

# 33IRO - Inteligentní robotika

## Plánování optimální trajektorie manipulátoru v neznámém prostředí

Pavel Jisl, Jindřich Očenášek, Radek Šmerák

Katedra Kybernetiky, Fakulta Elektrotechnická  
České Vysoké Učení Technické  
{jislp,ocenash,xsmerak}@fel.cvut.cz

12. ledna 2004

### Abstrakt

Práce se zabývá inženýrským úkolem, který spočívá v návrhu optimální cesty pro přenos tělesa robotem. Jsou využity znalosti z kinematiky, počítačového vidění a umělé inteligence. Pracovní plocha je pozorována perspektivní kamerou a z obrazu je získána poloha počátečního a koncového bodu trajektorie. Protože je nutné znát polohu chapadla, je řešena přímá a inverzní kinematická úloha. Optimální trajektorie přesunu tělesa v neznámém prostoru je navržena pomocí plánovacího algoritmu  $A^*$ . Správnost návrhu byla ověřena simulací v programu Matlab.

## 1 Úvod

Úkolem úlohy je návrh algoritmu, který přenesení těleso v neznámém prostředí po optimální trajektorii. Tento problém je často řešen např. v průmyslu při přemísťování předmětů robotem, umístěným nad pracovní plochou. Je totiž nutné zajistit, aby byl předmět přemístěn po nejkratší dráze bez ohrožení okolních předmětů a osob.

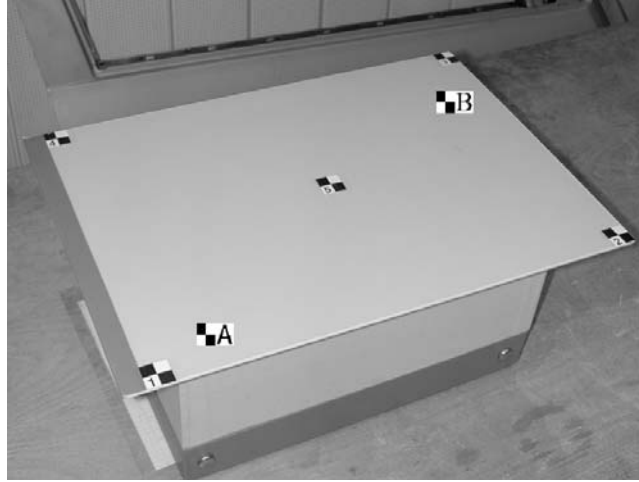
Vyřešení úlohy jsme rozdělili do tří podproblémů, které lze řešit odděleně. První krok spočívá ve zjištění souřadnic počátečního a koncového bodu. Tyto body jsou zadány pomocí obrázku, získaného z neznámé perspektivní kamery umístěné v neznámém bodě. Je tedy nutné nalézt transformaci, kterou získáme souřadnice počátečního a koncového bodu ze zadaného obrázku. Tímto se zabýváme v části 2.

Druhým krokem je popsání kinematiky robota. To spočívá v nalezení vztahu mezi pozicí chapadla v kartézských souřadnicích pracovní plochy a kloubovými souřadnicemi robota. Je tedy nutné vyřešit přímou a inverzní kinematickou úlohu. Řešení je v části 3.

Posledním krokem je vlastní plánování trasy, po které bude předmět přenášen. K tomu je využíván plánovací algoritmus  $A^*$ , kterým se zabýváme v části 4.

## 2 Zpracování obrazu z kamery

V této části úlohy se zabýváme zpracováním obrazu, získaného z perspektivní kamery, umístěné v neznámé poloze. V zadaném obrázku (obr. č. 1) jsou zobrazeny body **A** a **B** (tedy bod počáteční a cílový) a několik pomocných bodů se známými souřadnicemi. Tyto body využijeme pro výpočet transformační matice **H**.



Obrázek 1: Pohled kamery na pracovní plochu se zadanými body

Matici **H** vypočteme podle vztahu

$$\alpha_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad (1)$$

kde  $u_i, v_i$  jsou souřadnice bodu v obraze a  $x_i, y_i$  jsou souřadnice v kalibrační rovině.  $\alpha_i$  je zvětšení, které je pro každou dvojici bodů  $[u_i, v_i]$  a  $[x_i, y_i]$  jiné. Matice **H** je regulární o velikosti 3x3. Úpravami vztahu (1) pak pro každou dvojici bodů v obraze a na pracovní ploše získáváme následující dvojici rovnic.

$$\begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & u_i v_i & u_i y_i & u_i \\ 0 & 0 & 0 & -x_i & -y_i & -1 & v_i x_i & v_i y_i & v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ \vdots \\ h_{33} \end{bmatrix} = 0. \quad (2)$$

Pro určení matice  $\mathbf{H}$  jsou potřeba minimálně 4 rovnice (2), tedy čtyři různé body v obraze. Podmínkou však je, že žádné tři body v rovině nesmí ležet na přímce. Vlastní matici  $\mathbf{H}$  jsme vypočítali pomocí rozkladu SVD<sup>1</sup>. Tento singulární rozklad rozloží matici  $\mathbf{H}$  tak, že  $H = U.S.V^T$ , kde  $U$  a  $V$  jsou ortogonální matice a  $S$  je diagonální matice. Řešením soustavy pak je sloupec matice  $V$ , který odpovídá nejmenšímu singulárnímu číslu matice  $A$ . V našem případě to je poslední sloupec matice  $V$ . Přesný popis řešení soustav rovnic pomocí SVD rozkladu je v [1].

Výpočtem jsme získali následující matici homografie  $\mathbf{H}$

$$\mathbf{H} = \begin{bmatrix} -0.0038 & 0.0008 & -0.3717 \\ 0.0006 & 0.0018 & -0.9283 \\ -0.0000 & -0.0000 & -0.0020 \end{bmatrix}. \quad (3)$$

### 3 Kinematika robota

V části kinematika robota se zabýváme řešením přímé a inverzní kinematické úlohy. Protože se pohyb robota uskutečňuje nastavováním kloubových souřadnic a pozice chapadla je udávána v kartézských souřadnicích pracovní plochy, je nutné získat transformační rovnice mezi těmito vyjádřeními.

Pro převod z kartézských souřadnic, které určují pozici chapadla, do kloubových souřadnic robota je nutné vyřešit přímou kinematickou úlohu, opačný převod (tedy z kloubových souřadnic do kartézských) řeší kinematická úloha inverzní.

#### 3.1 Přímá kinematická úloha

Přímá kinematická úloha je úloha, kdy známe pozici chapadla v kartézské souřadné soustavě pracovního prostoru a je nutné vypočítat transformaci z těchto souřadnic do kloubových souřadnic robota.

Tuto úlohu jsme vyřešili pomocí Denavitovy-Hartenbergovy metody [4]. Tato metoda je výhodná, protože využívá homogenních souřadnic a pomocí jedné matice se dá vyjádřit jak rotace, tak translace. Vzhledem k tomu, že manipulátor má 3 klouby, dostáváme následující matice.

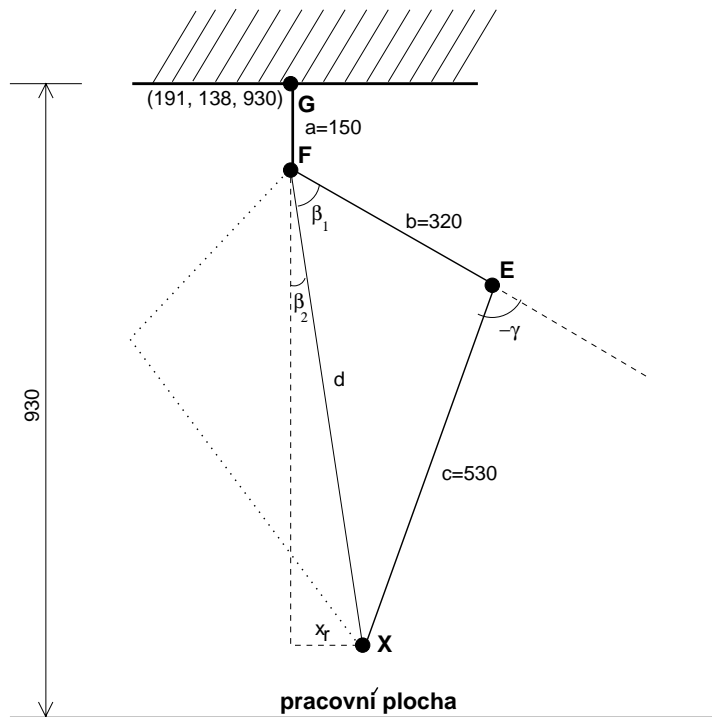
$$\mathbf{A}_{\mathbf{RG}} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) \quad \mathbf{A}_{\mathbf{FE}} = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma & 0 \\ 0 & 1 & 0 & 0 \\ \sin\gamma & 0 & \cos\gamma & -b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$\mathbf{A}_{\mathbf{GF}} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & -a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) \quad \mathbf{A}_{\mathbf{ch}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Výsledkem je pak následující výraz

---

<sup>1</sup>Singular Value Decomposition



Obrázek 2: Nákres ramen robota s vyznačením důležitých hodnot

$$\begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} = \mathbf{A}_{RG} \cdot \mathbf{A}_{GF} \cdot \mathbf{A}_{FE} \cdot \mathbf{A}_{ch} \cdot \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}, \quad (8)$$

kde  $\begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T$  jsou souřadnice bodu **G** (z obrázku 2).

### 3.2 Inverzní kinematická úloha

Inverzní kinematická úloha slouží k přepočtu ze známých kloubových souřadnic do kartézské soustavy souřadnic chapadla.

Výpočet jsme odvodili pomocí goniometrických funkcí. Výpočtem získaných vzorců získáme úhly pro dvě řešení (viz. obrázek 2) a další dvě získáme otočením v kloubu G (o úhel  $\alpha$ ).

$$x_r = \sqrt{x^2 + y^2} \quad (9)$$

$$d = \sqrt{x_r^2 + (z - a)} \quad (10)$$

$$\beta_1 = \arccos\left(\frac{d^2 - b^2 - c^2}{2bc}\right), \beta_2 = \arctan\left(\frac{x_r}{d}\right) \quad (11)$$

$$\beta = \beta_1 + \beta_2 \quad (12)$$

$$\gamma = 180^\circ - \arccos \frac{b^2 + c^2 - d^2}{2bc} \quad (13)$$

$$\alpha = \arctan \frac{y - y_p}{x - x_p} \quad (14)$$

Inverzní kinematická úloha má obvykle čtyři různá řešení. Na okraji pracovního prostoru má řešení jen dvě, na přípustné části osy kloubu G má nekonečně mnoho řešení. Mimo pracovní prostor úloha řešení nemá.

## 4 Plánování

Tato část úlohy spočívá v naplňování nejkratší cesty z počátečního bodu do bodu koncového. Souřadnice těchto bodů jsme získali v části 2. Naším úkolem je najít optimální (nejkratší) trajektorii, po které bude přesun proveden. Je nutné počítat s tím, že v pracovním prostoru jsou rozmístěné překážky s neznámou polohou, kterým se musí chapadlo robota vyhnout (nesmí do nich narazit).

Souřadnice, podle kterých se trajektorie plánuje, jsou zadávány v kartézské soustavě okolního světa. Pohyb robota je však uskutečňován natáčením jednotlivých ramen, proto jsou souřadnice bodů vyjadřovány v kloubových souřadnicích.

Stavový prostor je určen polohou koncového bodu chapadla. Jedná se tedy o množinu všech bodů, do kterých lze dopravit předmět. Tyto body získáme natočením kloubů G, F, H robota. Při změně jednoho či více kloubů získáme z aktuálního stavu 26 stavů následujících (nejmenší krok kloubu E je  $\Delta\alpha = 2^\circ$ , kloubu F  $\Delta\beta = 2^\circ$  a kloubu G  $\Delta\gamma = 3,4^\circ$ ).

Vzhledem k neznámé poloze překážek je výhodné použít metodu, která při prohledávání stavového prostoru odhaduje cenu dosažení cíle z daného stavu. Proto využijeme algoritmu  $A^*$  [2, 3].

### 4.1 Algoritmus $A^*$

Pro plánování implementujeme  $A^*$  algoritmus, který používá heuristickou funkci

$$\hat{f}(m) = \hat{g}(m) + \hat{h}(m), \quad (15)$$

kde

$\hat{f}(m)$  je odhad ceny optimální cesty z výchozího bodu (uzlu) B do cílového bodu A vedoucí přes bod  $m$ ,

$\hat{g}(m)$  je odhad ceny nejkratší cesty z výchozího bodu B do  $m$  (aktuální uzel), v našem případě reprezentuje dosud prošlou cestu z bodu B do bodu  $m$  (součet cen prošlých hran),  $\hat{h}(m)$  je odhad ceny nejkratší cesty z  $m$  do koncového bodu A, v naše případě to je odhad nejkratší cesty z bodu  $n$  do bodu A, který reprezentován euklidovskou vzdálenost mezi body  $m$  a A.

Tento algoritmus je přípustný [3], pokud je cena každé hrany větší než  $\delta > 0$  a  $\hat{h}(m) \leq h(m)$ .

Plánování nejkratší cesty po spuštění z počátečního bodu B pracuje podle definice algoritmu  $A^*$  [3] až do doby, kdy je proximitním čidlem detekována překážka. Tento bod je vyjmut ze stavového prostoru a provádí se nové plánování z předchozího bodu podle algoritmu  $A^*$ .

## 5 Experimenty

### 5.1 Přesnost měření

Bod	Souřadnice		Přepočtené souřadnice kamera $\rightarrow$ manipulátor
	v ss. manipulátoru	v ss.kamery	
Souřadnice použité pro výpočet			
1	[0, 0]	[187, 467]	[0, 0]
2	[383, 0]	[764, 295]	[383, 0]
3	[383, 277]	[549, 72]	[383, 277]
4	[0, 277]	[61, 174]	[0, 277]
Kontrolní bod			
5	[192, 138]	[403, 233]	[192, 138]

Tabulka 1: Tabulka znázorňující přesnost výpočtu souřadnic

Pro ověření správnosti výpočtu matice H jsme využili referenčních bodů, u kterých jsou známy jak souřadnice v souřadné soustavě manipulátoru tak i v souřadné soustavě kamery. Transformaci jsme provedli jak pro 4 body, které byly použity pro výpočet matice H, tak i pro kontrolní bod.

Z tabulky 1 je vidět, že matice H byla vypočtena správně, neboť souřadnice přepočtené ze ss. kamery do ss. manipulátoru souhlasí s reálnými souřadnicemi.

### 5.2 Korektnost přímé a inverzní kinematické úlohy

Správnost výpočtu přímé a inverzní kinematické úlohy je možné ověřit výpočtem přímé kinematické úlohy při znalosti natočení kloubů  $[\alpha, \beta, \gamma]$ , čímž získáme polohu chapadla  $[x, y, z]$  v kartézských souřadnicích. Po aplikaci inverzní kinematické úlohy na tyto souřadnice pak získáme hodnoty natočení kloubů  $[\alpha', \beta', \gamma']$ . Pokud je výpočet korektní, hodnoty  $[\alpha, \beta, \gamma]$  a  $[\alpha', \beta', \gamma']$  by měly být shodné (v určité toleranci). Tento výpočet ukazuje tabulka 2.

Kloubové souřadnice	Bod v prostoru	Kloubové souřadnice
$[\alpha, \beta, \gamma]$	$[x, y, z]$	$[\alpha', \beta', \gamma']$
$[20^\circ, 60^\circ, -30^\circ]$	$[700, 323, 161]$	$[20^\circ, 60, 1^\circ, -30, 2^\circ]$

Tabulka 2: Tabulka znázorňující korektnost IKU a PKU

### 5.3 Experimenty plánování

Simulaci hledání nejkratší cesty jsme implementovali v programu Matlab. Simulační obrázky 3 ukazují vlastní experimenty hledání nejkratší cesty. Nejkratší cesta byla nalezena za 123 kroků.

Obrázek 4 pak ukazuje námi navržený algoritmus  $A^*$ . Jeho overení bylo provedeno na pracovní ploše bez umístěných překážek. V tomto případě algoritmus našel cestu za 23 kroků.

## 6 Závěr

Navrhli jsme řešení inženýrského problému, který se týkal všech důležitých oblastí kybernetiky, robotiky a umělé inteligence.

V první části jsme zpracovali obraz z neznámé perspektivní kamery a získali transformační matici pro přepočet souřadnic počátečního a koncového bodu do souřadnic manipulátoru. Správnost výsledné matice homografie (3) byla experimentálně ověřena v části 5.1.

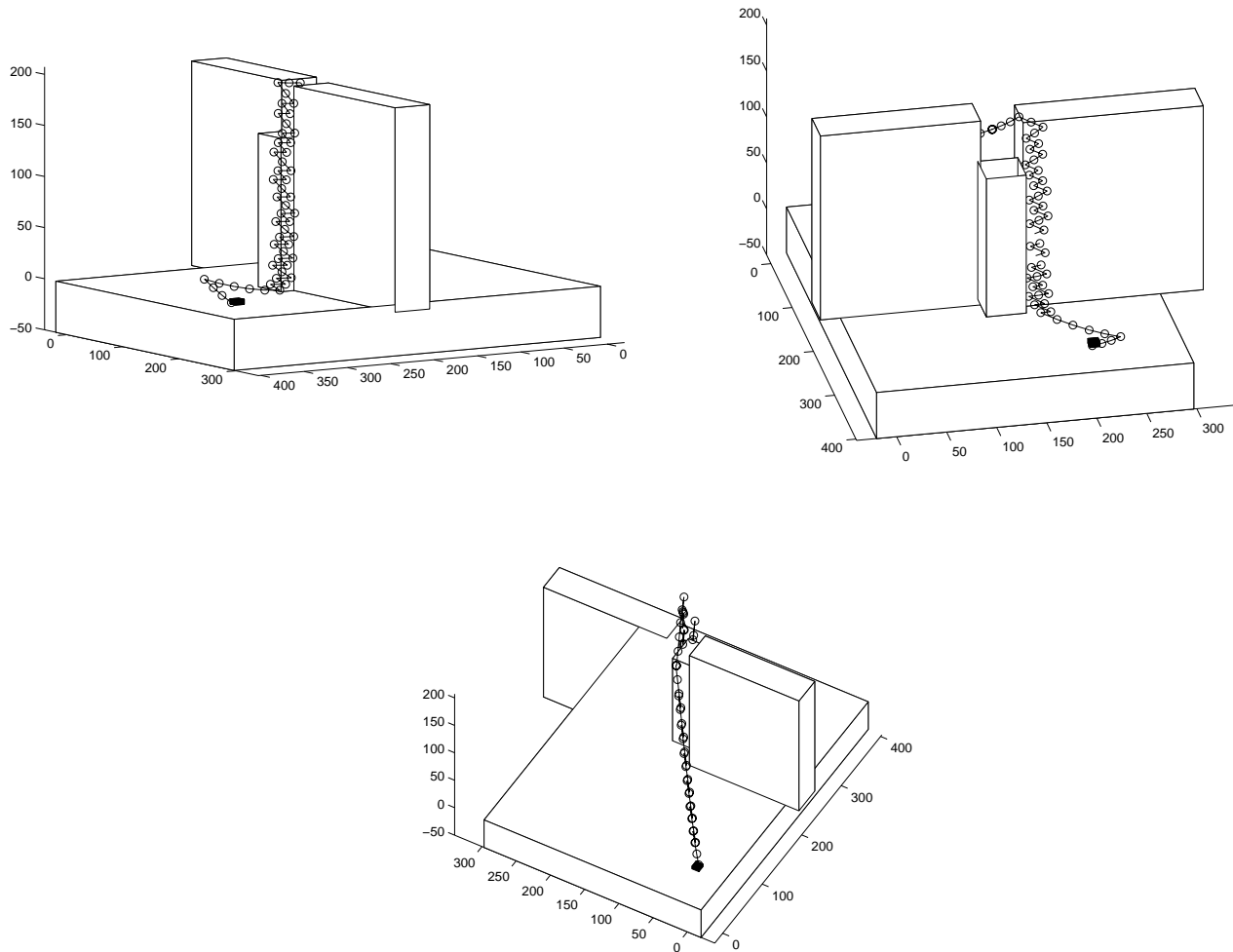
V druhé části jsme povedli výpočet přímé a inverzní kinematické úlohy. Korektnost řešení jsme ověřili v části 5.2.

V poslední části jsme navrhli plánovací algoritmus, který je založen na algoritmu  $A^*$ . Námi navržený algoritmus našel řešení za 123 kroků v případě hledání nejkratší cesty v prostoru s neznámými překážkami. V prostoru bez překážek našel řešení za 23 kroků.

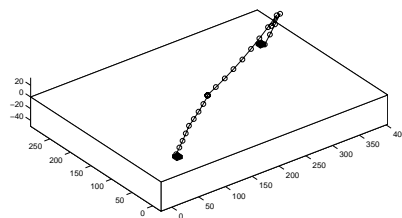
Námi navržený algoritmus bylo však nutné modifikovat. Původní verze, která při detekci kolize s překážkou kolizní bod vyjmula ze stavového prostoru a provedla přeplánování trajektorie pomocí algoritmu  $A^*$ , byla velice časově náročná, neboť robot neustále hledal nejkratší cestu a „ohmatával“ překážku. Proto jsme plánovací algoritmus modifikovali a to tak, že při první kolizi jsme preferovali přechod přes překážku směrem nahoru, a proto jsme zakázali při dalším plánování pohyb v kloubu G. Po přechodu přes překážku jsme opět použili vlastní algoritmus  $A^*$ .

## Reference

- [1] Krajník, E. *Maticový počet*. Praha: Vydavatelství ČVUT, 2000.
- [2] Mařík V., Štěpánková O., Lažanský J. *Umělá inteligence (1)*. Praha: Academica, 1993.
- [3] Pajdla, T. *Inteligentní robotika*, **online**, 1.1.2002  
<http://cyber.felk.cvut.cz/iro>.
- [4] Smutný, V. *Průsvitky předmětu Robotika*, **online**, 1.1.2002  
<http://cyber.felk.cvut.cz/robotika>.



Obrázek 3: Experimenty hledání nejkratší cesty



Obrázek 4: Ověření funkčnosti algoritmu  $A^*$