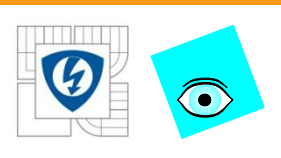


Buněčná dekompozice

- Cell Decomposition

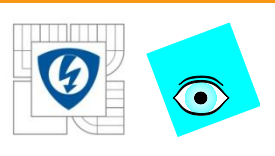
- Založené na hledání nejkratší cesty v grafu
- Snaha rozdělit část volného konfiguračního prostoru robotu na množinu disjunktních oblastí
- Na rozdíl od OG, kde byl disjunktně rozdělený celý pracovní prostor robotu nezávisle na existenci překážek, zde tvar překážek ovlivňuje rozdělení prostoru, ve kterém se robot může pohybovat.





Buněčná dekompozice

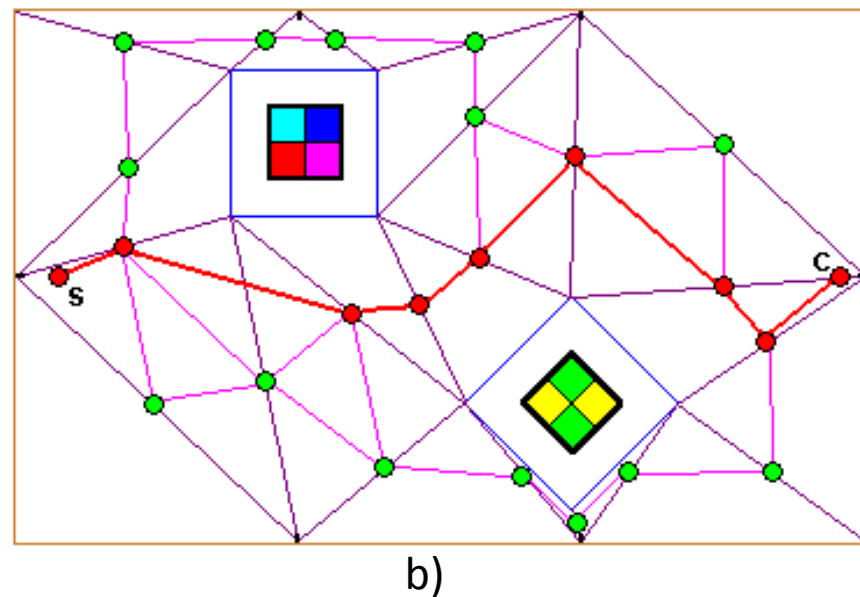
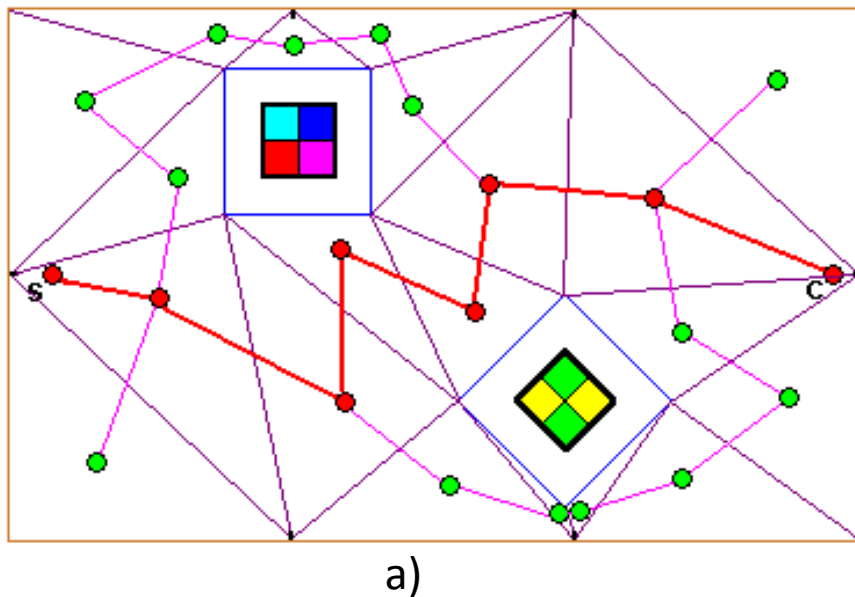
- Trojúhelníková dekompozice
 - Jedna z mnoha variací buněčné dekompozice
 - Volný konfigurační prostor robotu je rozdělen na množinu trojúhelníků
 - Popis základní verze trojúhelníkové dekompozice
 - 1) Trojúhelníková dekompozice vychází z množiny V vrcholů polygonů aproximujících překážky.
 - 2) Trojúhelníky, které jsou tvořeny krajními body hran překážek a bodem z V , který je k této hraně nejbližší, rozdělí volný konfigurační prostor množinou disjunktních polygonů.
 - 3) Polygony stupně většího než tři jsou rozděleny pomocí nejkratší dosud nepoužité spojnice vrcholů tohoto polygonu.
 - 4) Bod 2) je opakován dokud není volný konfigurační prostor rozdělený množinou disjunktních trojúhelníků.

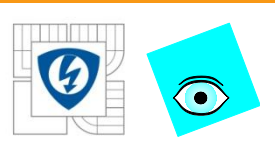


Buněčná dekompozice

- Trojúhelníková dekompozice

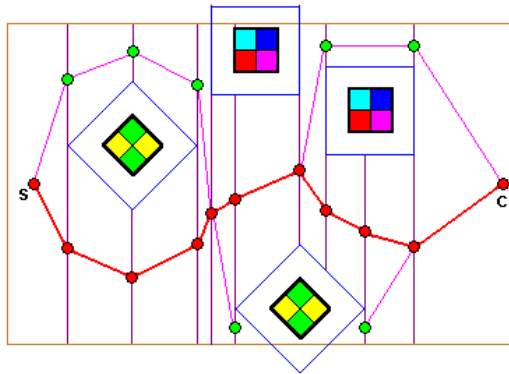
- 5) Uzly grafu: a) těžiště trojúhelníků nebo b) středy stran trojúhelníků
- 6) Hrany grafu: a) spojnice těžišť sousedních trojúhelníků nebo b) spojnice uzlů každého trojúhelníku

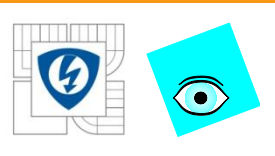




Buněčná dekompozice

- Lichoběžníková dekompozice (LD)
 - Jedna z mnoha variací buněčné dekompozice
 - Volný konfigurační prostor robotu je rozdělen množinou lichoběžníků
 - Popis základní verze trojúhelníkové dekompozice
 - 1) LD vychází z množiny V vrcholů polygonů aproximujících překážky
 - 2) Každým bodem z V je vedena přímka kolmá na spojnici S a C
 - 3) Volný konfigurační prostor mezi přímkami má tvar lichoběžníků

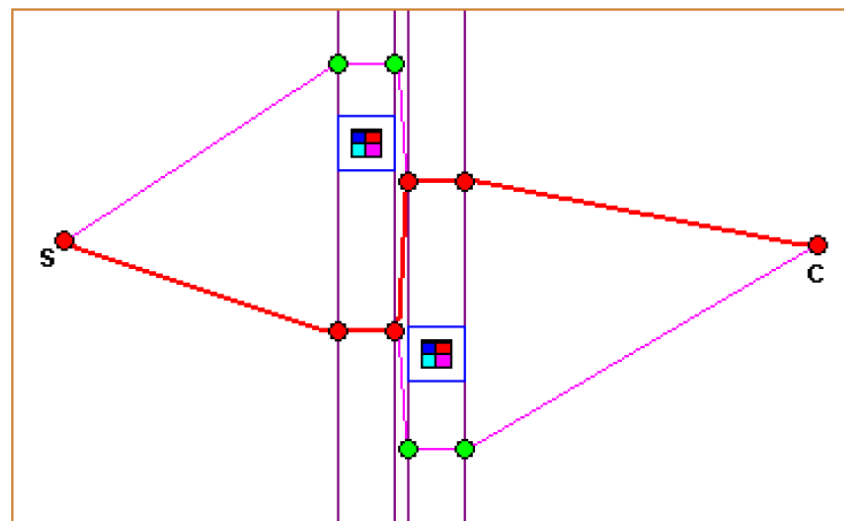
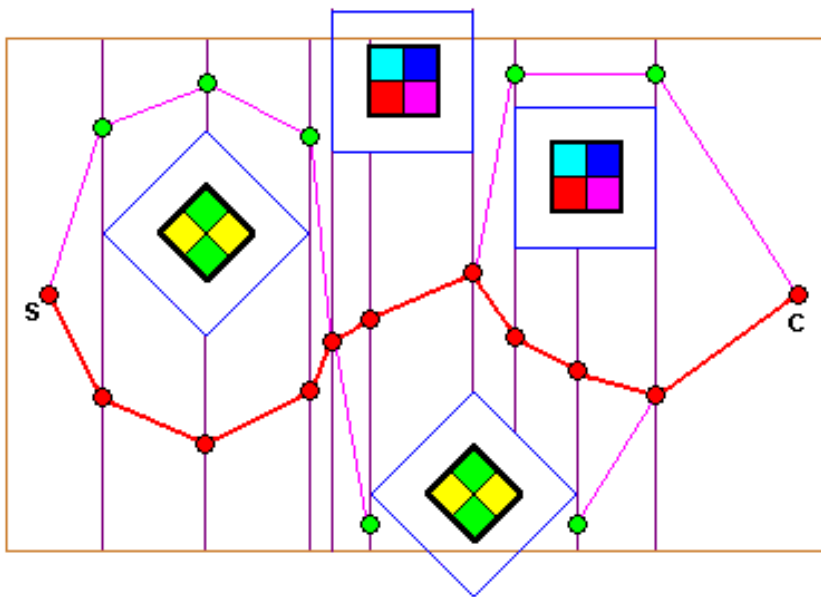


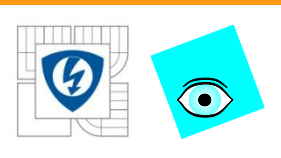


Buněčná dekompozice

- Lichoběžníková dekompozice

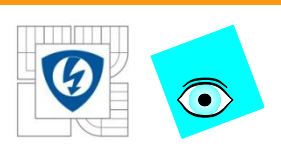
- 4) Uzly grafu: středy stran lichoběžníků vzniklých z vertikálních přímek
- 5) Hrany grafu: spojnice uzlů jednotlivých lichoběžníků





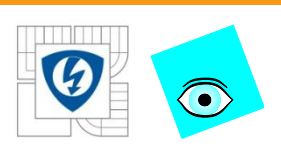
Rapidly-exploring Random Tree - RRT

- Rychle náhodně rostoucí stromy
 - Pravděpodobnostní plánovací algoritmus
 - RRT buduje strom rovnoměrně pokrývající volný konfigurační prostor
 - Uzly stromu jsou konfigurace robotu
 - Hrany stromu jsou přechody mezi konfiguracemi
 - Strom začíná v počáteční konfiguraci
 - Ukončující podmínky růstu stromu
 - Dosažení oblasti kolem požadované konfigurace
 - Vyčerpání daného časového kvanta (real-time aplikace)
 - Expanze maximálního počtu konfigurací



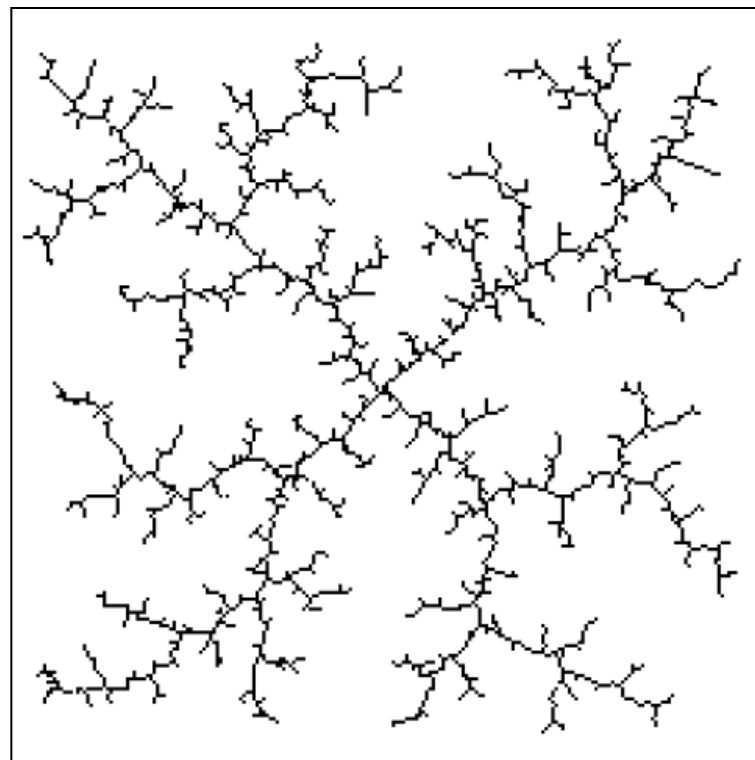
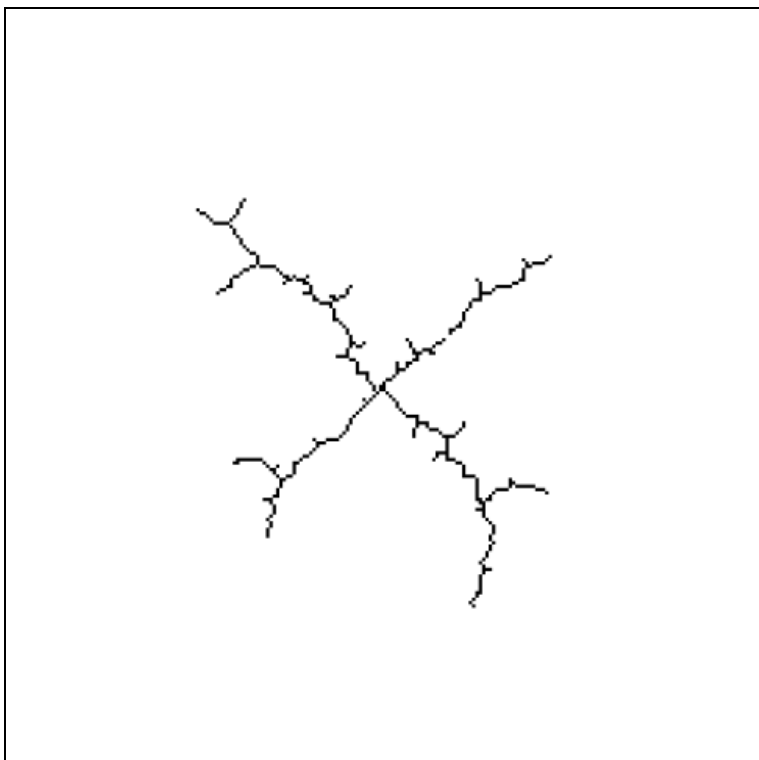
Rapidly-exploring Random Tree - RRT

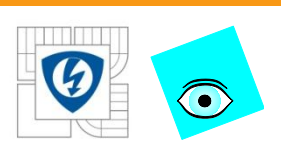
- Nejjednodušší RRT algoritmus:
 1. Inicializace stromu G – strom tvoří počáteční konfigurace
 2. Pro $k = 1..K$ opakuj:
 - a. náhodně vyber bod q v konfiguračním prostoru
 - b. vyber uzel q_b ze stromu G , který je nejbližší konfiguraci q
 - c. vyber akci, která převede konfiguraci q_b na q_n tak, aby q_n bylo co nejbližší q
 - d. pokud vzdálenost q_n a požadovaného stavu q_G je menší než ε ukonči algoritmus a vrať nalezené řešení
 3. Ukonči algoritmus a oznam, že řešení se nepodařilo nalézt



Rapidly-exploring Random Tree - RRT

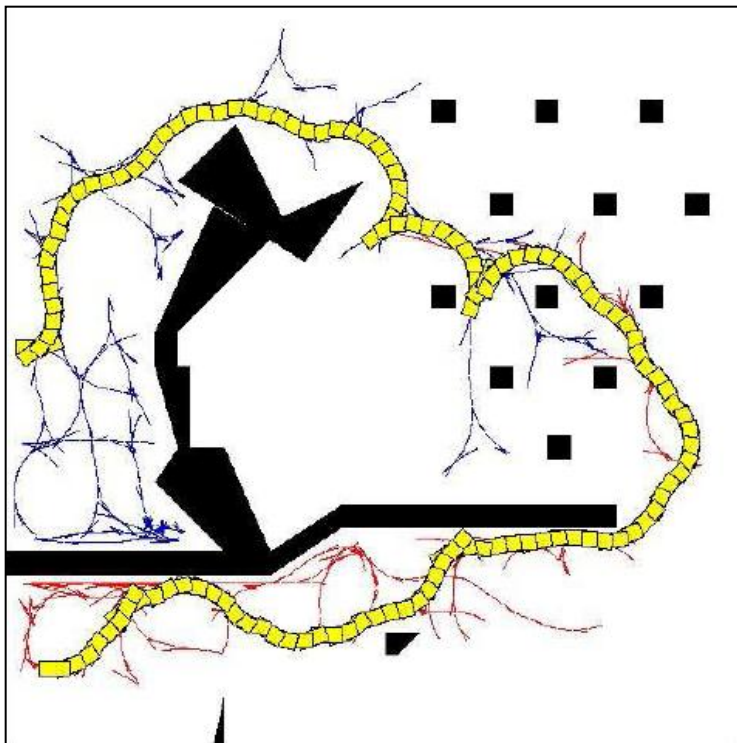
- Růst stromu v konfiguračním prostoru bez překážek



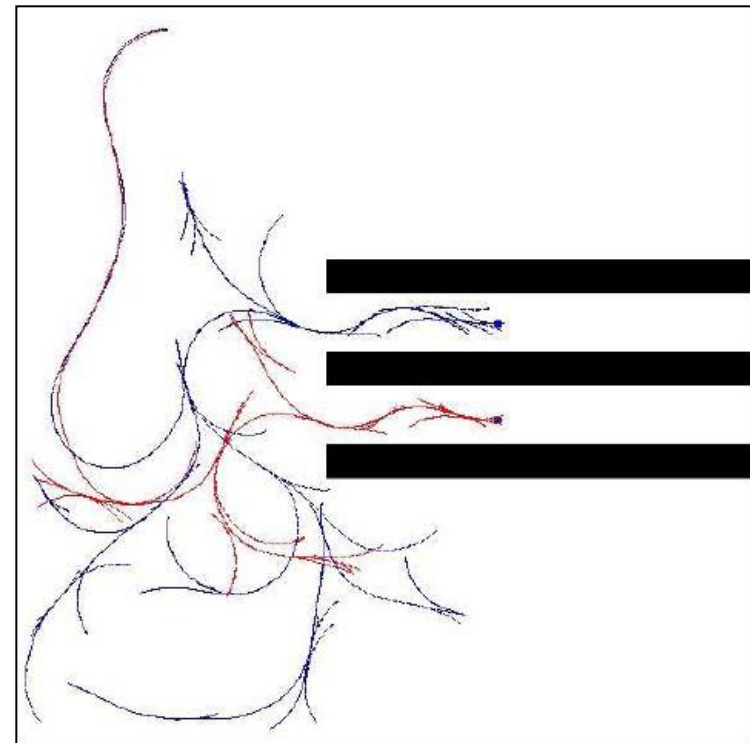


Rapidly-exploring Random Tree - RRT

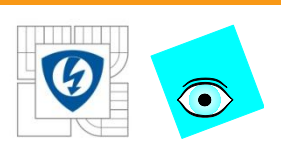
- Růst stromu s kinematickými omezeními



auto

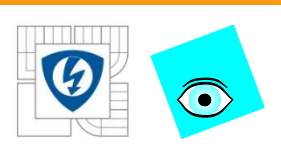


traktor se třemi valníky



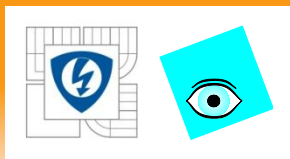
Rapidly-exploring Random Tree - RRT

- Výhody a nevýhody RRT
 - Nedeterministické ukončení výpočtu i vlastního běhu programu
 - Možnost nenalezení cesty i když existuje
 - Řešení je často velmi vzdálené od optima
 - Nutnost následného zpracování (vyhlazení) výsledky
 - + Rychle nalezne platné řešení úlohy plánování pohybu tělesa v prostoru vysoké dimenze
 - + Vysoká úspěšnost v prostoru s vysokým výskytem minim
 - + Hrany stromu jsou přechody mezi konfiguracemi
 - + Konverguje k dosažení rovnoměrného pokrytí konfiguračního prostoru



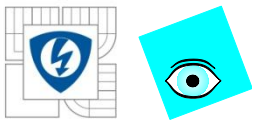
Multi-robotické systémy

- Motivace pro použití multi-robotických systémů
 - Zvýšení spolehlivosti díky možné redundanci
 - Nižší celkový čas řešení daný dekompozicí úlohy
 - v některých aplikacích i více než lineární pokles
 - Možný pokles nákladů použitím jednodušších robotů
 - Nová třída úloh řešitelná pouze MR systémem
- Možné aplikace multi-robotických systémů
 - „cooperative box-pushing“, „cooperative sweeping“, ...
 - robotické konvoje, autonomní dálnice, ...
 - kooperativní UAV: tankování za letu, let v zákrytu, UAV&UGV, ...
 - satelity: „trailing formations“, „cluster formations“, ...



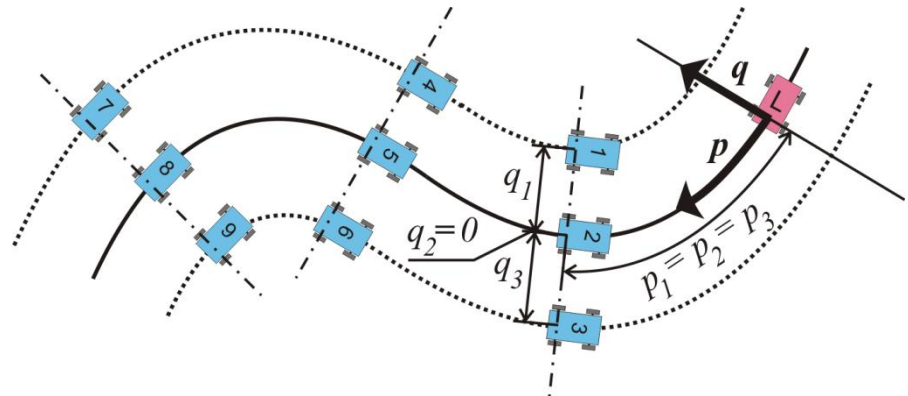
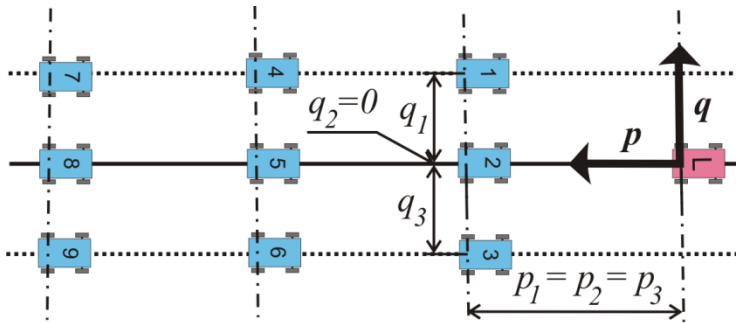
Leader-Follower

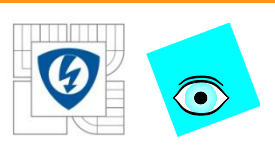
- Formace používající metodu Leader-Follower
 - Jeden případně i několik robotů vybrány jako lídři
 - Ostatní roboty přiřazeny jako jejich následovníci
 - Následovníci udržují definovanou pozici vzhledem k lídrovi
 - Dva základní přístupy udržování formací
 - znalost polohy robotů v mapě je sdílená pomocí komunikace
 - následovníci získají znalost o relativní poloze svého lídra pomocí senzorů
 - detekce a trekování identifikačního obrazce
 - detekce obrysů robotu dálkoměrným senzorem



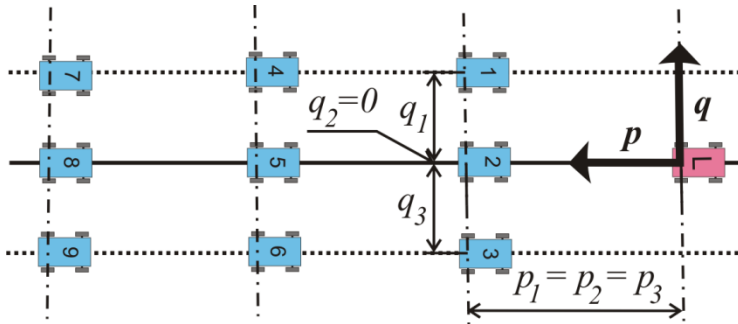
Leader-Follower

- Aplikace metody Leader-Follower
 - Motivovány použitím heterogenní skupiny dobře vybavených robotů a jednodušších (levnějších) robotů
 - Úlohy vyžadující kopírování pohybu vozidel vpředu
 - konvoje vozidel, formace UAV v zákrytu, páry satelitů mapujících dané území sekvenčně v čase.



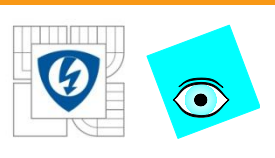


Leader-Follower



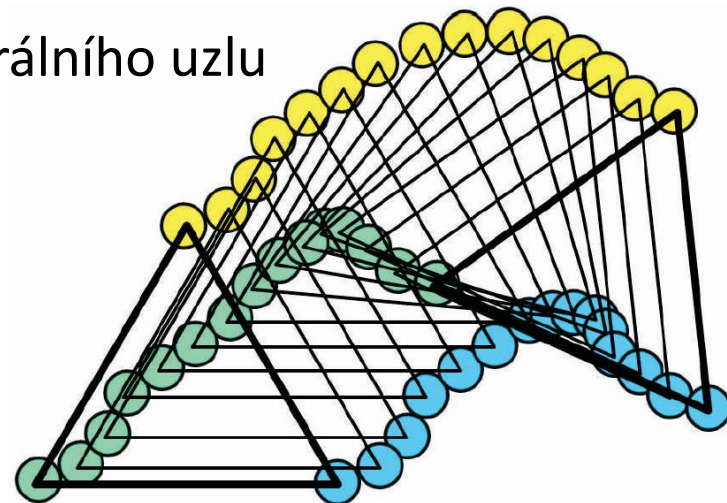
$$\begin{aligned}x_i(t) &= x_L(\tau_i) - q_i(\tau_i) \sin(\theta_L(\tau_i)) \\y_i(t) &= y_L(\tau_i) + q_i(\tau_i) \cos(\theta_L(\tau_i)) \\ \theta_i(t) &= \theta_L(\tau_i)\end{aligned}$$

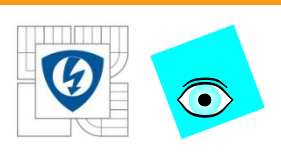
- Výhody a nevýhody metody Leader-Follower
 - Nižší robustnost při použití reálného vůdčího robotu
 - Obtížně použitelné pro koordinaci velkých skupin robotů
 - + Možnost analyzovat stabilitu pohybu formace a kovergenci robotů
 - + Vysoká přesnost udržování formace
 - + Snadná predikovatelnost pohybu jednotlivých robotů ve formaci



Virtuálních struktury

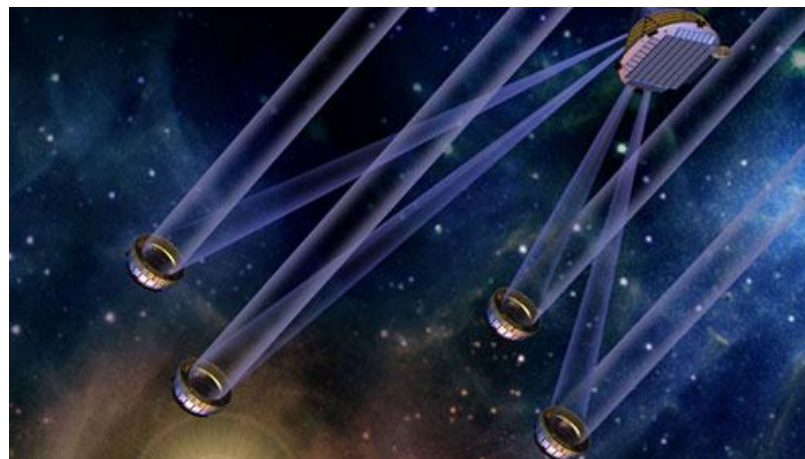
- Formace založené na principu virtuálních struktur
 - Celá skupina robotů uvažována jako jedno pevné virtuální těleso
 - Centrálně počítané řídicí vstupy pro jednotlivé roboty
 - Nutnost komunikace
 - senzorová data od robotů do centrálního uzlu
 - řídicí vstupy zpět k robotům

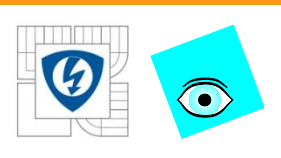




Virtuálních struktury

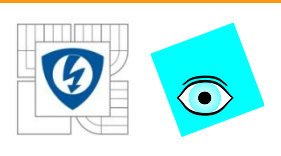
- Aplikace principu virtuálních struktur
 - Dané vysokou přesností udržování tvaru formace
 - Řízení satelitů (virtuální dalekohled/anténa o velkém poloměru)
 - Při transportu předmětů tvar tělesa určuje tvar formace
 - „cooperative box-pushing“





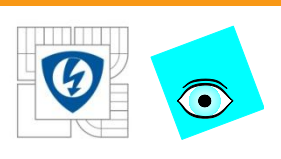
Virtuálních struktury

- Výhody a nevýhody principu virtuálních struktur
 - Obtížně formulovatelný pohyb ve formaci pro některé typy neholonomických robotů
 - Nutnost rekonfigurace řídicích schémat při selhání některého z robotů
 - nižší robustnost daná použitím centralizovaného konceptu řízení
 - Požadavek na přesnou znalost stavu ostatních robotů
 - Požadavek na komunikaci mezi všemi roboty
 - Problém při řízení rozsáhlých struktur
 - + Možnost dosáhnout požadovaného tvaru formace velmi přesně
 - + Pohyb robotů i formace jako celku je snadno predikovatelný

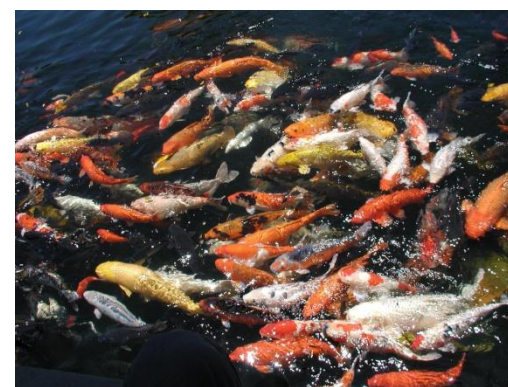


Behaviorální formace

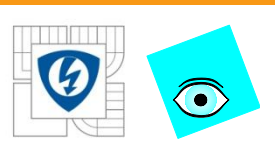
- Formace řízené behaviorálními pravidly
 - Založené na slučování různých vzorů chování do výsledného řízení robotů
 - Součet několika jednoduchých pravidel implikuje chování jednotlivců vedoucí ke splnění společného úkolu
 - Vliv pravidel je vážený podle jejich důležitosti
 - Ochrana před zničením má vyšší prioritu než úspora paliva...
 - Motivace přírodou
 - hejna ptáků, školky ryb, roje hmyzu, molekuly formujících krystal...



Behaviorální formace



- Aplikace behaviorálních formací
 - Dané nízkými nároky na komunikaci a senzorové vybavení
 - Řízení velkých skupin (rojů) jednoduchých robotů
 - Dané možností rychle reagovat na změny v okolí
 - Zpětná vazba od členů skupiny promítnuta do jejího chování okamžitě
 - Aplikace formací pohybujících se ve vysoce dynamickém prostředí



Behaviorální formace

- Výhody a nevýhody behaviorálních formací
 - Obtížná či nemožná schopnost predikce pohybu jednotlivých robotů
 - Obtížné dokázat stabilitu pohybu roje a konvergenci do cílového tvaru
 - Jen pro jednoduché příklady (konstantní natočení robotů apod.)
 - Obtížná formulace pravidel pohybu jednotlivých entit pro dosažení složitějších vzorců chování formace
- + Velmi jednoduchá pravidla stabilizují i poměrně složité struktury
- + Složitost řídicích algoritmů neroste s počtem robotů
- + Komunikační kanál mezi všemi roboty není nutný

