

# Robotika: Inverzní kynematická úloha

Ondřej Čermák, Pavel Jisl

13. listopadu 2001

Vyřešení inverzní kynematické úlohy pro robot  
Bosch.

## Zadání úlohy

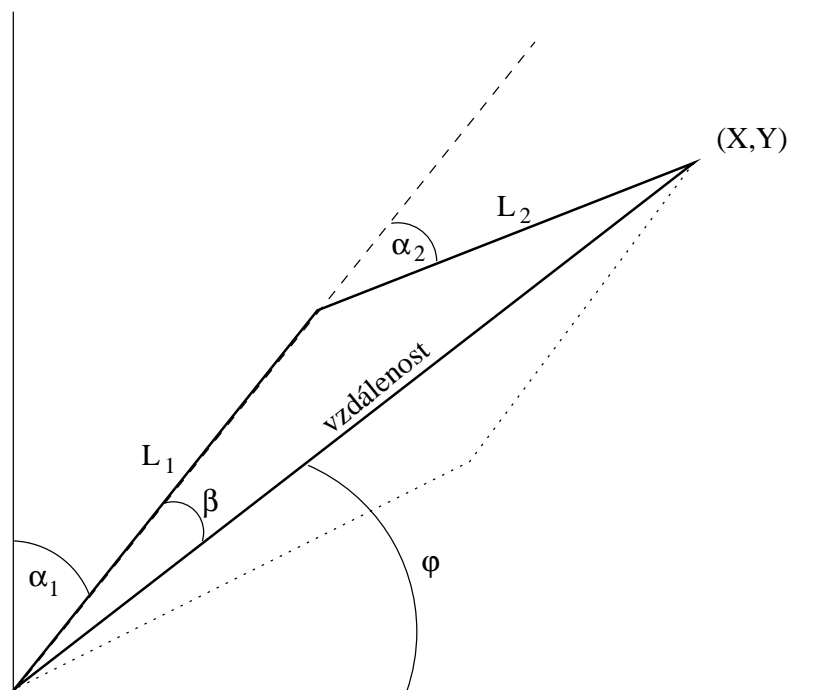
Úkolem je analyzovat kinematiku robotu, navrhnout postup výpočtu IKU a implementovat ho v Matlabu. Robot je řízen v kloubových souřadnicích. Vstupem robotu bude zadaný bod ve světových kartézských souřadnicích nebo úsečka daná dvěma body. Robot se na pokyn přesune do zadaného bodu, případně provede pohyb po úsečce. Chybové stavy budou ošetřeny. Písaná dokumentace k robotu bude k dispozici v místnosti 132, neodnášejte ji.

### 0.1 Úkoly

1. Seznamte se s vybraným robotem, jeho kinematikou a řízením.
2. Seznamte se s knihovnou RT toolbox pro Matlab, která implementuje komunikaci se seriovou linkou.
3. Vyřešte inverzní kinematickou úlohu pro robot Bluebot/Bosch a implementujte ji v matlabu.
4. Otestujte řešení na bodech v různých oblastech pracovního prostoru robotu.
5. Vyhotovejte zprávu o úloze. Zhodnoťte výsledky.

# 1 Vyřešení inverzní kynematické úlohy

Nákres souřadnic, pro výpočet potřebných úhlů a vzdáleností:



Z tohoto schématu pak dostáváme vzorec pro výpočet vzdálenosti

$$vzd^2 = L_1^2 + L_2^2 + 2L_1L_2\cos\alpha_2,$$

z čehož po jednoduché úpravě dostáváme

$$\alpha_2 = \pm \arccos\left(\frac{vzd^2 - L_1 - L_2}{2L_1L_2}\right).$$

Pro výpočet úhlu  $\beta$  vycházíme z rovnice

$$L_2^2 = vzd^2 + L_1^2 - 2.vzd.L_1\cos\beta,$$

kde po úpravě dostáváme výraz pro výpočet úhlu  $\beta$

$$\beta = \arccos\left(\frac{vzd^2 + L_1^2 - L_2^2}{2.vzd.L_1}\right).$$

Úhly  $\varphi$  a  $\alpha_1$  pak vypočteme pomocí vztahů

$$\varphi = \arctan\left(\frac{Y}{X}\right)$$

a

$$\alpha_1 = 90^\circ - \varphi \mp \beta.$$

Tento postup platí pro pravou polorovinu souřadné soustavy. Pro levou polorovinu je výpočet shodný (je s pravou polorovinou symetrická podle osy) a proto stačí předchozí výsledky hodnotou  $-1$ .

Pro výpočet těchto hodnot je použita funkce, napsaná v programovacím jazyku programu Matlab. Jeho výpis je v příloze č. 1.

## 2 Otestování pracovních bodů

### 3 Příloha č. 1 - implementace IKU v Matlabu

```
function [uhly]=mojeikt(robot,X,Y);
r1=250;
r2=200;
vzd = sqrt (X^2+Y^2);
uhly(1,2)=acos((X^2+Y^2-r1^2-r2^2)/(2*r1*r2))*180/pi;
if imag(uhly(1,2)) ~=0
    uhly=[];
    return
end
beta=acos((r2^2-vzd^2-r1^2)/(-2*vzd*r1))*180/pi;
if X==0
    gama=90;
else
    gama=atan(Y/X)*(180/pi)*sign(X);
end
uhly(1,1)=90-gama-beta;
uhly(1,3)=0;
uhly(1,4)=0;
uhly(2,1)=90-gama+beta;
uhly(2,2)=-uhly(1,2);
uhly(2,3)=0;
uhly(2,4)=0;
if X<0
    uhly(:,1:2)=uhly(:,1:2)*(-1);
    pom=uhly(1,:);
    uhly(1,:)=uhly(2,:);
    uhly(2,:)=pom;
end
```