

# PLANNING OPTIMAL PATH IN 2D SPACE WITH OBSTACLES

**Jakub Veřmiřovský**

Bachelor Degree Programme (4), FIT BUT

E-mail: xvermi00@stud.fit.vutbr.cz

Supervised by: Jan Samek

E-mail: samejan@fit.vutbr.cz

**Abstract:** This paper deals with the planning path of the robot in 2D space. Planning a path is often computationally very difficult operation and therefore we try to optimize of these methods and reduce their computational cost.

**Keywords:** Planning path, Nonholonomic robot, 2D space

## 1. ÚVOD

Cílem tohoto článku je seznámení s problémem plánování cesty ve 2D prostoru s překážkami pro neholonomní roboty. Což jsou roboti, kteří podléhají dynamickým a kinematickým omezením jako příklad můžeme uvést auto nebo letadlo. V článku budou prezentovány netriviální metody pro plánování, jejich analýza a v některých případech i vlastní úpravy těchto metod. Tyto metody budou mezi sebou vzájemně porovnány.

## 2. ZÁKLADNÍ POJMY

### 2.1. ROBOT

Neholonomní roboti jsou takoví roboti, kteří podléhají kinematickým a dynamickým omezením. V této práci je robot reprezentován jako letadlo. Jeho omezení je teda maximální úhel náklon křídla. A jelikož letadlo nemůže ve vzduchu zastavit, musíme uvažovat nenulovou rychlost.

### 2.2. DUBINSOVY KŘIVKY

Dubinsovy křivky byly zavedeny z důvodů potřeby hledat nejkratší cestu v prostoru. Nejkratší cesta mezi dvěma konfiguracemi pro *Dubinsovo auto* [1] může být vždy vyjádřena kombinací maximálně tří primitivních směrů (rovně S, doleva L, doprava R). Kombinace těchto primitiv definuje cestu ve 2D prostoru a detailněji je tento princip popsán v práci [1]. Množina možných kombinací tří primitiv:

$$\{LRL, RLR, LSL, LSR, RSL, RSR\}$$

Abychom mohli určit cestu správně, musí být primitiva přesně specifikována a to zavedením jejich délky. Pro směr R a L je zaveden celkový úhel o který se robot otočí a pro směr S je zavedena celková dráha jakou robot urazí.

## 3. PLÁNOVÁNÍ CESTY

Problematika plánování spočívá v tom, že je vytvořen vnitřní model prostředí a nalezena posloupnost akcí taková, aby se po jejich provedení robot nacházel v cíli. Tato posloupnost akcí se nazývá plán. Plánování cesty spočívá v nalezení postupu, jak má robot dosáhnout cíle [2]. Pohyb robota je omezen překážkami a způsobem reprezentace prostředí (spojité, diskrétní).

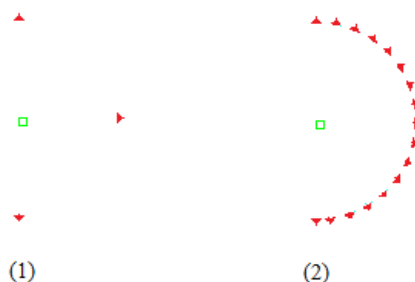
### 3.1. SYSTEMATICKÉ INKREMENTÁLNÍ VZORKOVÁNÍ

Systematické inkrementální vzorkování [2] je diskrétní metoda pro plánování cesty. Tato metoda pracuje s využitím stromového grafu, který je sestrojován inkrementálně pomocí systematické expanze. Podstatou této metody spočívá v tom, že při inicializaci algoritmu je mapa rozložena na buňky velké velikosti a pokud algoritmus s tímto rozložením nenalezne řešení, velikost buněk je zmenšena což způsobí jemnější prohledávání. Zmenšení buněk se provádí o polovinu, ale nesmí klesnout na menší velikost, než jsou rozměry robota.

### 3.2. ACCELERATED A\*

Metoda Accelerated A\* (zkráceně AA\*) [3] patří do kategorie adaptivních algoritmů. Metoda je vhodná, pokud potřebujeme prohledávat velké prostor bez ztráty přesnosti. Odstraňuje kompromis mezi rychlostí a přesností pomocí zavedení adaptivního vzorkování. Princip je takový, že čím jsme dál od překážky, děláme větší kroky, a čím se přibližujeme, kroky zmenšujeme.

V mé bakalářské práci byla implementována základní verze této metody a vlastní úprava. Tato úprava spočívá v rozvoji jednotlivých uzlů. Uzel expandujeme všemi možnými směry, oproti originálu, kdy se expanduje pouze s maximálním zatočením do obou směrů a rovným směrem. Díky této úpravě jsme byli schopni snížit délku cesty, neboť originální verze není optimální (tj. že metoda nenalezne nejkratší cestu).



Obrázek 1: Expanze metody AA\* (1), úprava expanze AA\* (2).

### 3.3. ITERATIVE ACCELERATED A\*

Metoda Iterative Accelerated A\* [4] je rozšíření algoritmu AA\* obsahující několik vylepšení, díky kterým je možné většině případů výslednou cestu získat rychleji, než při použití AA\*. Zjednodušeně řečeno tento algoritmus používá AA\*, kterou spouští pouze s překážkami, s kterými vznikl konflikt v předchozím kroku.

### 3.4. RYCHLE ROSTOUCÍ NÁHODNÉ STROMY

Myšlenkou rychle rostoucích náhodných stromů (RRT) [1] je co nejrychleji a nejrovnoměrněji prohledávat konfigurační prostor. Tato metoda RRT je založena na stromové struktuře, jež se rozrůstá za pomoci náhodně vygenerovaných uzlů (konfigurací) a hrany stromu jsou jednotlivé akce robota.

Pro zrychlení konvergence k cíli metody RRT byla zavedena úprava GoalBias [1]. Modifikace spočívá upravením funkce pro generování náhodné konfigurace zavedením malé pravděpodobnostní hodnoty, že nově vygenerovaná konfigurace bude cílová.

Ve své práci jsem rovněž implementoval rozšíření metody RRT. Tato úprava spočívá v tom, že při generování náhodného uzlu je pravděpodobnost 0.06, že náhodný uzel bude cílový (stejně jako úprava GoalBias), ale dále je prostor rozdělen na 4. kvadranty a kvadrantům, ve kterých se nachází počáteční nebo cílový uzel je zvednuta pravděpodobnost nově vygenerovaného uzlu o 0.11. Pravděpodobnostní parametry byly stanoveny empiricky na základě provedených experimentů. Tyto experimenty rovněž ukázaly, že ve složitých prostorech poskytuje vlastní úprava metody lepší konvergenci k cíli než GoalBias.

## 4. EXPERIMENTY

V rámci výše popsaných algoritmů a jejich úprav byly provedeny experimenty pro dva různé stavové prostory. První stavový prostor (označen jako *Mapa 1*) obsahuje jedinou překážku, kterou specifikuje zeď mezi počáteční a cílovou konfigurací. Druhý stavový prostor (*Mapa 2*) je složitější struktura překážek, která představuje bludiště. V následující tabulce jsou shrnuty dosažené experimentální výsledky. Jednotlivé výsledky v tabulce byly získány jako průměr z 10 spuštění hledání.

Název metody	Mapa 1, počet vygenerovaných uzlů	Mapa 1 počet iterací	Mapa 2, počet vygenerovaných uzlů	Mapa 2 počet iterací
Systematické inkrementální vzorkování	18	15	23	20
AA*	46	37	Nenalezeno	Nenalezeno
AA* s úpravou	304	122	24	2
Iterative AA*	23	2	154	4
RRT	14985	999	28850	1923
RRT GoalBias	9090	606	11034	736
RRT s úpravou	4412	294	5760	384

**Tabulka 1:** Porovnání jednotlivých metod pro různé stavové prostory.

## 5. ZÁVĚR

V mojí bakalářské práci jsem se zaměřil na problematiku plánování cesty ve 2D prostoru s překážkami. V této práci byly popsány tři používané metody a to včetně mých vlastních rozšíření. Všechny tyto popsané metody byly implementovány v jazyce Java včetně jejich vizualizací. Rovněž byly implementovány i vlastní rozšíření, které jsou v této práci rovněž popsány.

S využitím implementované aplikace jsem provedl experimentální ověření jednotlivých metod včetně vlastních rozšíření a výsledky těchto experimentů jsem popsal v kapitole 4. Z výsledků vyplývá, že diskretní metoda dosahuje největší rychlosti, ale není tak přesná jako ostatní metody. Algoritmy AA\* mohou diskretní metodě v rychlosti konkurovat, ale jsou mnohem přesnější. AA\* bohužel není úplná, ale po úpravě dosahuje nejlepších výsledků a je úplná. Metody RRT jsou určeny pro prohledání velkých prostorů a proto jejich výsledky jsou v porovnání s ostatními takto špatné. Vlastní úpravy metod, jak vyplývá z experimentů, podávají mnohem lepší výsledky v případě složitých map.

## PODĚKOVÁNÍ

Tato práce byla částečně podpořena operačním programem Výzkum a vývoj pro inovace v rámci projektu Centrum Excellence IT4Innovations (CZ.1.05/1.1.00/02.0070).

## REFERENCE

- [1] LaValle, Steven M. *Planning algorithms*. Cambridge university press, 2006. ISBN 0-521-86205-1.
- [2] P. Křeček. *Plánování cesty autonomního lokomočního robotu na základě strojového učení*. Brno, 2010. Disertační práce. Vysoké učení technické v Brně.
- [3] D. Šišlák, P. Volf a M. Pěchouček. Flight Trajectory Path Planning. *Proc. of Intl. Scheduling and Planning Applications Workshop*. 2009.
- [4] Kopriva, S.; Sislak, D.; Pavlicek, D.; Pechoucek, Michal. Iterative accelerated A\* path planning, *49th IEEE Conference on Decision and Control (CDC)*, 2010, vol., no., pp.1201,1206, 15-17 Dec. 2010