

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

PLZEŇ, 2013

Bc. JAN HRDINA

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 17.5.2013

.....
vlastnoruční podpis

PODĚKOVÁNÍ

Touto cestou bych rád poděkoval svému vedoucímu diplomové práce panu Prof. Ing. Miloši Schlegelovi CSc. za cenné rady a připomínky poskytnuté během tvorby tohoto textu. Dík patří i Ing. Miroslavu Jiříkovi a Ing. Martinu Švejdovi. V neposlední řadě bych chtěl poděkovat své rodině a mojí přítelkyni za podporu během celého studia.

Anotace

Cílem této diplomové práce je navrhnout alternativní architekturu mobilního robotu pro ultrazvukovou diagnostiku svarů potrubí. Robot by se měl pohybovat po potrubí z feromagnetického materiálu a sledovat svar, který bude zvýrazněn barevnou čarou kontrastní k barvě potrubí. Informace z okolního prostředí by měly být získávány pouze pomocí kamery umístěné na těle robotu, čili není uvažovaný žádný navigační systém, který by zařízení dodával informace o jeho aktuální poloze a natočení vzhledem k nějakému neměnnému (světovému) souřadnému systému.

Klíčová slova

mobilní kolový robot, oskulační kružnice, PD regulátor, homogenní transformační matice, vizuální zpětná vazba, počítačové vidění, barevný model RGB, barevný model HSV, prahování

Annotation

This thesis aims to propose an alternative architecture of a mobile robot for an ultrasonic diagnosis of pipeline welds. The robot should move on the surface of the pipeline made of a ferromagnetic material and follow the weld, which will be highlighted by a color contrasting to the color of the pipeline. Information from the environment should be obtained only by a camera mounted on the robot body; thus, it is not considered any navigation system that should provide any information about robot's current position and rotation with respect to the world coordinate system.

Keywords

mobile wheeled robot, osculating circle, PD controller, homogenous transformation matrix, vision-based feedback, computer vision, RGB color model, HSV color model, thresholding

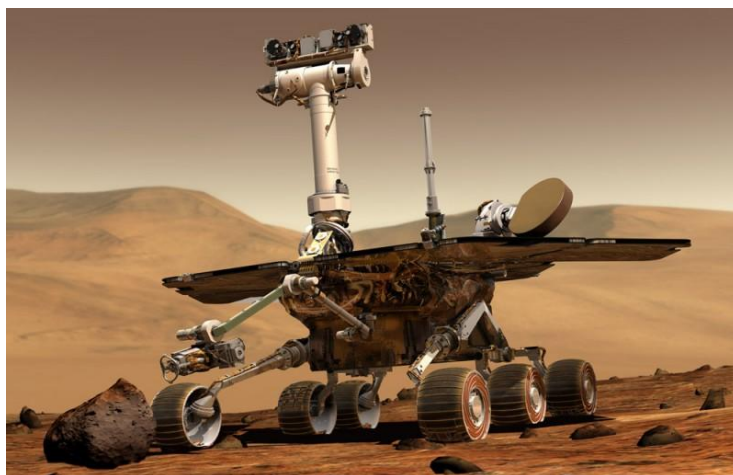
Obsah

Kapitola 1	1
1 Úvod.....	1
1.1 Cíle práce	3
2 Mobilní robot s nezávisle řízenými koly.....	6
2.1 Kinematika.....	8
2.2 Návrh řízení předpokládající dokonalou znalost operačního prostoru	13
2.3 Dynamika.....	16
2.4 Model kolového robotu s nezávisle řízenými koly	21
3 Použité metody potřebné pro návrh řízení	23
3.1 Pohyb po kružnici	23
3.2 Oskulační kružnice.....	26
3.3 PD regulátor.....	31
3.4 Geometrické transformace v prostoru.....	33
3.5 Vizualní zpětná vazba	36
Kapitola 2	39
4 Počítačové vidění	39
4.1 Barevné modely RGB a HSV.....	39
4.2 Ideální obrázek - prahování, aproximace polynomem a konstrukce oskulační kružnice ..	44
Kapitola 3	51
5 Experiment.....	51
5.1 Tvorba testovacího prostředí	52
5.2 Pohled z robotu	53
5.3 Práhování reálného obrázku	56
5.4 Proložení reálného obrázku polynomem a konstrukce oskulační kružnice	58
5.5 Návrh řízení	61
5.6 Rozšíření na perspektivní pohled z robotu.....	68
5.7 Tvorba videozáznamu.....	73
5.8 Testování na různých křivkách	74
Závěr.....	80
Použitá literatura	82

Kapitola 1

1 Úvod

Návrh mobilních kolových robotů je nedílnou součástí vědního oboru *robotika*. Praktický význam těchto strojů roste, protože rostou i požadavky lidstva na čím dál větší podíl automaticky fungujících zařízení. Někdy je navíc příliš nebezpečné nebo dokonce nemožné, aby se člověk na dané místo dostal. Proto vznikají například dálkově řízené kolové roboty na průzkum vyhořelých budov s porušenou statikou. Reprezentativním příkladem místa, kam se člověk (zatím) nedokáže sám dostat, a proto tam vyšle kolový stroj, je planeta Mars (Obr. (1)).



Obr. (1): Sonda Opportunity na planetě Mars. [14]

Robot může být ovládán úplně (*teleoperovaný* robot), nebo mohou být některé činnosti částečně autonomní (*semiautonomní* robot). Plně *autonomní* robot je schopen činnosti zcela bez lidského zásahu, případně je činnost lidského operátora omezena na zadávání cílů robotu, který potom sám volí nejlepší způsob dosažení daného cíle.

Kolové roboty mohou mít nejrůznější architektury. Některé z nich jsou decentní, jiné až bizarní. Několik příkladů neotřelých návrhů mobilních kolových robotů je znázorněno na Obr. (2).



Obr. (2): Příklady extraordinárních architektur kolových robotů. [14]

Mezi klasické typy kolových robotů (Obr. (3)) patří například roboty podobné automobilu (*car-like robots*) se čtyřmi koly nebo šestikolová vozítka (viz výše zmíněná sonda Opportunity).

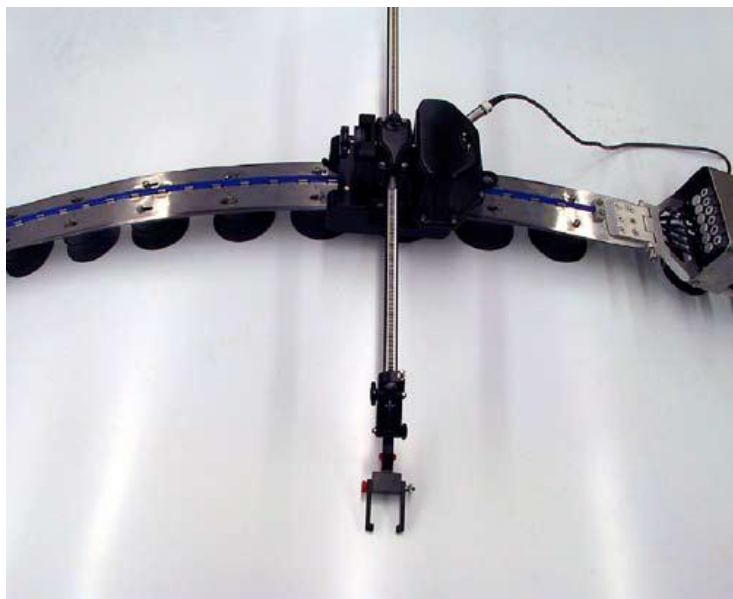


Obr. (3): Příklady klasických architektur mobilních kolových robotů. [14]

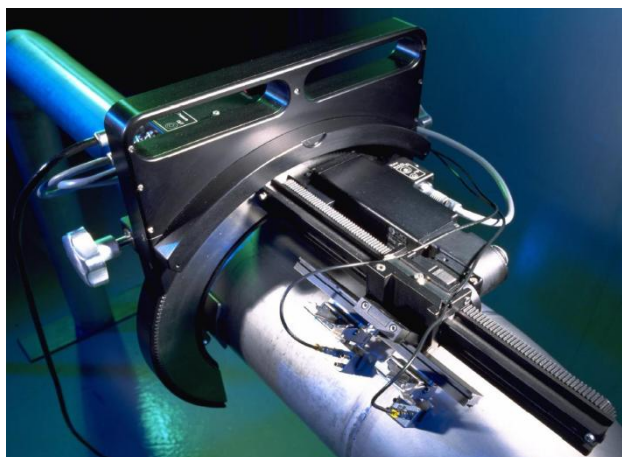
1.1 Cíle práce

Tato diplomová práce má za cíl navrhnout alternativní architekturu mobilního robotu pro ultrazvukovou diagnostiku svarů potrubí. Robot by se měl pohybovat po potrubí z feromagnetického materiálu a sledovat svar, který bude zvýrazněn nějakou barevnou čarou kontrastní k barvě potrubí.

Používané architektury mobilních robotů pro ultrazvukovou diagnostiku mají většinou základ buď v kombinaci nějakého pásu, který se manuálně připevní na potrubí a po kterém se robot posléze pohybuje, s posuvným ramenem, na němž se nachází ultrazvuková diagnostika (Obr. (4)), nebo je robot přímo zkonstruován tak, že se připevní na trubku a tu potom obkrouží efektor s ultrazvukovou diagnostikou (Obr. (5)), anebo může být celá procedura pohybu po svaru provedena ručně (Obr. (6)).



Obr. (4): Robot s označením Wing scanner od firmy Olympus. [13]



Obr. (5): Robot s označením APS-3 od firmy Force Technology. [13]



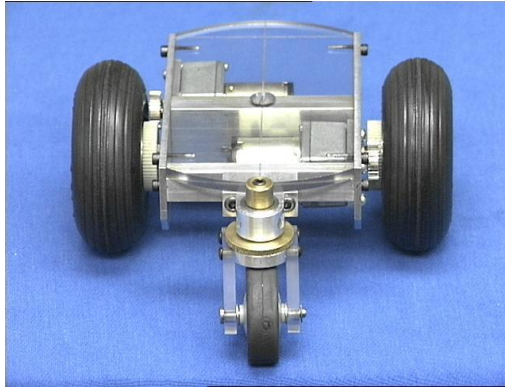
Obr. (6): Robot s označením MWS-5 od firmy Force Technology. [13]

Lze nalézt samozřejmě i portálové roboty nebo speciální architektury pro potrubí s velkým průměrem (Obr. (7)).



Obr. (7): Portálový robot s označením AWS-8 (vlevo) a robot určený pro potrubí s velkým průměrem s označením AMS-41 (vpravo). Oba stroje od firmy Force Technology. [13]

Alternativní konstrukce, která bude popsána v této diplomové práci, vychází z architektury mobilního robotu s dvěma nezávisle řízenými magnetickými koly a jedním kolem hnaným pro účely stability celé platformy (Obr. (8)). Návrh konstrukce magnetických kol ani popis samotného procesu ultrazvukové diagnostiky nejsou součástí této práce.



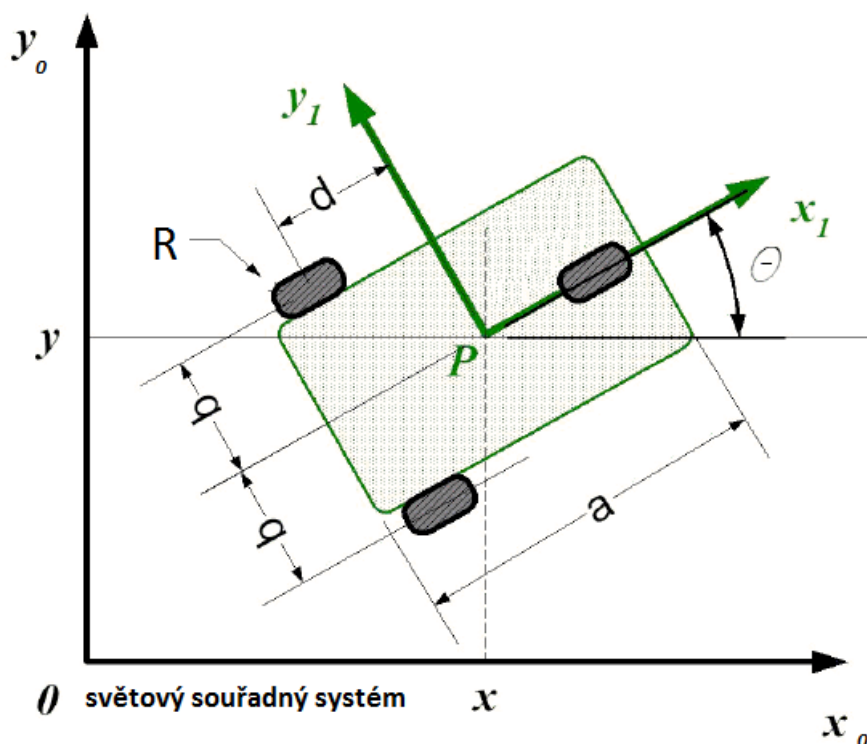
Obr. (8): Navrhovaná architektura robotu. [14]

Informace z okolního prostředí by měly být získávány pouze pomocí kamery umístěné na těle robotu, čili není uvažovaný žádný navigační systém, který by zařízení dodával informace o jeho aktuální poloze a natočení vzhledem k nějakému neměnnému (světovému) souřadnému systému.

Práce je rozdělena do tří kapitol, přičemž první dvě pojednávají o obecných metodách a teoretických základech použitých pro návrh řízení, respektive o základech počítačového vidění a některých principech pro rozpoznávání objektů z obrázku. Třetí kapitola popisuje, jak byly tyto teoretické základy upraveny, eventuálně rozšířeny, pro reálný problém a jaké jsou simulační výsledky získaného algoritmu řízení.

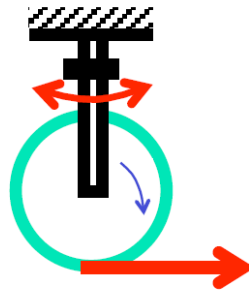
2 Mobilní robot s nezávisle řízenými koly

V této kapitole bude pojednáváno především o kinematickém modelu kolového robotu se dvěma nezávisle řízenými koly (Obr. (9)). Dále zde bude navrženo jednoduché řízení. Nezávisle poháněná kola jsou umístěna na společné příčné ose otáčení. Ze známých požadavků na pozdější využití stroje je možno předpokládat, že se robot bude pohybovat po prakticky vodorovné ploše (poloměr potrubí bude vzhledem k velikosti robotu a vzdálenosti kol velký). Dále se předpokládá, že povrch potrubí bude hladký bez nějakých výrazných nerovností a že kola robotu budou v neustálém kontaktu s podložkou bez eventuálního prokluzu. Uvažovaný model kolového robotu rovněž zanedbává třecí síly mezi koly a podložkou stejně jako třecí a odporové síly způsobené činností motorů urychlujících hnací kola, odpor vzduchu apod.



Obr. (9): Schéma modelu kolového robotu. [1]

Model obsahuje ještě třetí kolo umístěné v přední části platformy, které však slouží pouze pro stabilizaci a není urychlováno žádným aktuátorem. (Obr. (10)) Pro účely práce se předpokládá, že se toto kolo otáčí ideálně kolem své osy otáčení a také kolem osy z světového souřadného systému a že nemá žádný vliv na kinematický model robotu.



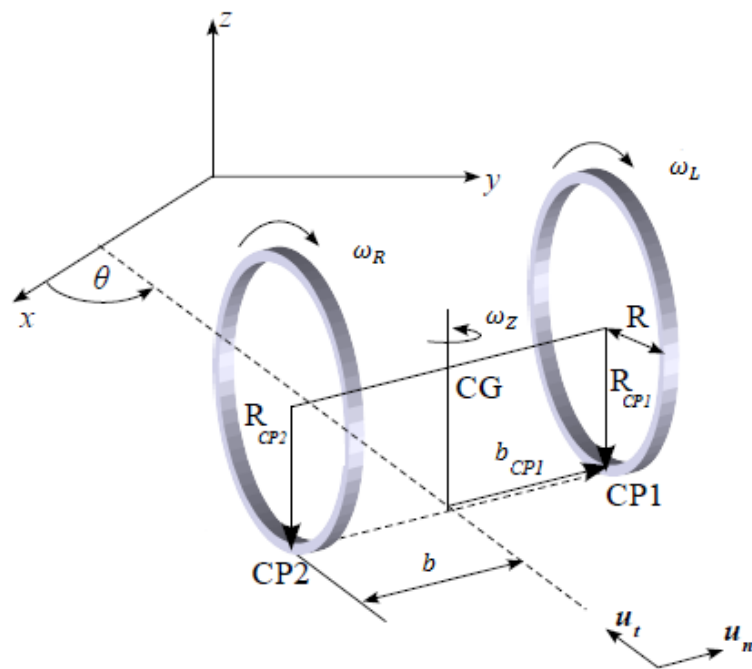
Obr. (10): Přední kolo.

Nyní budou odvozeny pohybové rovnice, kvůli kterým musí být nadefinován světový souřadný systém tak, jak je popsáno na Obr. (9). Stále však platí, že pro účely řízení robotu nejsou absolutní souřadnice v daném prostoru známy (v souladu s 1.1).

2.1 Kinematika

Kinematika zkoumá pohyb hmotného bodu nebo tělesa bez ohledu na příčiny pohybu. Jinak řečeno, kinematika se zabývá popisem polohy, rychlosti, zrychlení eventuálně „změnou zrychlení“ (třetí derivace polohy, tzv. *jerk*) bez ohledu na síly nebo momenty sil, které pohyb způsobily.

Pro odvození kinematických pohybových rovnic [6],[7] je možno využít následující Obr. (11):



Obr. (11): Schéma kolového robotu s vyznačenými důležitými vektory. [7] (s úpravami)

kde jednotlivé vektory lze definovat jako:

$$u_t = (-\cos \theta, -\sin \theta, 0)$$

$$u_n = (-\sin \theta, \cos \theta, 0)$$

$$b_{CP1} = (-b \sin \theta, b \cos \theta, 0) = -b_{CP2}$$

$$\omega_{LW} = (-\omega_L \sin \theta, \omega_L \cos \theta, 0) = \omega_{RW}$$

$$R_{CP1} = (0, 0, -R) = R_{CP2}$$

$$\begin{aligned}
\mathbf{v}_{CP1} &= \mathbf{v}_{CG} + \omega_Z \times \mathbf{b}_{CP1} + \omega_{LW} \times R_{CP1} = \\
&= (\dot{x}, \dot{y}, 0) + (0, 0, \dot{\theta}) \times (-b \sin \theta, b \cos \theta, 0) + (-\omega_L \sin \theta, \omega_L \cos \theta, 0) \times (0, 0, -R) = \\
&= \dot{x} + \dot{y} - b\dot{\theta} \sin \theta e_y - b\dot{\theta} \cos \theta e_x - R\omega_L \sin \theta e_y - R\omega_L \cos \theta e_x = \\
&= (\dot{x} - b\dot{\theta} \cos \theta - R\omega_L \cos \theta, \dot{y} - b\dot{\theta} \sin \theta - R\omega_L \sin \theta, 0) = \mathbf{v}_L
\end{aligned}$$

$$\begin{aligned}
\mathbf{v}_{CP2} &= \mathbf{v}_{CG} + \omega_Z \times \mathbf{b}_{CP2} + \omega_{RW} \times R_{CP2} = \\
&= (\dot{x}, \dot{y}, 0) + (0, 0, \dot{\theta}) \times (b \sin \theta, -b \cos \theta, 0) + (-\omega_L \sin \theta, \omega_L \cos \theta, 0) \times (0, 0, -R) = \\
&= \dot{x} + \dot{y} + b\dot{\theta} \sin \theta e_y + b\dot{\theta} \cos \theta e_x - R\omega_R \sin \theta e_y - R\omega_R \cos \theta e_x = \\
&= (\dot{x} + b\dot{\theta} \cos \theta - R\omega_R \cos \theta, \dot{y} + b\dot{\theta} \sin \theta - R\omega_R \sin \theta, 0) = \mathbf{v}_R
\end{aligned}$$

Jelikož se předpokládá, že kola nemohou prokluzovat do strany, musí platit:

$$\mathbf{v}_L \cdot \mathbf{u}_n \equiv \mathbf{v}_R \cdot \mathbf{u}_n = 0$$

V tomto případě:

$$\begin{aligned}
\mathbf{v}_L \cdot \mathbf{u}_n &= (\dot{x} - b\dot{\theta} \cos \theta - R\omega_L \cos \theta, \dot{y} - b\dot{\theta} \sin \theta - R\omega_L \sin \theta, 0) \cdot (-\sin \theta, \cos \theta, 0) = 0 \\
&= -\dot{x} \sin \theta + b\dot{\theta} \sin \theta \cos \theta + R\omega_L \sin \theta \cos \theta + \dot{y} \cos \theta - b\dot{\theta} \sin \theta \cos \theta - R\omega_L \sin \theta \cos \theta = 0 \equiv \\
&\equiv \mathbf{v}_R \cdot \mathbf{u}_n
\end{aligned}$$

První kinematická rovnice je ve tvaru:

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (1)$$

Dále se předpokládá, že kola nemohou prokluzovat ani ve směru pohybu, čili musí platit:

$$\mathbf{v}_L \cdot \mathbf{u}_t = 0$$

V tomto případě:

$$\begin{aligned}
\mathbf{v}_L \cdot \mathbf{u}_t &= (\dot{x} - b\dot{\theta} \cos \theta - R\omega_L \cos \theta, \dot{y} - b\dot{\theta} \sin \theta - R\omega_L \sin \theta, 0) \cdot (-\sin \theta, -\cos \theta, 0) = 0 \\
&= -\dot{x} \cos \theta + b\dot{\theta} \cos^2 \theta + R\omega_L \cos^2 \theta - \dot{y} \sin \theta + b\dot{\theta} \sin^2 \theta + R\omega_L \sin^2 \theta = 0
\end{aligned}$$

Druhá kinematická rovnice je ve tvaru:

$$\dot{x} \cos \theta + \dot{y} \sin \theta - b\dot{\theta} - R\omega_L = 0 \quad (2)$$

Analogicky lze totéž předpokládat i pro druhé kolo, tedy:

$$v_R \cdot u_t = 0$$

$$v_R \cdot u_t = (\dot{x} + b\dot{\theta} \cos \theta - R\omega_R \cos \theta, \dot{y} + b\dot{\theta} \sin \theta - R\omega_R \sin \theta, 0) \cdot (-\sin \theta, -\cos \theta, 0) = 0$$

$$-\dot{x} \cos \theta - b\dot{\theta} \cos^2 \theta + R\omega_R \cos^2 \theta - \dot{y} \sin \theta - b\dot{\theta} \sin^2 \theta + R\omega_R \sin^2 \theta = 0$$

Třetí kinematická rovnice je ve tvaru:

$$\dot{x} \cos \theta + \dot{y} \sin \theta + b\dot{\theta} - R\omega_R = 0 \quad (3)$$

Pokud bude ještě uvažováno, že osy kol neprocházejí těžištěm robotu [1] (bez odvození), lze pohyb kolového robotu s dvěma nezávisle řízenými koly podle Obr. (9) popsat pomocí následujících kinematických rovnic (někdy též nazývaných jako rovnice omezení):

$$\dot{y} \cos \theta - \dot{x} \sin \theta - d\dot{\theta} = 0 \quad (4)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta + b\dot{\theta} = R\omega_R \quad (5)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta - b\dot{\theta} = R\omega_L \quad (6)$$

kde:

- x, y - je poloha těžiště platformy
- θ - je natočení robotu vzhledem ke kladné části osy x světového souřadného systému
- d - je vzdálenost osy otáčení hnacích kol od s ní rovnoběžné přímky procházející těžištěm robotu
- b - je polovina vzdálenosti mezi poháněnými koly
- R - je poloměr kol
- ω_R - je úhlová rychlost pravého kola
- ω_L - je úhlová rychlost levého kola

Rovnice (4) vyjadřuje kinematické omezení popisující vzájemný vztah rychlostí v osách x a y .

Rovnice (5) a (6) popisují kinematická omezení na neprokluzování levého, respektive pravého kola ve směru pohybu.

V našem případě se předpokládá, že osa otáčení poháněných kol prochází těžištěm platformy, čili parametr $\mathbf{d} = \mathbf{0}$. Při použití uvedeného zjednodušení mohou být kinematické pohybové rovnice zapsány ve tvaru:

$$\begin{aligned}\dot{x} &= \frac{R\omega_R}{2} \cos \theta + \frac{R\omega_L}{2} \cos \theta \\ \dot{y} &= \frac{R\omega_R}{2} \sin \theta + \frac{R\omega_L}{2} \sin \theta \\ \dot{\theta} &= \frac{R\omega_R - R\omega_L}{2b}\end{aligned}\tag{7}$$

Ve stavovém popisu:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{2b} & -\frac{R}{2b} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}\tag{8}$$

Aplikací vztahů:

$$\begin{aligned}\omega_R &= \frac{1}{R}v + \frac{b}{R}\omega \\ \omega_L &= \frac{1}{R}v - \frac{b}{R}\omega\end{aligned}\tag{9}$$

kde:

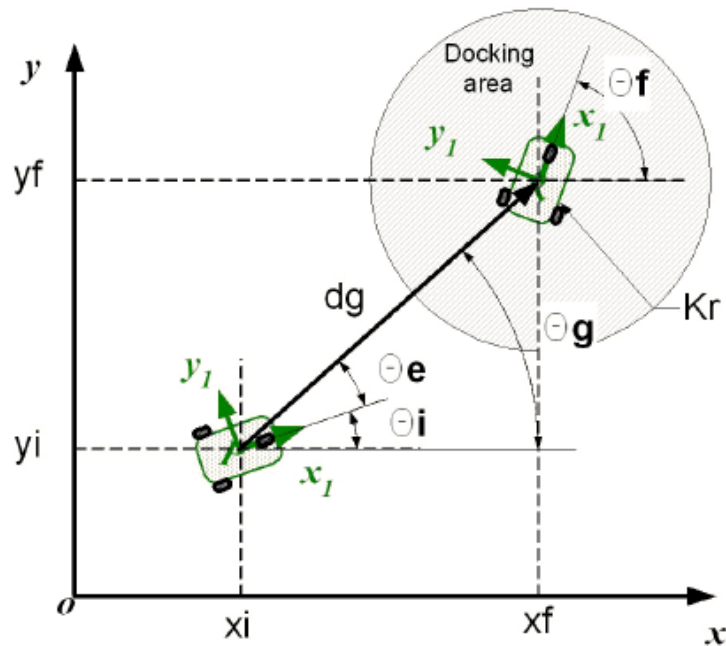
- v – je lineární rychlost těžiště robotu
- ω – je úhlová rychlost těžiště platformy vzhledem ke světovému souřadnému systému

a rozšířením vektoru stavu může být dosaženo následujícího stavového popisu:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \\ 1/R & b/R \\ 1/R & -b/R \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (10)$$

2.2 Návrh řízení předpokládající dokonalou znalost operačního prostoru

Nyní bude uvažována jednoduchá situace, kdy je přesně známa počáteční poloha kolového robotu a je známa i koncová poloha, do které je požadováno přesunout těžiště robotu. [1] Celá situace je znázorněna na Obr. (12):



Obr. (12): Znázornění pohybu robotu z počáteční polohy do libovolné koncové polohy.[1]

kde:

- x_i, y_i - jsou souřadnice počáteční polohy těžiště
- x_f, y_f - jsou souřadnice koncové polohy těžiště
- d_G - je úsečka spojující počáteční a koncovou polohu
- θ_G - je úhel natočení k cíli vzhledem ke kladné části osy x světového souřadného systému
- θ_i - je skutečné natočení robotu vzhledem ke kladné části osy x světového souřadného systému
- θ_e - je tzv. chybový úhel, čili rozdíl úhlu natočení k cíli a aktuálního natočení vozíku

$$\theta_e = \theta_G - \theta_i \quad (11)$$

- K_R - je poloměr pomyslné kružnice se středem v koncové poloze, uvnitř které robot brzdí

Jednoduchými výpočty lze vypočítat velikost úsečky d_G a velikost úhlu θ_G :

$$|d_G| = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2} \quad (12)$$

$$\theta_G = \arctan \frac{y_f - y_i}{x_f - x_i}$$

Nyní jsou uvažovány konstantní parametry:

- v_{max} - je maximální lineární rychlost vozíku
- ω_{max} - je maximální úhlová rychlost vozíku („jak rychle se může otáčet“)

Při daných hodnotách parametrů v_{max} , ω_{max} , K_R je možno definovat jednoduchý zákon řízení (13) [1] a použít ho pro stavový model (10):

$$v = \begin{cases} v_{max} & , |d_G| > K_R \\ \frac{v_{max}}{K_R} \cdot |d_G| & , |d_G| \leq K_R \end{cases} \quad (13)$$

$$\omega = \omega_{max} \sin \theta_e$$

Toto řízení v podstatě zajistí to, že se robot bude pohybovat maximální lineární rychlostí, pokud se bude nacházet mimo vyznačený cílový prostor (*docking area*). Naopak uvnitř bude brzdít. Obdobně pro úhlovou rychlost; ta bude nulová pro nulový chybový úhel, čili pokud bude robot natočen přesně k cíli. Naopak maximální úhlová rychlost bude pro chybový úhel $\pm 90^\circ$. Čím větší chybový úhel, tím více se bude robot točit.

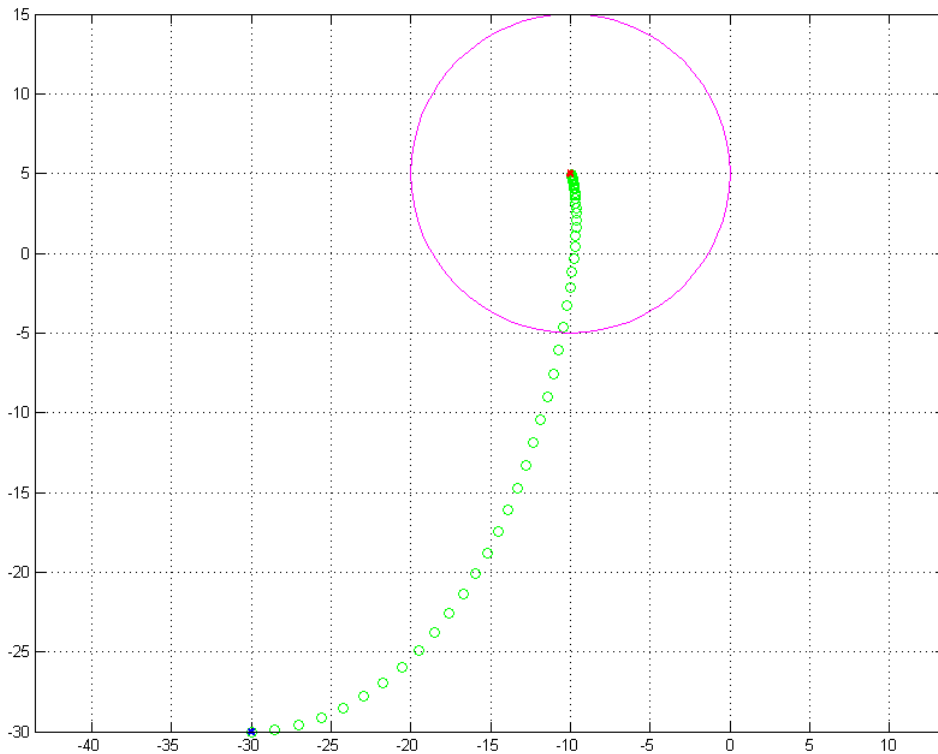
Na Obr. (13) je znázorněna simulace pohybu robotu pro parametry:

$$b = 1, R = 0.1, K_R = 10,$$

$$x_i = -30, y_i = -30, \theta_i = 0^\circ$$

$$x_f = -10, y_f = 5,$$

$$v_{max} = 30, \omega_{max} = \pi$$



Obr. (13): Simulace pohybu robotu pomocí výše uvedeného zákona řízení (13).

Je zřejmé, že robot se opravdu dostane z počátečního stavu do koncového a za znázorněnou hranicí (kružnicí) zpomaluje, až úplně zastaví. Nevýhodou tohoto přístupu je bezesporu to, že není známo vůbec nic o průběhu trajektorie, kterou robot urazí. Nemůže se obecně říct, zda je tento způsob řízení optimální, a to v žádném smyslu. Navíc zde existuje předpoklad na dokonalou znalost prostředí (neměnný světový souřadný systém), čili by bylo nutné použít nějaký navigační systém, nebo by se robot musel pohybovat v předem známém a v nějakém smyslu ohraničeném prostoru, aby bylo možné tento způsob řízení použít. Problémem je i citlivost řízení na volbě v_{max} a ω_{max} . Při špatné volbě těchto parametrů robot nemusí být schopen do cílového prostoru vůbec dorazit. Další části práce se budou věnovat jinému způsobu řízení, který by bylo možné použít i bez dokonalé znalosti okolního prostředí.

2.3 Dynamika

Dynamika zkoumá pohyb hmotného bodu nebo tělesa z hlediska působení sil (momentů sil, energií, hybností, atd...).

Pro výpočet dynamiky vozíku lze využít Lagrangeovy rovnice. Jelikož se ovšem jedná o neholonomní systém, je nutno použít obecnější tvar [5],[7]:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum_j^n \lambda_j a^j + Q_i \quad (14)$$

Rovnice omezení (1), (2), (3) lze přepsat do tvaru výhodnějšího pro výpočet tak, že se provede součet (2) + (3) a rozdíl (2) - (3). Vzniknou následující rovnice omezení:

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (15)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta - \frac{R}{2} (\omega_R + \omega_L) = 0 \quad (16)$$

$$\dot{\theta} + \frac{R}{2b} (\omega_L - \omega_R) = 0 \quad (17)$$

Nechť jsou zobecněné souřadnice systému definovány jako:

$$q_i = (x, y, \theta, \psi_L, \psi_R)^T \quad (18)$$

kde:

- ψ_L, ψ_R - jsou úhly natočení jednotlivých kol

Vektory a^j jsou složeny z parametrů u jednotlivých proměnných v rovnicích (15) - (17) splňujících rovnost:

$$\sum_j^n a^j(q_i) \dot{q}_i = 0 \quad (19)$$

Rovnice (19) reprezentuje obecný popis rovnic omezení.

Konkrétní výpočet parametrů:

$$\begin{aligned}
 a^1 \cdot (\dot{x}, \dot{y}, \dot{\theta}, \omega_L, \omega_R)^T &\Rightarrow a^1 = (-\sin \theta, \cos \theta, 0, 0, 0) \\
 a^2 \cdot (\dot{x}, \dot{y}, \dot{\theta}, \omega_L, \omega_R)^T &\Rightarrow a^2 = \left(\cos \theta, \sin \theta, 0, -\frac{R}{2}, -\frac{R}{2} \right) \\
 a^3 \cdot (\dot{x}, \dot{y}, \dot{\theta}, \omega_L, \omega_R)^T &\Rightarrow a^3 = \left(0, 0, 1, \frac{R}{2b}, -\frac{R}{2b} \right)
 \end{aligned} \tag{20}$$

Vozík je **neholonomní systém**, což znamená, že jeho aktuální poloha je závislá nejenom na hodnotách zobecněných souřadnic, ale i na jejich derivacích. Pokud pro pohybové rovnice (rovnice omezení) platí následující vztah:

$$\sum_j^n \frac{\partial a^j(q_i)}{\partial q_i} \dot{q}_i = 0 \tag{21}$$

pak je daná rovnice holonomní (někdy též nazývaná integrabilní). Pokud je naopak výraz na levé straně rovnice (21) nenulový, daná rovnice je neholonomní (neintegrabilní). V našem případě platí:

$$\begin{aligned}
 \sum_j^n \frac{\partial (-\sin \theta, \cos \theta, 0, 0, 0)}{\partial (x, y, \theta, \psi_L, \psi_R)} (\dot{x}, \dot{y}, \dot{\theta}, \omega_L, \omega_R) &= -\dot{\theta}(\sin \theta + \cos \theta) \neq 0 \\
 \sum_j^n \frac{\partial \left(\cos \theta, \sin \theta, 0, -\frac{R}{2}, -\frac{R}{2} \right)}{\partial (x, y, \theta, \psi_L, \psi_R)} (\dot{x}, \dot{y}, \dot{\theta}, \omega_L, \omega_R) &= \dot{\theta}(\sin \theta + \cos \theta) \neq 0 \\
 \sum_j^n \frac{\partial \left(0, 0, 1, \frac{R}{2b}, -\frac{R}{2b} \right)}{\partial (x, y, \theta, \psi_L, \psi_R)} (\dot{x}, \dot{y}, \dot{\theta}, \omega_L, \omega_R) &= 0
 \end{aligned} \tag{22}$$

tedy rovnice (15), (16) jsou neholonomní a rovnice (17) je holonomní.

Pro výpočet dynamických rovnic definujeme Lagrangian:

$$L = T - V \quad (23)$$

kde T je kinetická energie soustavy, V je potenciální energie soustavy.

V našem případě:

$$L = T_{trans} + T_{RR} + T_{RW} \quad (24)$$

kde:

- T_{trans} - je translační energie robotu
- T_{RR} - je rotační energie robotu
- T_{RW} - je rotační energie kol

Platí:

$$L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_R\omega^2 + \frac{1}{2}I_W(\omega_R^2 + \omega_L^2) \quad (25)$$

kde:

- \dot{x}, \dot{y} - jsou rychlosti těžiště v příslušných osách
- m - je hmotnost celého robotu
- I_R - je moment setrvačnosti robotu
- I_W - je moment setrvačnosti kola
- $\omega = \dot{\theta}$ - je úhlová rychlost robotu (platformy)
- $\omega_R = \dot{\theta}_R$, respektive $\omega_L = \dot{\theta}_L$ - jsou úhlové rychlosti jednotlivých kol

Dosazením do (14) je možno dostat:

$$\begin{aligned} (m\ddot{x}, m\ddot{y}, I_R\ddot{\theta}, I_W\dot{\omega}_L, I_W\dot{\omega}_R)^T - 0 &= \lambda_1(-\sin\theta, \cos\theta, 0, 0, 0)^T + \\ &+ \lambda_2\left(\cos\theta, \sin\theta, 0, -\frac{R}{2}, -\frac{R}{2}\right)^T + \lambda_3\left(0, 0, 1, \frac{R}{2b}, -\frac{R}{2b}\right)^T \\ &+ M_L(0, 0, 0, 1, 0)^T + M_R(0, 0, 0, 0, 1)^T \end{aligned} \quad (26)$$

Přepsáním lze získat soustavu rovnic:

$$\begin{aligned}
 m\ddot{x} &= -\lambda_1 \sin \theta + \lambda_2 \cos \theta \\
 m\ddot{y} &= \lambda_1 \cos \theta + \lambda_2 \sin \theta \\
 I_R \ddot{\theta} &= \lambda_3 \\
 I_W \dot{\omega}_L &= M_L - \frac{R}{2} \lambda_2 + \frac{R}{2b} \lambda_3 \\
 I_W \dot{\omega}_R &= M_R - \frac{R}{2} \lambda_2 - \frac{R}{2b} \lambda_3
 \end{aligned} \tag{27}$$

Řešením této soustavy [7] je možné získat:

$$\begin{bmatrix} \frac{mR}{2} \cos \theta & \frac{mR}{2} \sin \theta & -\frac{R}{2b} I_R \\ \frac{mR}{2} \cos \theta & \frac{mR}{2} \sin \theta & \frac{R}{2b} I_R \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} I_W & 0 \\ 0 & I_W \end{bmatrix} \begin{bmatrix} \dot{\omega}_R \\ \dot{\omega}_L \end{bmatrix} = \begin{bmatrix} M_R \\ M_L \end{bmatrix} \tag{28}$$

Pro úpravu tohoto výrazu je možno použít formální časové derivace rovnic (8):

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{2b} & -\frac{R}{2b} \end{bmatrix} \begin{bmatrix} \dot{\omega}_R \\ \dot{\omega}_L \end{bmatrix} + \begin{bmatrix} -\frac{R}{2} \dot{\theta} \sin \theta & -\frac{R}{2} \dot{\theta} \sin \theta \\ \frac{R}{2} \dot{\theta} \cos \theta & \frac{R}{2} \dot{\theta} \cos \theta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \tag{29}$$

Po dosazení (29) do (28):

$$\left(\frac{mR^2}{4} + \frac{I_R R^2}{4b^2} + I_W \right) \dot{\omega}_R + \left(\frac{mR^2}{4} - \frac{I_R R^2}{4b^2} \right) \dot{\omega}_L = M_R \tag{30}$$

$$\left(\frac{mR^2}{4} + \frac{I_R R^2}{4b^2} + I_W \right) \dot{\omega}_L + \left(\frac{mR^2}{4} - \frac{I_R R^2}{4b^2} \right) \dot{\omega}_R = M_L \tag{31}$$

Dynamické rovnice lze tedy zapsat následujícím způsobem:

$$A\dot{\omega}_R + B\dot{\omega}_L = M_R \tag{32}$$

$$A\dot{\omega}_L + B\dot{\omega}_R = M_L$$

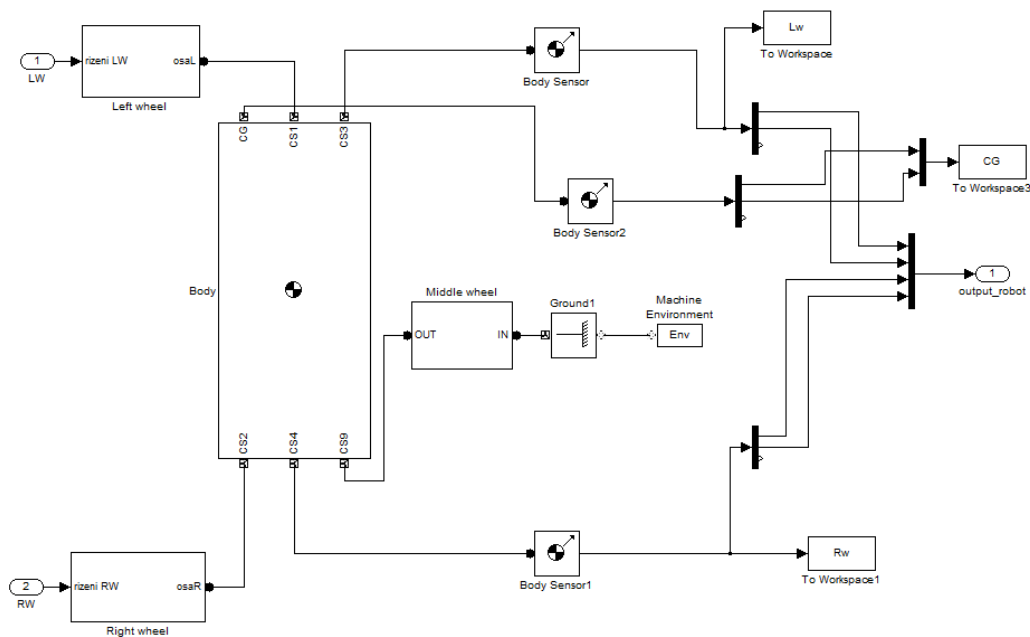
kde:

$$A = \frac{mR^2}{4} + \frac{I_R R^2}{4b^2} + I_W \tag{33}$$

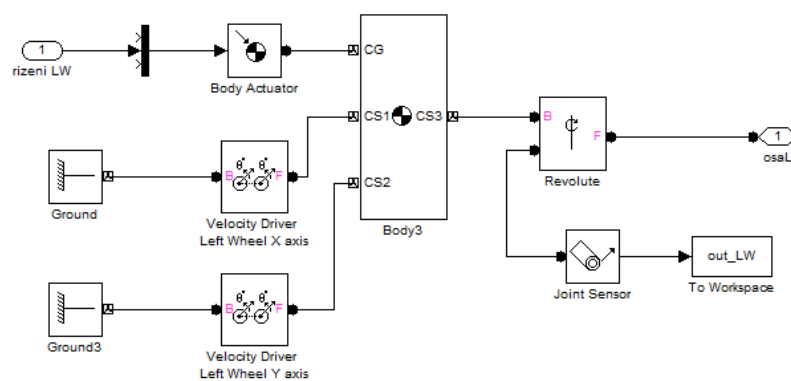
$$B = \frac{mR^2}{4} - \frac{I_R R^2}{4b^2}$$

2.4 Model kolového robotu s nezávisle řízenými koly

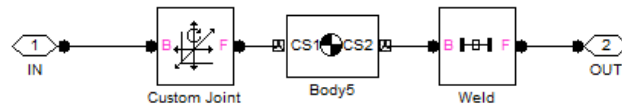
Tento model byl vytvořen v programu Simulink/SimMechanics softwarového nástroje Matlab. Ve schématu se vyskytují tři subsystémy popisující levé, pravé a střední kolo. Platformu robotu symbolizuje jediný funkční blok *Body*. Subsystém levého kola je formálně shodný se subsystémem pravého kola. Pohyb robotu v prostoru umožňují bloky *Velocity Driver*. Níže jsou uvedena schémata jednotlivých subsystémů (Obr. (14-16)). Podrobný popis všech funkčních bloků viz [2].



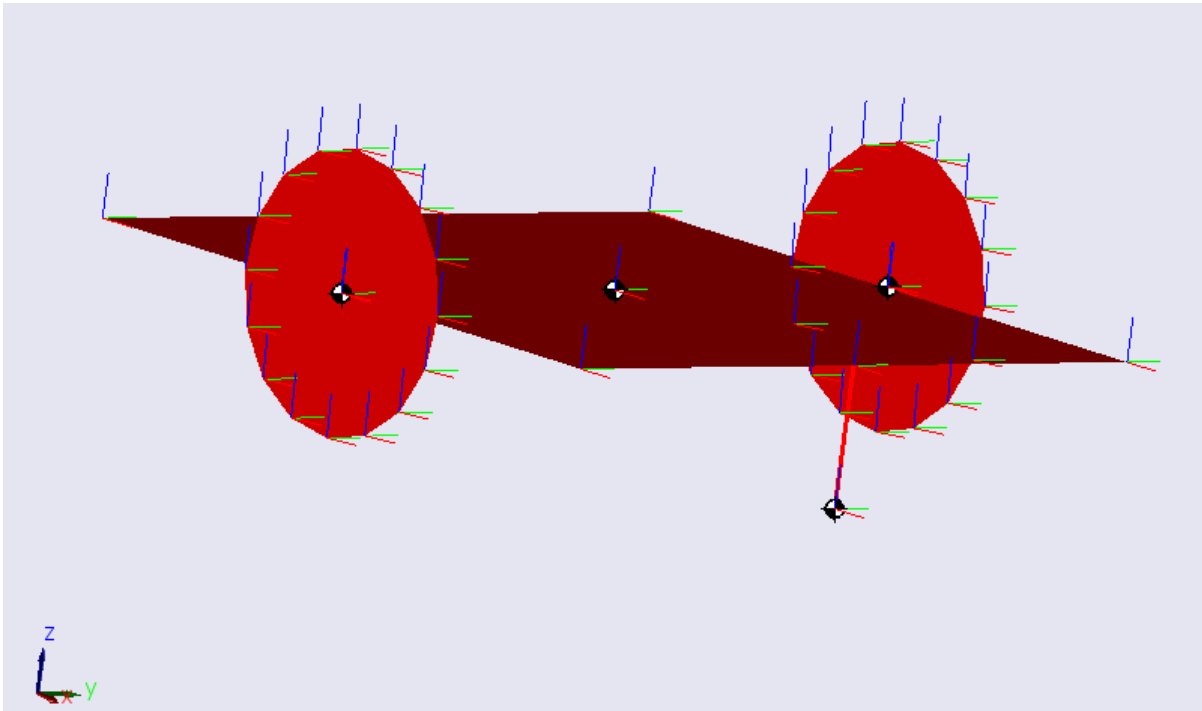
Obr. (14): Schéma zapojení modelu kolového robotu.



Obr. (15): Subsystém levého kola.



Obr. (16): Subsystem středního kola.



Obr. (17): Kolový robot v programu SimMechanics.

Model je vytvořen s přihlédnutím na reálné využití. Parametry (viz podkapitola 2.1) jsou zvoleny následovně:

$$b = 0,05m$$

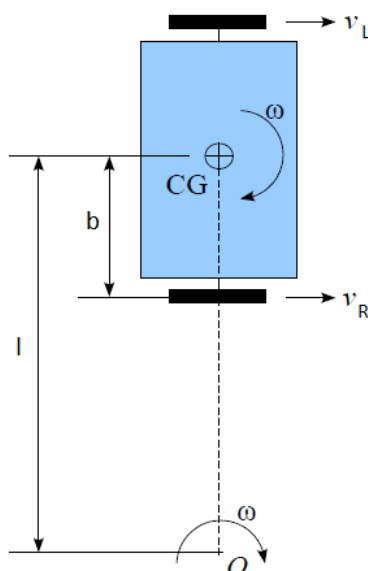
$$R = 0,02m$$

3 Použité metody potřebné pro návrh řízení

V této kapitole budou popsány obecné principy a teoretické metody potřebné pro budoucí návrh řízení (viz Kapitola 3).

3.1 Pohyb po kružnici

Nejprve je důležité si připomenout základní vztahy pro pohyb bodu po kružnici.



Obr. (18): Schéma pohybu robotu po kružnici. [7]

- v_L - je lineární rychlost levého kola
- v_R - je lineární rychlost pravého kola
- v - je lineární rychlost těžiště
- b - je vzdálenost těžiště od kola
- l - je poloměr kružnice, po které se těžiště robotu pohybuje
- ω - je úhlová rychlost robotu

Pro lineární rychlosti těžiště robotu a jednotlivých kol platí:

$$v = l\omega$$

$$v_L = (l + b)\omega \quad (34)$$

$$v_R = (l - b)\omega$$

Úhlová rychlost je pro všechny případy stejná. Z toho plyne:

$$\frac{v}{l} = \frac{v_R}{l - b} = \frac{v_L}{l + b}$$

$$v_R = v \cdot \frac{l - b}{l} \quad (35)$$

$$v_L = v \cdot \frac{l + b}{l}$$

Na základě jednoduchých vztahů a (35) je nyní možno vypočítat požadovanou úhlovou rychlost jednotlivých kol:

$$\omega_R = \frac{v_R}{R} = \frac{v}{R} \cdot \frac{l - b}{l} \quad (36)$$

$$\omega_L = \frac{v_L}{R} = \frac{v}{R} \cdot \frac{l + b}{l}$$

Analogicky můžeme postupovat i pro určení lineárního a úhlového zrychlení:

$$a = l\dot{\omega}$$

$$a_L = (l + b)\dot{\omega} \quad (37)$$

$$a_R = (l - b)\dot{\omega}$$

$$\frac{a}{l} = \frac{a_R}{l-b} = \frac{a_L}{l+b}$$

$$a_R = a \cdot \frac{l-b}{l} \tag{38}$$

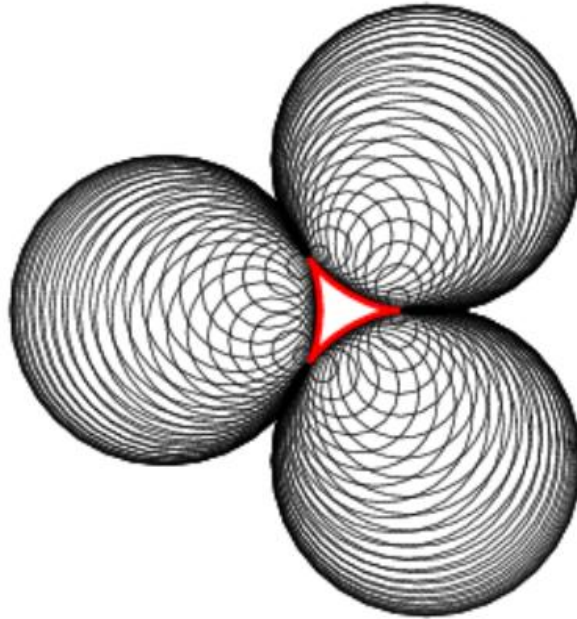
$$a_L = a \cdot \frac{l+b}{l}$$

$$\dot{\omega}_R = \frac{a_R}{R} = \frac{a}{R} \cdot \frac{l-b}{l} \tag{39}$$

$$\dot{\omega}_L = \frac{a_L}{R} = \frac{a}{R} \cdot \frac{l+b}{l}$$

3.2 Oskulační kružnice

Oskulační kružnice popisuje míru zakřivení křivky v daném bodě. Můžeme ji definovat jako kružnici, která prochází daným bodem a několika sousedními nekonečně blízkými body křivky a tím ji aproximuje. [3] Pro lepší představu je uvedená situace znázorněna na následujícím obrázku (Obr. (19)):



Obr. (19): Oskulační kružnice zvýrazněného obrazce. [3]

Křivka v rovině je dána parametrickými rovnicemi:

$$\begin{aligned}x &= x(t) \\ y &= y(t)\end{aligned}\tag{40}$$

V daném bodě ji lze aproximovat následujícím způsobem [3]. Nejprve definujeme výraz:

$$\kappa = \frac{d\Phi}{ds} = \frac{\frac{d\Phi}{dt}}{\frac{ds}{dt}}\tag{41}$$

kde:

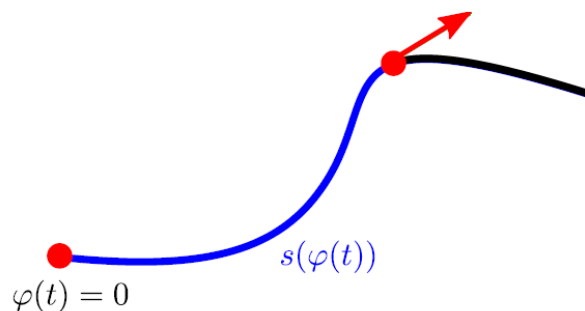
- Φ - je tečný úhel vzhledem ke kladné části osy x
- s - je délka křivky

Délku křivky lze obecně definovat jako [8]:

$$s(\varphi(t)) = \int_0^{\varphi(t)} \sqrt{\left(\frac{\partial\Phi_1(\varphi(t))}{\partial\varphi(t)}\right)^2 + \left(\frac{\partial\Phi_2(\varphi(t))}{\partial\varphi(t)}\right)^2 + \left(\frac{\partial\Phi_3(\varphi(t))}{\partial\varphi(t)}\right)^2} \quad (42)$$

kde je každý bod definován jako:

$$X = \begin{bmatrix} \Phi_1(\varphi(t)) \\ \Phi_2(\varphi(t)) \\ \Phi_3(\varphi(t)) \end{bmatrix} \quad (43)$$



Obr. (20): Parametrická křivka a její délka.

V tomto případě je však možné se omezit pouze na dvourozměrný prostor. Platí tedy:

$$\frac{ds}{dt} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \quad (44)$$

Dosazením (44) do (41) lze získat:

$$\kappa = \frac{\frac{d\Phi}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}} = \frac{\frac{d\Phi}{dt}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \quad (45)$$

Dále je zřejmé, že:

$$\tan \Phi = \frac{dy}{dx} = \frac{\dot{y}}{\dot{x}} \quad (46)$$

Derivací lze získat následující vztah:

$$\frac{d}{dt} \tan \Phi = \sec^2 \Phi \cdot \frac{d\Phi}{dt} = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2} \quad (47)$$

kde

- $\sec(*)$ - je funkce sekans definovaná jako: $\sec(*) = \frac{1}{\cos(*)}$

Postupnými úpravami lze dostat:

$$\begin{aligned} \frac{d\Phi}{dt} &= \frac{1}{\sec^2 \Phi} \cdot \frac{d}{dt} \tan \Phi = \frac{1}{1 + \tan^2 \Phi} \cdot \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2} = \frac{1}{1 + \frac{\dot{y}^2}{\dot{x}^2}} \cdot \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2} \\ \frac{d\Phi}{dt} &= \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \end{aligned} \quad (48)$$

Dosazením (48) do (45):

$$\kappa = \frac{\frac{d\Phi}{dt}}{\sqrt{\dot{x}^2 + \dot{y}^2}} = \frac{\frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \quad (49)$$

V daném bodě lze tedy křivku aproximovat následujícím výrazem:

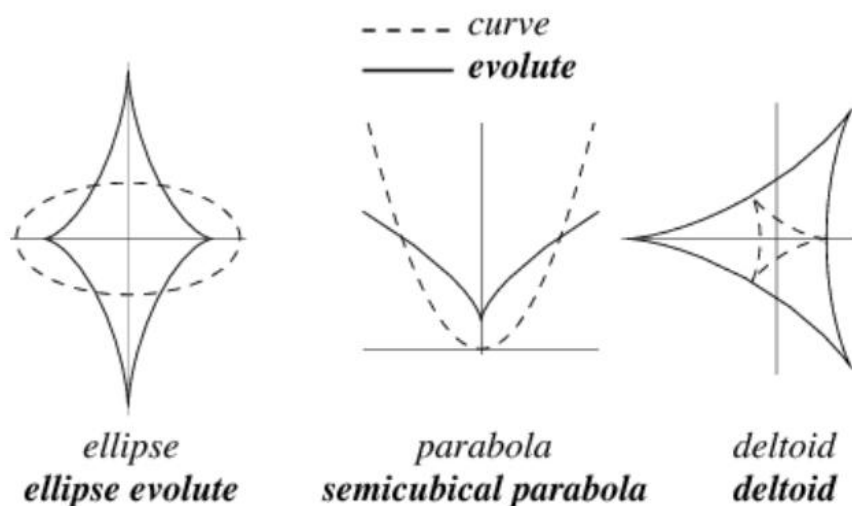
$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\sqrt{(\dot{x}^2 + \dot{y}^2)^3}} \quad (50)$$

kde výrazy $\dot{x}, \ddot{x}, \dot{y}, \ddot{y}$ jsou časové derivace parametrických rovnic daných v (40).

Poloměr křivosti v daném bodě, někdy také nazýván poloměr oskulační kružnice, se může snadno vypočítat podle vztahu:

$$l = \frac{1}{|\kappa|} = \frac{\sqrt{(\dot{x}^2 + \dot{y}^2)^3}}{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|} \quad (51)$$

K výpočtu středu oskulační kružnice je potřeba vysvětlit pojem *evoluta*. Evoluta rovinné křivky je obalová křivka všech jejích normál a zároveň geometrické místo všech jejích středů křivosti.



Obr. (21): Příklady některých křivek a jejich evolut.[3] (čárkovaně je znázorněna křivka, plnou čarou její evoluta)

Pro výpočet středu oskulační kružnice lze s výhodou využít obecné rovnice evoluty:

$$\begin{aligned}x_0 &= x - l \sin \Phi \\y_0 &= y + l \cos \Phi\end{aligned}\tag{52}$$

kde:

- (x_0, y_0) - jsou souřadnice bodů evoluty
- l - je poloměr oskulační kružnice
- Φ - je tečný úhel vzhledem ke kladné části osy x

Pro libovolnou křivku $r(t)$ určitě platí [3]:

$$\frac{\dot{r}}{|\dot{r}|} = \begin{bmatrix} \cos \Phi \\ \sin \Phi \end{bmatrix}\tag{53}$$

Navíc ze vztahů pro výpočet úhlů v pravoúhlém trojúhelníku plyne:

$$\begin{aligned}\cos \Phi &= \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \sin \Phi &= \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}}\end{aligned}\tag{54}$$

Kombinací (51), (52) a (54) lze tedy získat závěr, že souřadnice středu křivosti (středu oskulační kružnice) ležící na evolutě je možno vypočítat podle vzorců:

$$\begin{aligned}x_l &= x - \frac{(\dot{x}^2 + \dot{y}^2)\dot{y}}{\dot{x}\ddot{y} - \dot{y}\ddot{x}} \\ y_l &= y + \frac{(\dot{x}^2 + \dot{y}^2)\dot{x}}{\dot{x}\ddot{y} - \dot{y}\ddot{x}}\end{aligned}\tag{55}$$

3.3 PD regulátor

Pro řízení pohybu robotu po dané křivce je v této práci použit paralelní PD (proporcionálně-derivační) regulátor. V této podkapitole jsou popsány některé jeho obecné vlastnosti ([9],[10]). Význam jednotlivých složek na kvalitu regulace:

- P - proporcionální složka, která reprezentuje nedynamickou část regulátoru; zvětšováním P-složky se zlepšuje přesnost regulace, zvyšuje se rychlost odezvy, ale zvětšuje se přeregulování a snižuje se robustnost ve stabilitě
- D - derivační složka, která reprezentuje dynamickou část regulátoru; je úměrná derivaci regulační odchylky, zvyšuje rychlost odezvy, zmenšuje přeregulování, pro malé hodnoty zvyšuje robustnost ve stabilitě

Ideální PD regulátor generuje řízení, které je realizováno dvěma složkami podle vztahu:

$$u(t) = K \left[e(t) + \frac{T_D de(t)}{dt} \right] \quad (56)$$

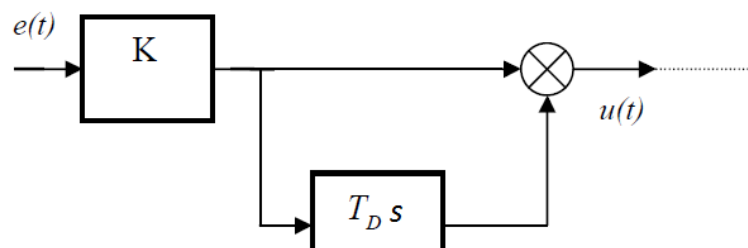
Po aplikaci Laplaceovy transformace:

$$U(s) = K[E(s) + T_D sE(s)] \quad (57)$$

Přenos ideálního PD regulátoru je tedy ve tvaru:

$$F_{PD}(s) = \frac{U(s)}{E(s)} = K(1 + T_D s) = K + K_D s \quad (58)$$

Schéma zapojení PD regulátoru (Obr. (22)):



Obr. (22): Ideální PD regulátor.

Jelikož takto navržená derivační složka činí velké problémy při skokových změnách vstupního signálu, vysokofrekvenčních poruchách a ve skutečnosti je fyzikálně nerealizovatelná, často je modifikována tzv. filtrovanou derivací. Přenos reálného regulátoru je po této náhradě ve tvaru:

$$F_{PD}(s) = K + \frac{K_D s}{\tau s + 1} \quad (59)$$

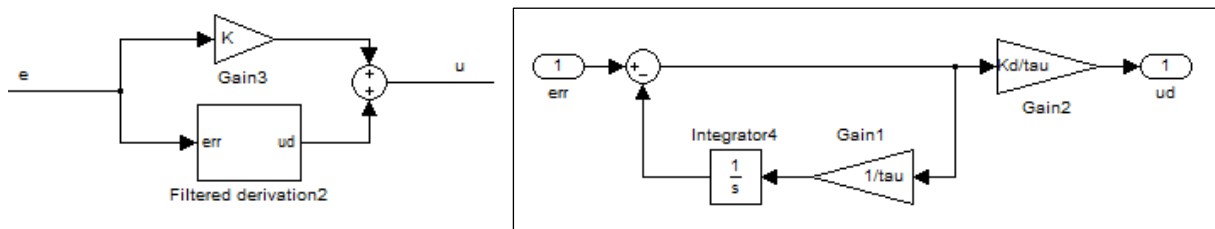
kde:

$$\tau = \frac{T_D}{3 \div 20} \quad (60)$$

Tvar derivační složky použitý pro simulaci:

$$\frac{K_D s}{\tau s + 1} = \frac{K_D}{\tau + \frac{1}{s}} = \frac{\frac{K_D}{\tau}}{1 + \frac{1/\tau}{s}} \quad (61)$$

Realizace PD regulátoru s filtrovanou derivací v programu Simulink (Obr. (23)):



Obr. (23): PD regulátor v programu Simulink (vlevo) a subsystém filtrované derivace (vpravo).

3.4 Geometrické transformace v prostoru

Pro experimentální účely (viz Kapitola 3) je nezbytné, aby byly vysvětleny některé metody a principy užívané v počítačové grafice a projektivní geometrii. Jedná se především o geometrické transformace objektů v rovině a prostoru.

Celou problematiku lze popsat pomocí tzv. *homogenní transformační matice*, která je definovaná jako:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_1 \\ r_{21} & r_{22} & r_{23} & d_2 \\ r_{31} & r_{32} & r_{33} & d_3 \\ p_1 & p_2 & p_3 & g \end{bmatrix} = \begin{bmatrix} R & d \\ p^T & g \end{bmatrix} \quad (62)$$

kde:

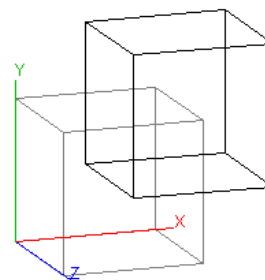
- R - je matice rotace
- d - je vektor posunutí od počátku souřadného systému
- p - je vektor ovlivňující úběžníky (perspektivu)
- g - je číslo ovlivňující zvětšení nebo zmenšení objektu

Výhodou popisu pomocí homogenních transformačních matic je především sloučení rotace a translace do jednoho výrazu, se kterým se dá pracovat jako s celkem. Na tomto základě jsou definované úmluvy (Denavit-Hartenbergova, Khalil-Kleinfingerova) pro popis sériových a paralelních manipulátorů, ve kterých je tento popis výhodný pro definování souřadných systémů jednotlivých ramen a které jsou hojně používány při výpočtech přímých a inverzních kinematických/geometrických úloh.

Příklady některých základních operací s objektem ve tvaru krychle s využitím homogenní transformační matice (použit pravotočivý systém):

➤ **posunutí**

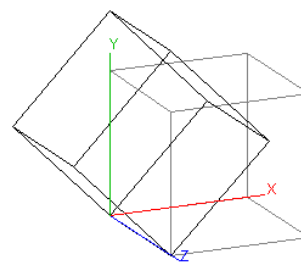
$$T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Obr. (24): Homogenní transformační matice a grafické znázornění posunutí objektu o hodnotu 1 po ose x a po ose y.[11]

➤ **rotace**

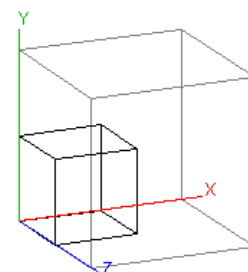
$$T = \begin{bmatrix} 0,707 & -0,707 & 0 & 0 \\ 0,707 & 0,707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Obr. (25): Homogenní transformační matice a grafické znázornění rotace objektu o 45° kolem osy z v kladném smyslu otáčení.[11]

➤ **zmenšení/zvětšení**

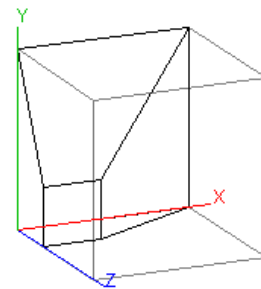
$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$



Obr. (26): Homogenní transformační matice a grafické znázornění zmenšení objektu na polovinu.[11]

➤ *perspektiva (zkosení)*

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & 1 \end{bmatrix}$$



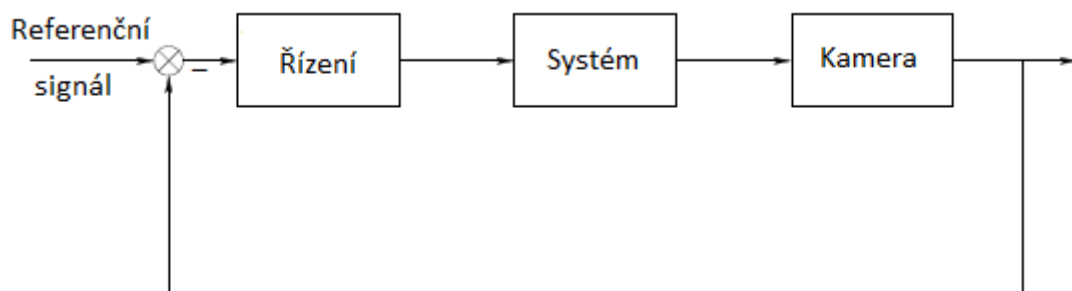
Obr. (27): Homogenní transformační matice a grafické znázornění zkosení objektu podél osy z.[11]

3.5 Vizualní zpětná vazba

V této podkapitole budou přehledově popsány některé principy řízení využívající vizuální zpětnou vazbu (*visual-based robot control*, *visual servoing*) [12].

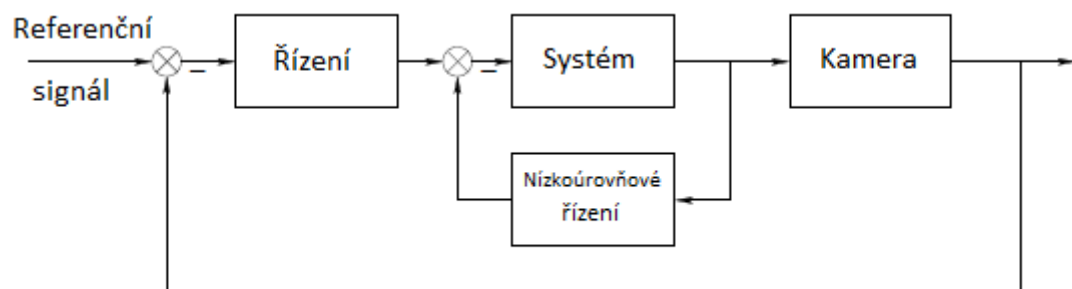
Řízení pomocí vizuální zpětné vazby má mnoho aplikací a může být použito pro řízení různých dynamických systémů, jako jsou například manipulátory, letadla, ponorky nebo mobilní kolové roboty. Takto konstruované systémy se klasifikují podle počtu kamer, podle umístění kamer vzhledem k robotu, podle definice chybové funkce apod.

V praxi se používají především dvě schémata řízení pomocí vizuální zpětné vazby. První se nazývá *přímé řízení využívající vizuální zpětnou vazbou* (*direct visual servoing*) (Obr. (28)):



Obr. (28): Schéma přímého řízení využívající vizuální zpětnou vazbu. [12]

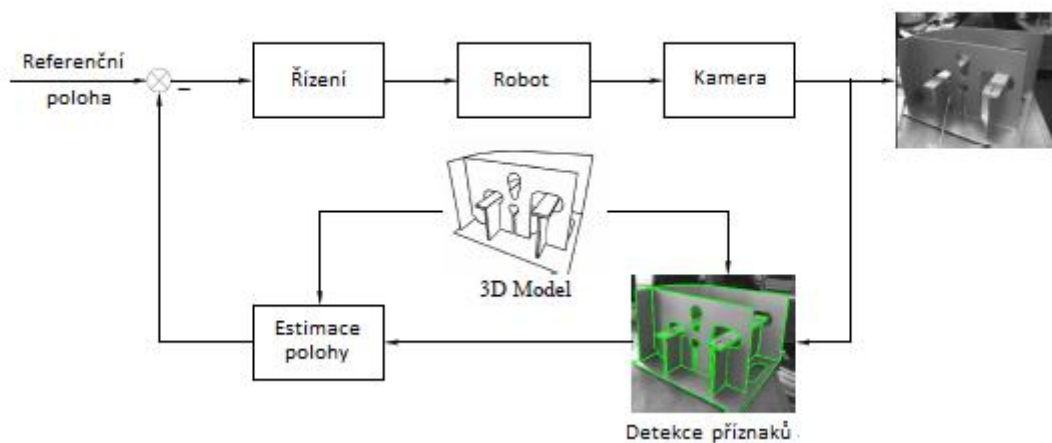
V tomto schématu blok *Vision-based Control* generuje přímo vstup do řízeného systému. Naproti tomu druhé řídicí schéma využívá vnitřní smyčku s nízkourovňovým regulátorem, kde blok *Vision-based Control* generuje jen referenční řízení. Tato architektura se nazývá *nepřímé řízení využívající vizuální zpětnou vazbou* (*indirect visual servoing*) (Obr. (29)):



Obr. (29): Schéma nepřímého řízení využívající vizuální zpětnou vazbu. [12]

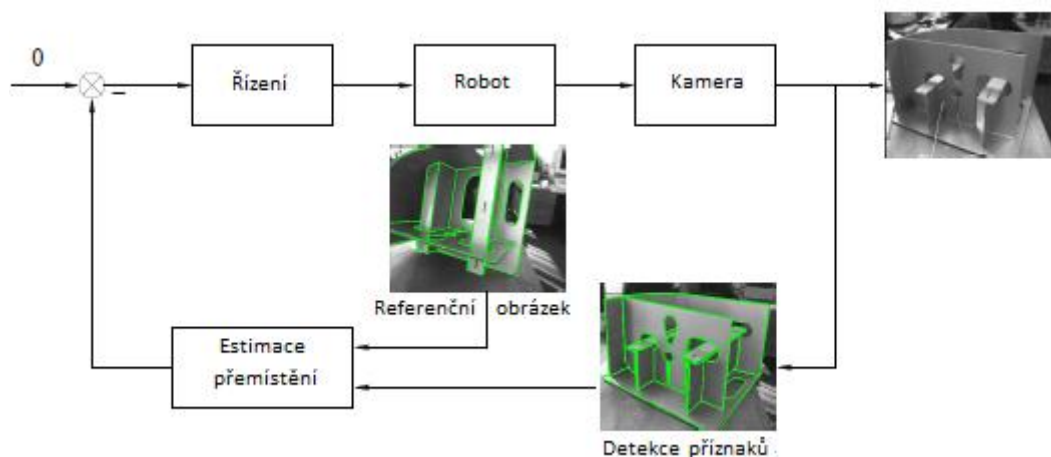
Vizuální zpětná vazba se může použít jak pro řízení robotů, tak například i pro vizuální kontrolu výrobků. Mezi známé aplikace patří například v automobilovém průmyslu hojně používaná vizuální kontrola součástek nově vyrobeného motoru. Všechny aplikace však využívají některou z následujících metod zpracování získaných dat.

První z nich je řízení s vizuální zpětnou vazbou využívající 3D model objektu (*Model-based 3D Visual Servoing*). Jak již vyplývá z názvu, při řízení je využit 3D model objektu, jehož parametry jsou dokonale známé (Obr. (30)):



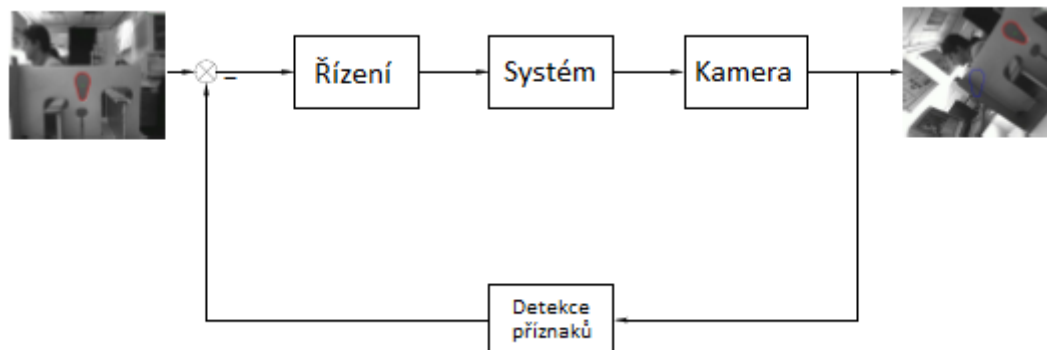
Obr. (30): Řízení s vizuální zpětnou vazbou využívající 3D model objektu. [12]

Další z metod je řízení s vizuální zpětnou vazbou nevyužívající 3D model objektu (*Model-free 3D Visual Servoing*). Pro získání rotace a posunutí již není použit 3D model, jako v předchozím případě, nýbrž referenční obrázek (Obr. (31)):



Obr. (31): Řízení s vizuální zpětnou vazbou nevyužívající 3D model objektu. [12]

Poslední uvedená metoda používá pro určení regulačních odchylek pouze data získaná z 2D obrázku. Nazývá se *řízení s vizuální zpětnou vazbou ve 2D (2D Visual Servoing, Image-based Visual Servoing)* (Obr. (32)):



Obr. (32): Řízení s vizuální zpětnou vazbou ve 2D. [12]

V této práci bude použita varianta poslední uvedené metody. Pro určení regulačních odchylek však nebude použit referenční obrázek, ale geometrický střed obrázku (viz 5.5).

Smyslem této podkapitoly bylo uvést některé principy pro řízení systémů pomocí vizuální zpětné vazby. Těžištěm práce však není detailní popis funkce jednotlivých algoritmů a metod. Reference pro hlubší pochopení problematiky lze nalézt v [12].

Kapitola 2

4 Počítačové vidění

Počítačové vidění (*computer vision*) je odvětví výpočetní techniky a vývoje softwaru zabývající se vytvářením zařízení schopných získávat informaci ze zachyceného obrazu. Využití nalezne především v oblastech řízení procesů, detekce jevů, modelování objektů a prostředí nebo interakce člověka s počítačem. [5]

4.1 Barevné modely RGB a HSV

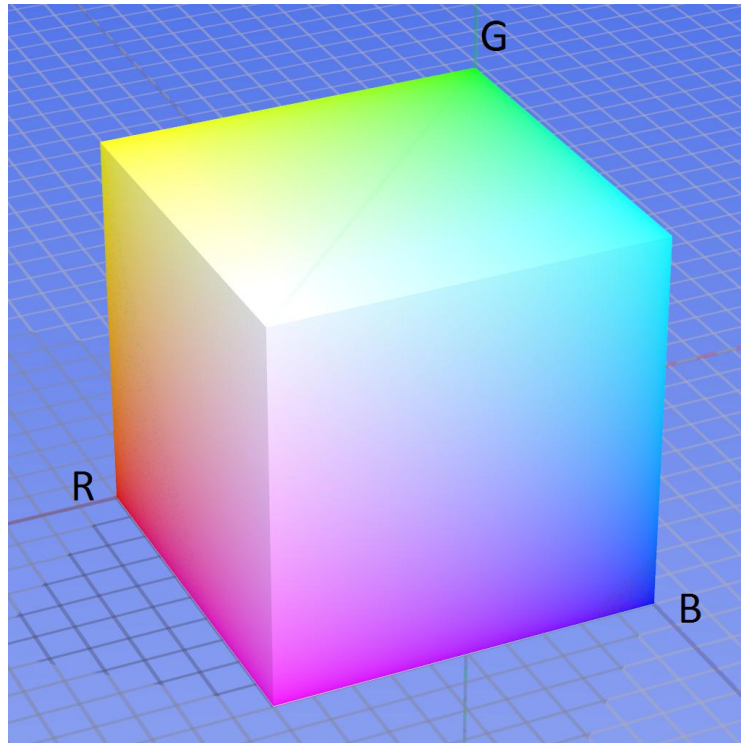
Pro úspěšné rozpoznání barevné křivky na obrázku je důležité správně definovat a reprezentovat jednotlivé barvy a také jejich odstíny. K tomuto účelu slouží tzv. *barevné modely*. [4] Tato část práce se bude věnovat konkrétně dvěma modelům:

- **RGB** (Red, Green, Blue)
- **HSV** (Hue, Saturation, Value)

Model RGB je příkladem součtové (aditivní) reprezentace barev. Výsledná barva vznikne namícháním tří základních složek:

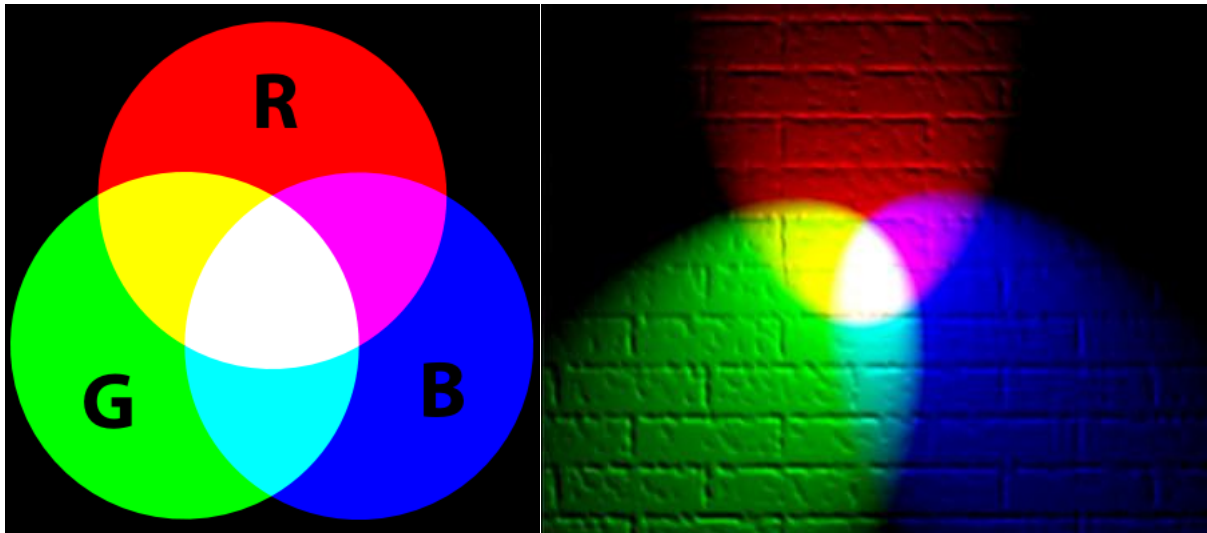
- **R** (*red, červená*) o předepsané vlnové délce $\lambda = 630nm$
- **G** (*green, zelená*) o předepsané vlnové délce $\lambda = 530nm$
- **B** (*blue, modrá*) o předepsané vlnové délce $\lambda = 450nm$

Hodnota jednotlivých složek udává míru jasu světla dané barvy. Způsob, jakým se jednotlivé odstíny mění v závislosti na hodnotách parametrů R, G a B, je možno znázornit např. pomocí krychle (Obr. (33)), kde označené nesousední vrcholy představují základní barvy modelu.



Obr. (33): Reprezentace barevného modelu RGB do krychle.[5]

Jakým způsobem vznikají jednotlivé odstíny a barvy pomocí aditivního míchání v teorii a praxi je znázorněno na následujících obrázcích (Obr. (34)):



Obr. (34): Znázornění aditivního míchání barev.[5]

Model HSV popisuje barvu nikoliv pomocí trojice základních barev, ze kterých je složena, nýbrž pomocí trojice složek, které popisují spíše vlastnosti dané barvy. Konkrétně se jedná o následující složky:

- **H** (*hue, barevný tón*)
- **S** (*saturation, sytost barvy*)
- **V** (*value, hodnota jasu*)

K modelu HSV ještě několik poznámek:

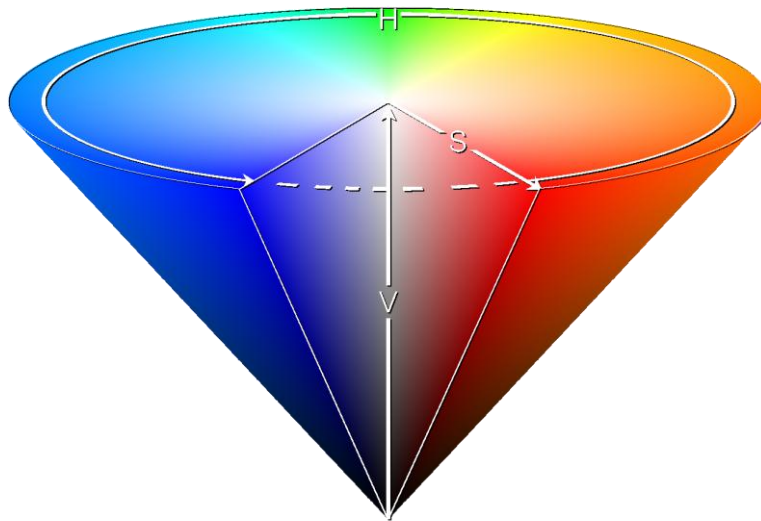
Hue – udává barevný tón, je to barva odražená nebo procházející objektem.

Saturation – udává sytost barvy, příměs jiné barvy, někdy také čistotu barvy; představuje poměr mezi bílou a sytou barvou.

Value - hodnota jasu, relativní světlost nebo tmavost barvy; vyjadřuje, kolik světla barva odráží.

V použité práci bylo pracováno se softwarovým nástrojem Matlab, který všechny tyto parametry (stejně jako parametry modelu RGB) zobrazuje do intervalu 0 až 1. Možné jsou však i reprezentace pomocí úhlu v intervalu 0° až 360° v případě parametru *Value* modelu HSV, bitové reprezentace základních barev v intervalech od 0 do 255 (8 bitů), eventuálně od 0 do 65535 (16 bitů) modelu RGB nebo reprezentace v procentech, aj.

Způsob, jakým se jednotlivé odstíny mění v závislosti na hodnotách parametrů H, S a V, lze tentokrát znázornit např. pomocí kuželu (Obr. (35)):



Obr. (35): Repräsentace barevného modelu HSV pomocí kuželu. [5]

Pro oba modely platí, že obrázek je vlastně tenzor (třírozměrná matice) o rozměrech $(x,y,3)$, ve kterém každému bodu (pixelu) o souřadnicích $[x,y]$ přísluší tři čísla definující jeho barvu.

Ještě pro úplnost je zde uvedena tabulka, ve které jsou zřejmé hodnoty parametrů barevných modelů RGB a HSV pro vybrané barvy (Tabulka (1)):

	R	G	B	H	S	V
bílá	1	1	1	0	0	1
modrá	0	0	1	0,666	1	1
zelená	0	1	0	0,333	1	1
červená	1	0	0	0	1	1
černá	0	0	0	0	0	0

Tabulka (1): Hodnoty parametrů pro vybrané barvy.

Výhodou modelu HSV je především to, že není tak citlivý např. na změnu přisvětlení. Zatímco lehce odlišné světelné podmínky způsobí u modelu RGB to, že původně žlutá barva začne být klasifikována jako např. šedá. Barevný model HSV je v tomto ohledu daleko více robustní.

Ještě je potřebné dodat, jakým způsobem jsou popisovány obrázky v prostředí softwarového nástroje Matlab. Osy souřadného systému obrázku totiž tvoří levotočivý systém, což s sebou nese požadavky na přepočítávání některých údajů v celém algoritmu rozpoznávání a řízení.



Obr. (36): Definice souřadného systému obrázku.

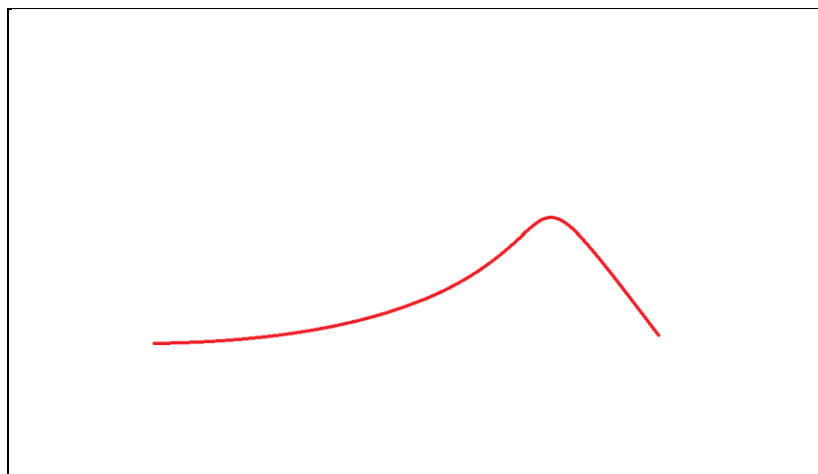
4.2 Ideální obrázek - prahování, aproximace polynomem a konstrukce oskulačních kružnic

V této kapitole bude popsán algoritmus sloužící jako základ pro pozdější návrh řízení tak, jak by fungoval pro ideální obrázky bez jakéhokoliv šumu.

Prahování je obecně transformace vstupního obrazu g na výstupní binární obraz f s prahem T . Výsledkem je binární obraz, kde obrazové elementy náležející objektu (jas/parametrizovaná barva větší než práh) mají hodnotu 1 a pixely náležející k pozadí hodnotu 0:

$$f(i,j) = \begin{cases} 1 & \text{pro } g(i,j) \geq T \\ 0 & \text{pro } g(i,j) < T \end{cases} \quad (63)$$

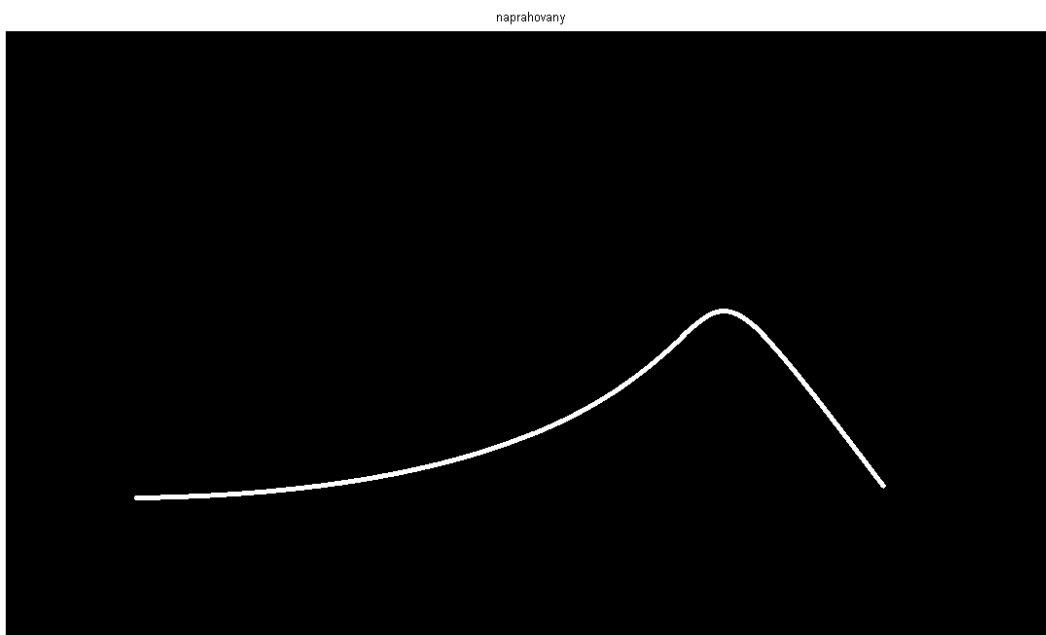
V našem případě byla pro účely testování zkonstruována libovolná hladká křivka červené barvy, která je umístěna na dokonalém bílém pozadí bez jakéhokoliv šumu (Obr. (37)).



Obr. (37): Původní obrázek (ideální).

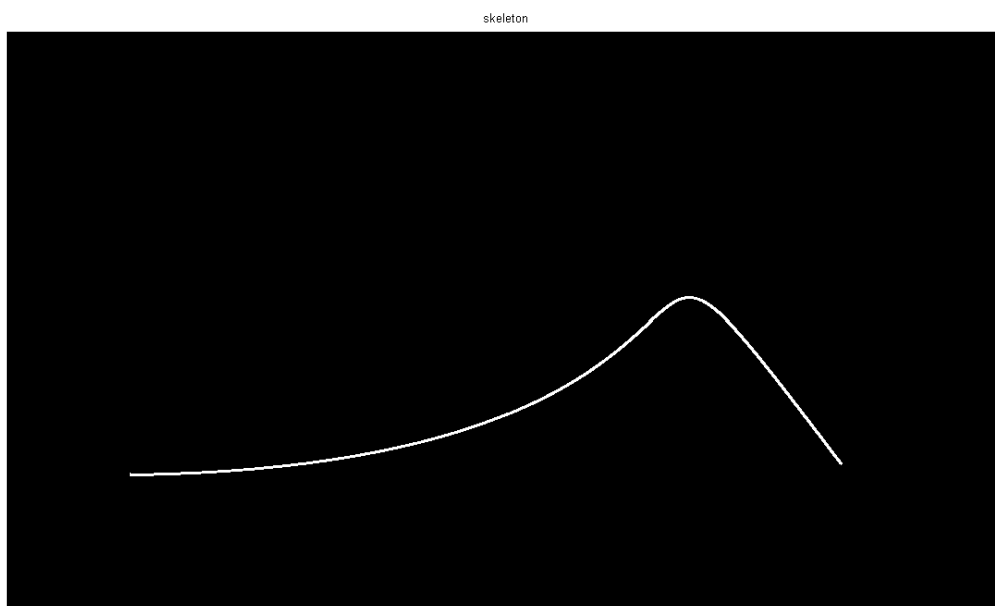
Tento obrázek se nyní naprahuje podle vztahu (63). Vybraná barva je popsána třemi středními hodnotami a příslušnými rozptyly. To znamená, že jsou brány v úvahu všechny tři složky barevného modelu HSV, tedy *Hue*, *Saturation* a *Value*. Pokud je na obrázku nějaký pixel s parametry, které splňují podmínky a náleží do uvedené oblasti kolem střední hodnoty pro H, S i V referenční barvy, algoritmus tomuto pixelu přiřadí hodnotu 1. Jako práh je tedy použit rozptyl od střední hodnoty zvážený vhodným koeficientem pro jednotlivé složky určité barvy reprezentované pomocí modelu HSV.

Výsledkem je Obr. (38):



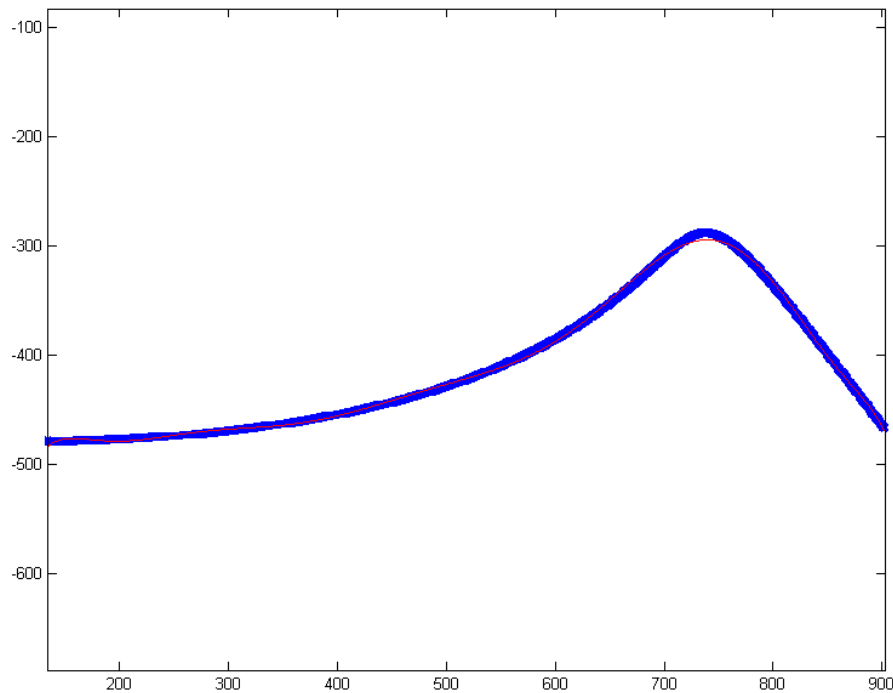
Obr. (38): Naprahovaný obrázek.

Nyní se ještě naprahovaný obrázek upraví metodou *skeleton*, která vybere pouze ty body, které tvoří tzv. kostru. Tím naprahovanou křivku ztenčí a usnadní další postup (Obr. (39)).



Obr. (39): Obrázek upravený metodou skeleton.

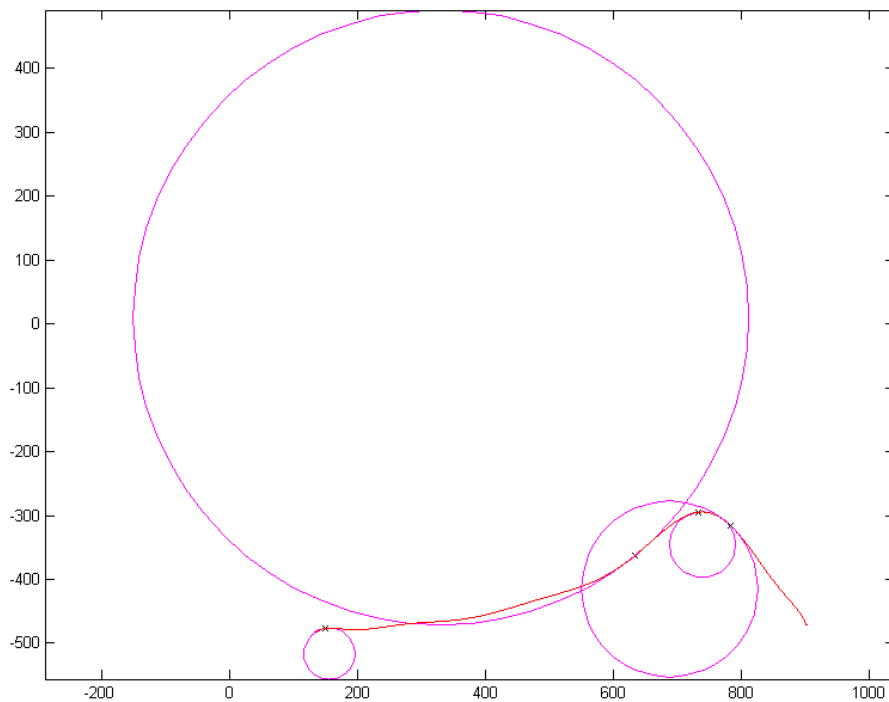
Teď se metodou nejmenších čtverců proloží body polynomem 10. řádu (Obr. (40)):



Obr. (40): Aproximace křivky polynomem 10. řádu.

V této simulaci jsou používány všechny naprahované body pro výpočet koeficientů hledaného polynomu. Také řád polynomu je vysoký, čili celá úloha je špatně podmíněná. Oba tyto problémy způsobí, že takto nastavené řešení není vhodné pro cílovou platformu, neboť výpočetní rychlost použitého procesoru bude o poznání menší než na PC. Řešením může být vybrání pouze několika naprahovaných bodů křivky v dostatečné vzdálenosti od sebe a zvolením nižšího stupně polynomu (viz Kapitola 3 - Experiment).

Pokud je znám polynomiální předpis aproximované křivky, může se už pomocí (51) a (55) vypočítat poloměr a střed oskulačních kružnic potřebných pro řízení robotu. Některé oskulační kružnice jsou znázorněny níže (Obr. (41)):



Obr. (41): Znáznornění oskulačních kružnic ve vybraných bodech aproximované křivky.

Poloměry jednotlivých kružnic (v pixelech) jsou po řadě:

$$l_1 = -39,99$$

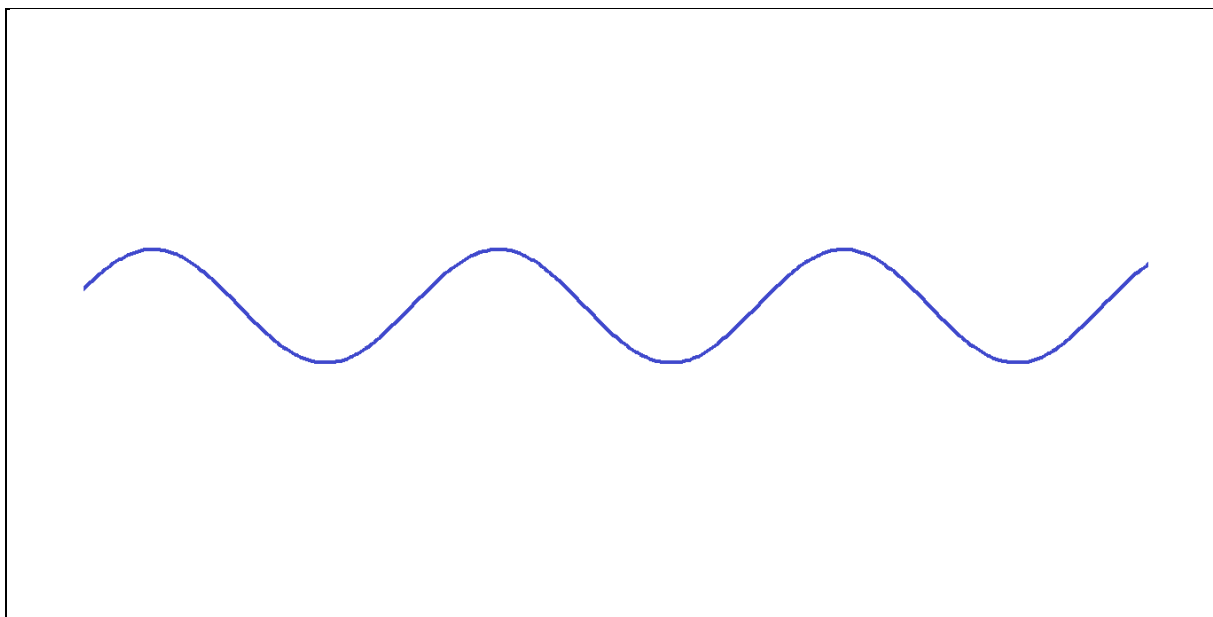
$$l_2 = 481,42$$

$$l_3 = -51,13$$

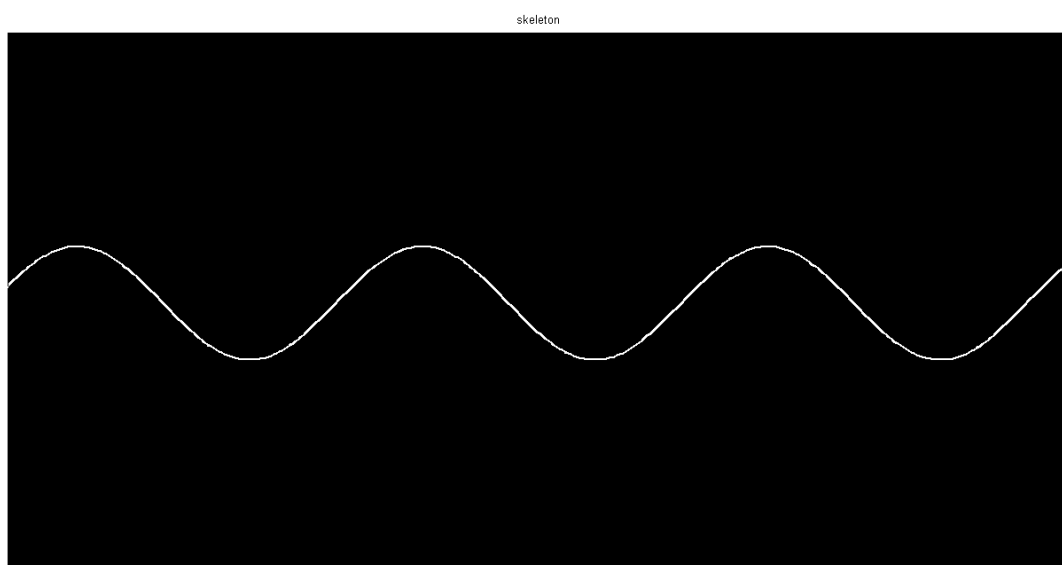
$$l_4 = -137,77$$

kde znaménko udává polohu oskulační kružnice vzhledem ke křivce, tedy buď „nad“ křivkou (znaménko +), nebo „pod“ křivkou (znaménko -).

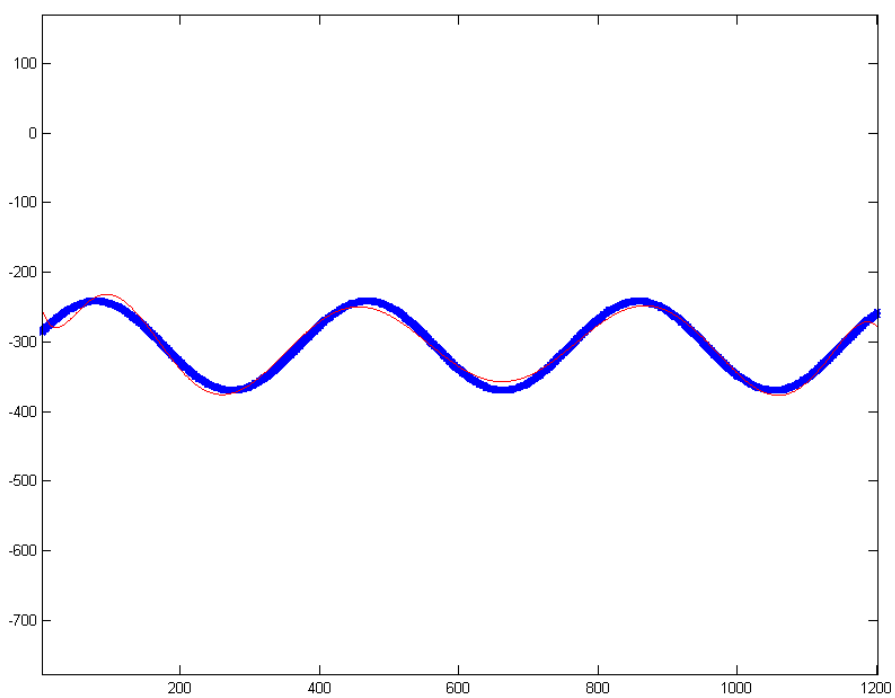
V této práci je použit příkaz *imcrop*, pomocí kterého uživatel vybere barvu (oblast), která se má na obrázku rozpoznat. Důsledkem je to, že lze celý postup zopakovat pro jakoukoliv barvu, což je výhodné pro finální aplikaci, jelikož barva křivky nemusí být pokaždé stejná. Jako příklad je zde uvedena sinusoida v modré barvě a celý postup je zopakován:



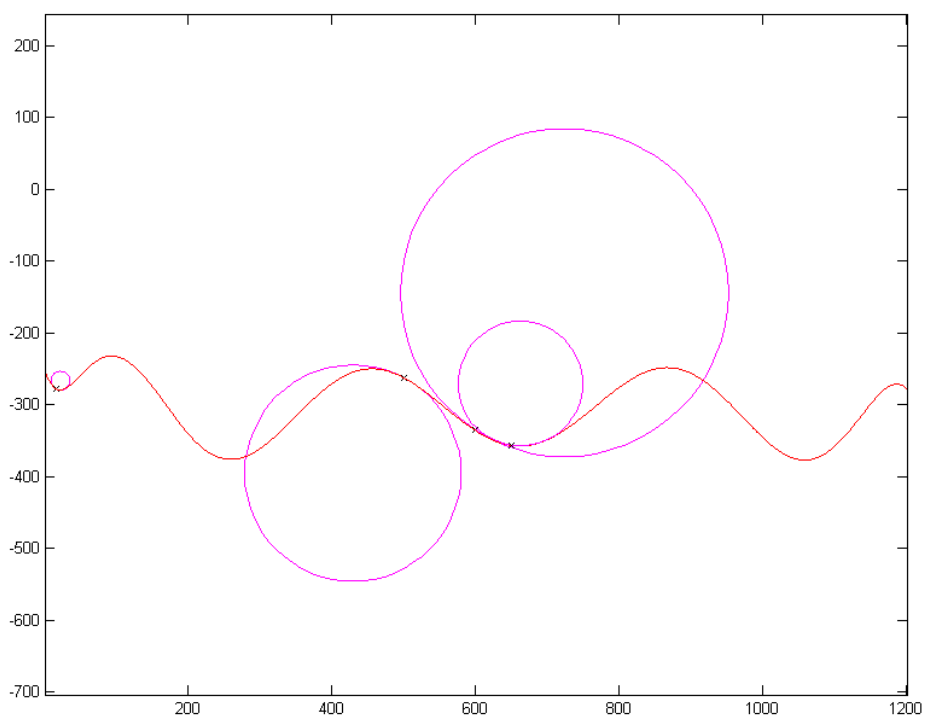
Obr. (42): Původní obrázek (ideální).



Obr. (43): Naprahovaný obraz upravený funkcí skeleton.



Obr. (44): Aproximace polynomem 10. řádu.



Obr. (45): Oskulační kružnice ve vybraných bodech.

Poloměry jednotlivých kružnic (v pixelech) jsou po řadě:

$$l_1 = 13,00$$

$$l_2 = -150,63$$

$$l_3 = 228,31$$

$$l_4 = 86,64$$

Kapitola 3

5 Experiment

Teoretické předpoklady musejí být nějakým způsobem ověřeny. V této práci není uvažován reálný fyzický model robotu, na kterém by se mohl návrh řízení otestovat. Proto bylo nutné nějaké virtuální testovací prostředí vytvořit.

Základní idea tkvěla v tom, že by se robot měl naprosto volně pohybovat v omezeném prostoru, avšak bez nutnosti znát svoji přesnou polohu a natočení vůči světovému souřadnému systému. Řízení musí být závislé pouze na vstupu z kamery, tedy reagovat pouze na to, co robot imaginárně „vidí“. Simulace pohybu v takto vytvořené virtuální 2D realitě by využívala kinematický model popsany v 2.1, tedy by použila výstupy (polohu a natočení robotu) pro znázornění a kontrolu pohybu v takto omezeném prostoru.

Souřadnice polohy a natočení jsou tedy používány pouze za účelem simulace a nevstupují do algoritmu řízení!

Pro návrh řízení bylo také nutné zvolit měřítko daného obrázku a další parametry. V této práci jsou používány následující přepočty:

$$\begin{aligned} 1px &= 2 \cdot 10^{-4}m = 0,2mm \\ 1m &= 5 \cdot 10^3px \end{aligned} \tag{64}$$

Maximální rychlost těžiště robotu zvolena jako:

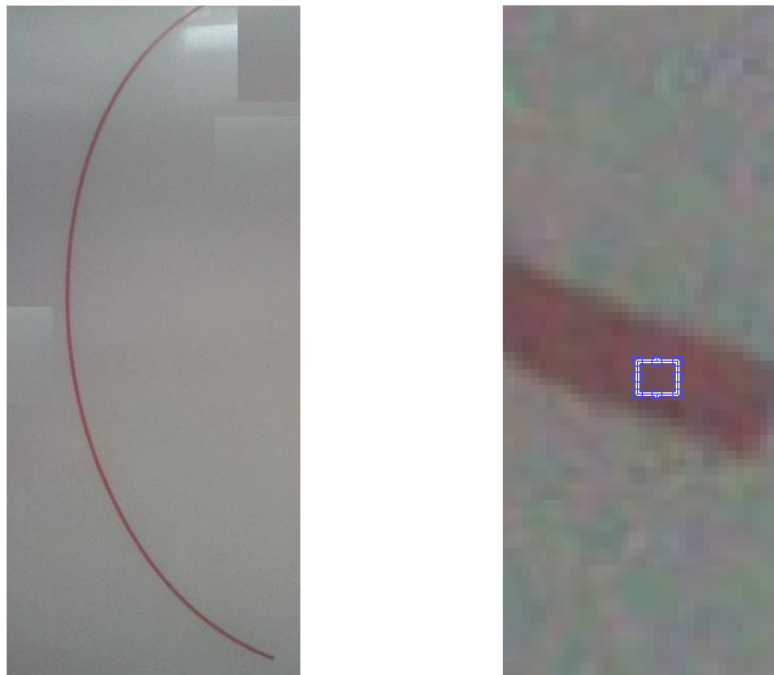
$$v_{max} = 10^{-2}ms^{-1} = 1 \text{ cm/s} \tag{65}$$

Definujeme také krok systému (vzorkovací periodu) jako fixní s hodnotou:

$$T_S = 0,05s = 50ms \tag{66}$$

5.1 Tvorba testovacího prostředí

Za účelem otestování algoritmu řízení bylo vytvořeno několik průhledných samolepek se zvýrazněnou křivkou červené barvy, které se dají nalepit na různé materiály. Křivky mají různé tvary s různými poloměry křivosti zatáček tak, aby se daly co nejlépe otestovat vlastnosti navrženého řízení. Po aplikaci samolepky na nějaký materiál se tato čára vyfotografuje a tím vznikne jakási mapa - ohraničený prostor, ve kterém se může robot pohybovat. Výhodou tohoto přístupu je bezesporu to, že na křivku působí veškeré rušivé faktory (odlesky, změna odstínu čáry v důsledku změny přisvětlení apod.), které komplikují rozpoznávání, ale které by se reálně vyskytovaly i při použití robotu v praxi. Pro účely práce se předpokládá, že je robot položen přímo na čáru ve směru požadovaného pohybu. Po uvedení do provozu se vybere malá část snímaného obrazu pod těžištěm robotu, která slouží pro identifikaci barvy křivky. Reálně tedy není použita funkce *imcrop* popsaná v podkapitole 4.2, protože nemusí být kvůli výpočetním nárokům implementována v cílové platformě a také není možné předpokládat, že uživatel bude ručně zadávat odstín čáry na nějakém externím ovládacím zařízení. Znázornění celkové mapy a výběru oblasti pro identifikaci barvy je zřejmé na Obr. (46).

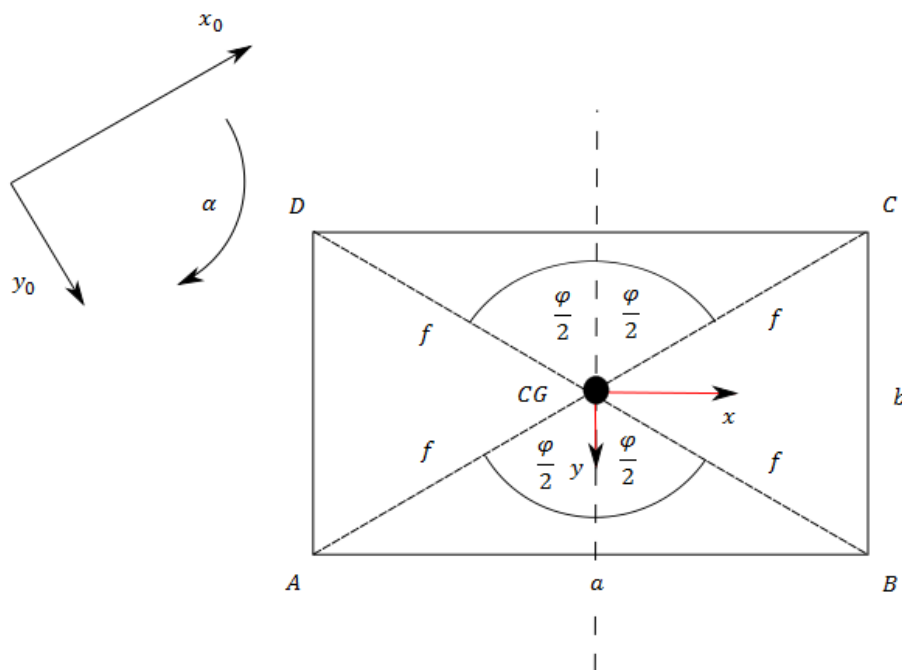


Obr. (46): Původní reálný obrázek (vlevo) a vybraná část (vpravo).

5.2 Pohled z robotu

Dalším krokem k úspěšně vykonanému experimentu byla simulace obrázku, který by snímala kamera umístěná na robotu. Zatím se předpokládá, že kamera je umístěna kolmo nad podložkou a těžiště robotu leží nad touto kamerou, čili střed pořízeného obrázku je totožný s polohou těžiště robotu.

Pro simulaci pohledu z robotu je v této práci použita funkce *PohledKamera.m*, která využívá následující geometrické úvahy (Obr. (47)):



Obr. (47): Posunutí a rotace obecného obdélníku v rovině.

Ze známé polohy těžiště $CG = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ a natočení obdélníku α vzhledem ke světovému souřadnému systému je možné se znalostí některých základních geometrických vztahů vypočítat souřadnice jednotlivých vrcholů následujícím postupem:

$$f = \sqrt{\left(\frac{a}{2}\right)^2 + \left(\frac{b}{2}\right)^2} \quad (67)$$

$$\frac{\varphi}{2} = \operatorname{arctg} \frac{\frac{a}{2}}{\frac{b}{2}} = \operatorname{arctg} \frac{a}{b}$$

$$A = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -f \sin\left(\frac{\varphi}{2} + \alpha\right) + x_1 \\ f \cos\left(\frac{\varphi}{2} + \alpha\right) + x_2 \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} f \sin\left(\frac{\varphi}{2} - \alpha\right) + x_1 \\ f \cos\left(\frac{\varphi}{2} - \alpha\right) + x_2 \end{bmatrix}$$

(68)

$$C = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} f \sin\left(\frac{\varphi}{2} + \alpha\right) + x_1 \\ -f \cos\left(\frac{\varphi}{2} + \alpha\right) + x_2 \end{bmatrix}$$

$$D = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} -f \sin\left(\frac{\varphi}{2} - \alpha\right) + x_1 \\ -f \cos\left(\frac{\varphi}{2} - \alpha\right) + x_2 \end{bmatrix}$$

Uvedené souřadnice bodů definují oblast, která se vyřízne z celkové mapy. Takto získaná oblast se poté transformuje na obdélník o zadané šířce a výšce pomocí funkce *imtransform()*, která využívá principy popsané v podkapitole 3.4. V této práci je použit výřez o rozměrech 200x100 pixelů, ale přiložený program umožňuje libovolné nastavení šířky a výšky vybrané oblasti.

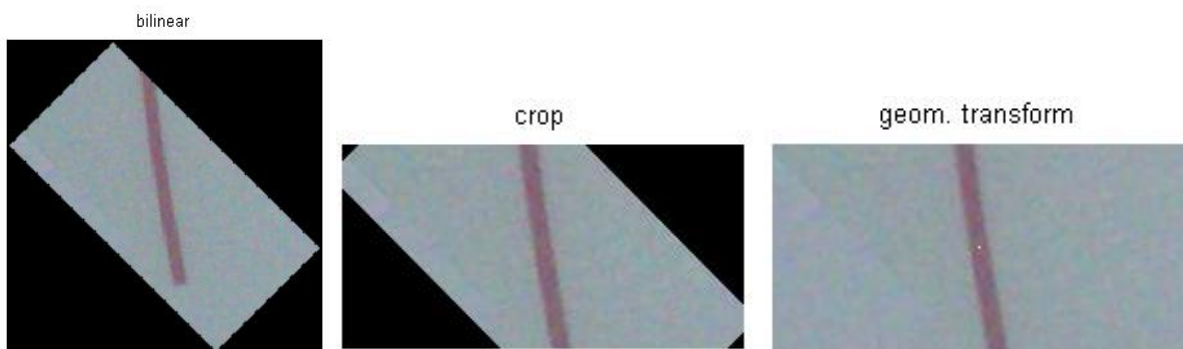
pohled z kamery



Obr. (48): Pohled z imaginární kamery.

Černá oblast nalevo na Obr. (48) znázorňuje prostor za hranicí mapy, nejedná se tedy o ztrátu dat způsobenou chybnou transformací.

Data by naopak byla ztracena použitím jednoduššího přístupu a předdefinovaných metod pro otáčení obrázků v prostředí Matlab (např. *imrotate*). Na následujících obrázcích (Obr. (49)) je zřejmá výhoda použití metod projektivní geometrie, protože takto získaný obraz je reálným zobrazením toho, co kamera z robotu doopravdy vidí.

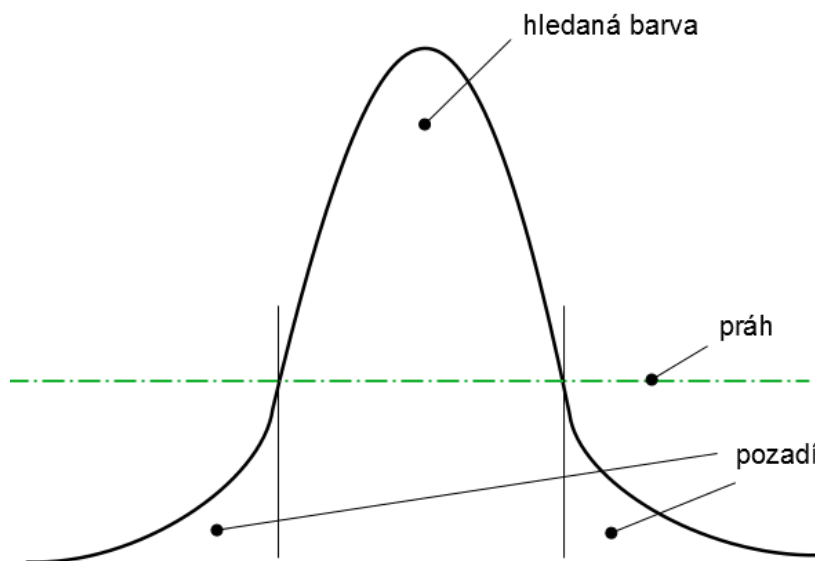


*Obr. (49): Otočená část mapy získaná metodou *imrotate(bilinear)* - vlevo, metodou *imrotate(crop)* - uprostřed, metodou geometrické transformace - vpravo.*

5.3 Prahování reálného obrázku

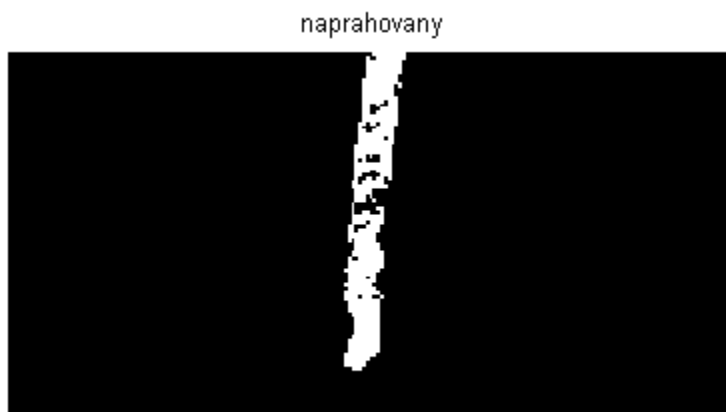
V oblasti prahování bylo nutné učinit některé změny oproti teoretickým předpokladům popsaných v podkapitole 4.2 už jen kvůli přítomnosti reálných obrázků. Pro tyto účely slouží v příloženém programu funkce *detect_realPics.m*. Barva čáry se tentokrát reprezentuje pomocí Gaussovy křivky následujícím způsobem:

1. vybraná část obrázku, na které je zachycen pouze vnitřek čáry ve formátu RGB, je převedena do formátu HSV
2. složky tenzoru vzniklého výřezu jsou po sloupcích převedeny do tzv. vektoru příznaků pomocí funkce *reshape*
3. vzniklým vektorem příznaků se proloží Gaussova křivka (tedy nad všemi složkami modelu HSV) a určí se práh, v této práci je použita hodnota: $prah = 10^{-4}$
4. pro každý bod celého obrázku (opět ve formátu HSV), který je snímán kamerou robotu, se vypočítá hodnota věrohodnosti
5. pokud tato hodnota věrohodnosti spadá do pásma hledané barvy, je pixelu přiřazena hodnota 1, v opačném případě je tento pixel považován za pozadí a je mu přiřazena hodnota 0



Obr. (50): Prahování pomocí Gaussovy křivky.

Výsledkem prahování Obr. (48) je následující Obr. (51):



Obr. (51): Naprahovaný reálný obrázek.

V této práci je ještě navíc použito adaptivní prahování, které spočívá v tom, že se každých 50 kroků algoritmu sejme čtverec 5x5px (1x1mm) kolem bodu, ve kterém se počítá oskulační kružnice. To do značné míry eliminuje chybu při prahování způsobenou změnou odstínu čáry. Na začátku natrénovaný odstín barvy se totiž může během chodu cyklu změnit natolik, že systém postupem času nemusí být schopen čáru vůbec rozpoznat. Takto navržené adaptivní prahování naopak dostatečnou měrou zaručuje, že nový vzor (vybraný čtverec), podle kterého se bude prahovat, bude skutečně ležet na sledované čáře, a proto bude dostatečně spolehlivě reflektovat případné změny odstínů.

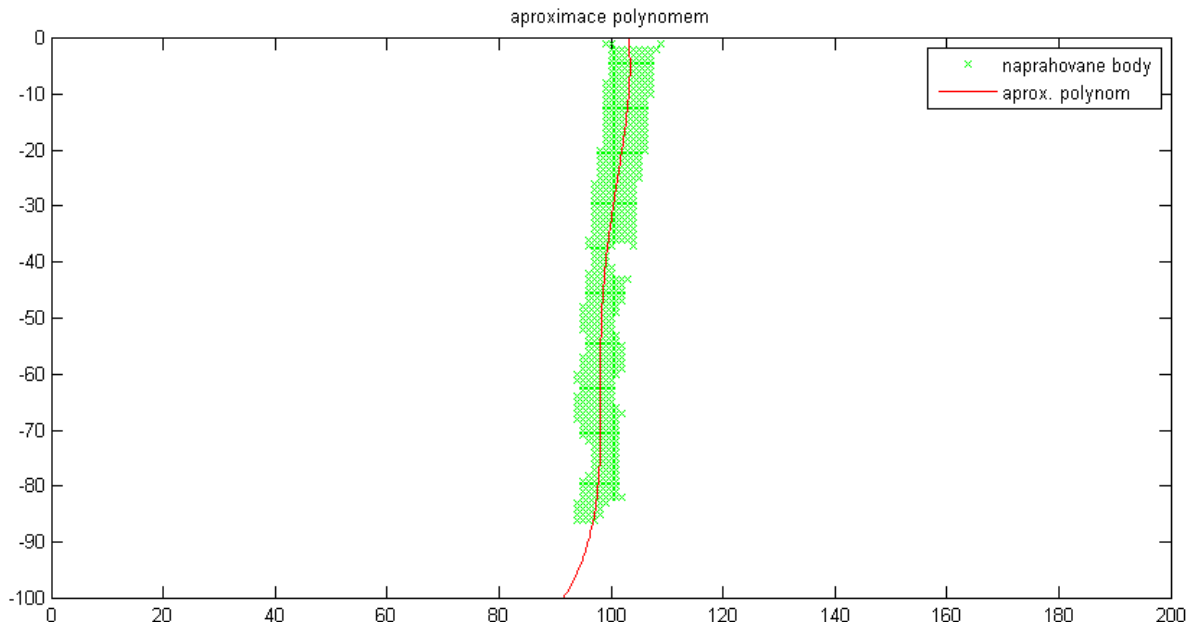
5.4 Proložení reálného obrázku polynomem a konstrukce oskulační kružnice

I co se týče aproximace naprahaného útvaru polynomem, musely být provedeny některé úpravy v porovnání s ideálním případem.

V první řadě se polynom nyní počítá jako funkce proměnné y , aby se předešlo „pádu“ algoritmu v případě, že čára bude dokonale kolmá ke kameře. V tomto případě by totiž proložený polynom proměnné x neexistoval, neboť by to nebyla funkce. Naopak se nepředpokládá, že čára bude z pohledu kamery vodorovná, čili řešení v podobě proložení polynomem, který je funkcí y , se zdá jako lepší a robustnější.

Druhou úpravou je snížení řádu aproximovaného polynomu z deseti na čtyři kvůli snížení výpočetních nároků.

Výpočetní nároky se mohou snížit i omezením počtu bodů, ze kterých se počítá metodou nejmenších čtverců aproximovaný polynom. Na následujících obrázcích (Obr. (52, 53)) je však vidět, že velká selekce bodů má velký vliv na přesnost následných výpočtů.



Obr. (52): Aproximace polynomem 4. řádu (použity všechny naprahané body).

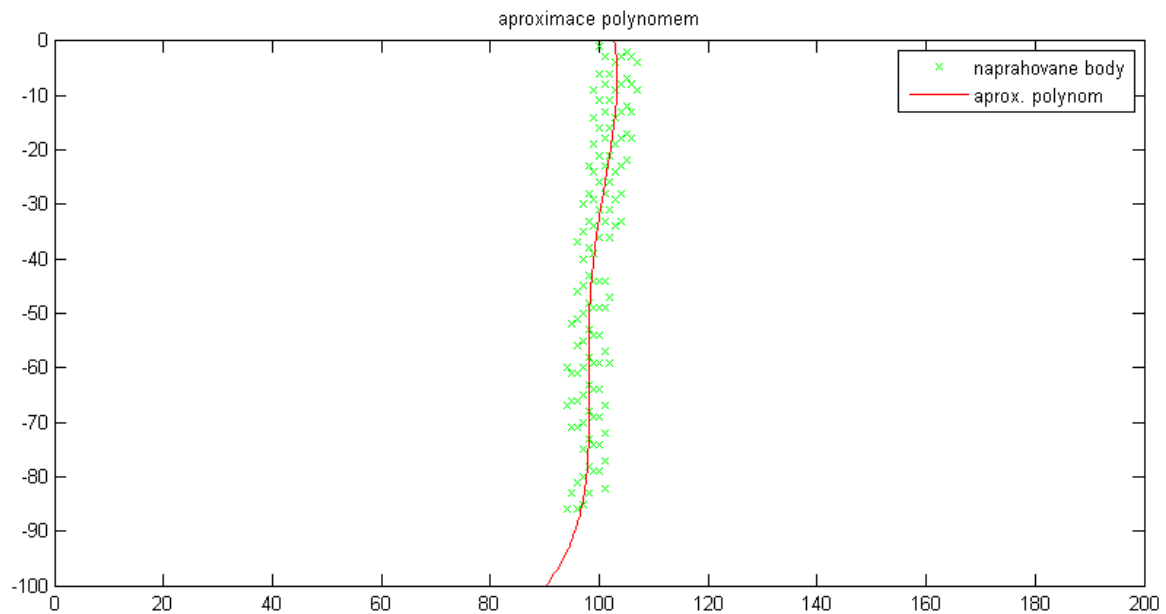
Polynom je ve tvaru:

$$p(y) = -1,29 \cdot 10^{-6}y^4 - 2,30 \cdot 10^{-4}y^3 - 0,01y^2 - 0,10y + 103,14 \quad (69)$$

Poloměr oskulační kružnice v prostředním bodu (Obr. (52) - není znázorněna):

$$l = -175,1071 \text{ [px]} \quad (70)$$

Při použití pětiny bodů se poloměr oskulační kružnice (72) již značně liší:



Obr. (53): Aproximace polynomem 4. řádu (použita pětina naprahovaných bodů).

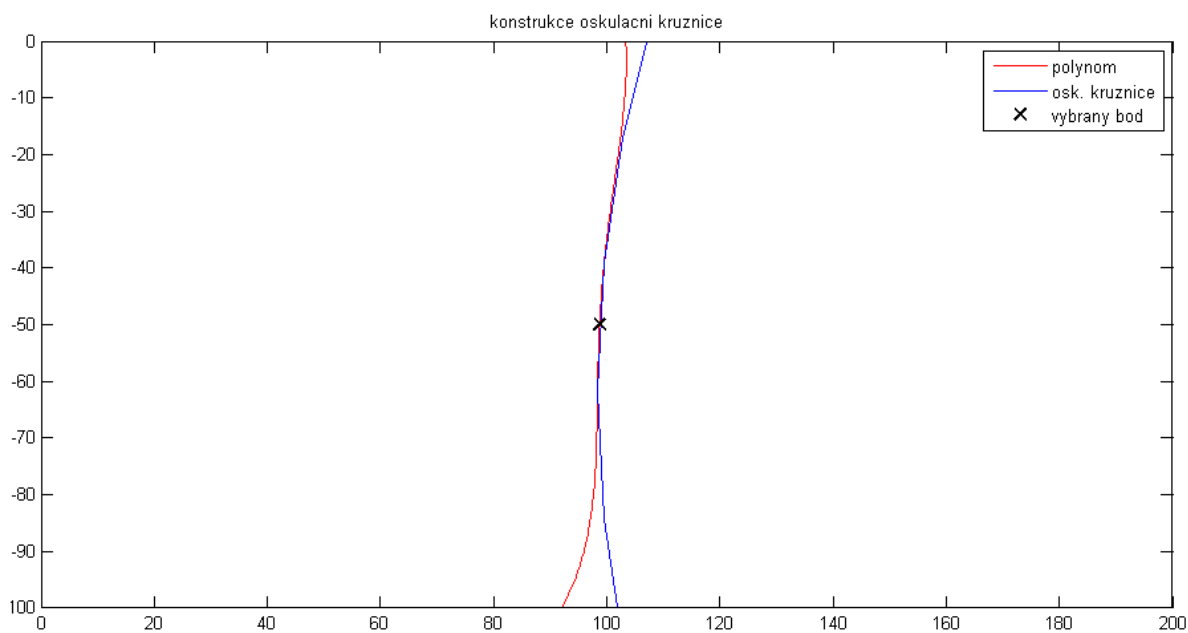
Polynom je ve tvaru:

$$p(y) = -1,28 \cdot 10^{-6}y^4 - 2,82 \cdot 10^{-4}y^3 - 0,02y^2 - 0,17y + 107,74 \quad (71)$$

Poloměr oskulační kružnice v prostředním bodu (Obr. (53) - není znázorněna):

$$l = -152,5158 \text{ [px]} \quad (72)$$

Konstrukce osculační kružnice ve vybraném bodu (středu křivky) - Obr. (54):

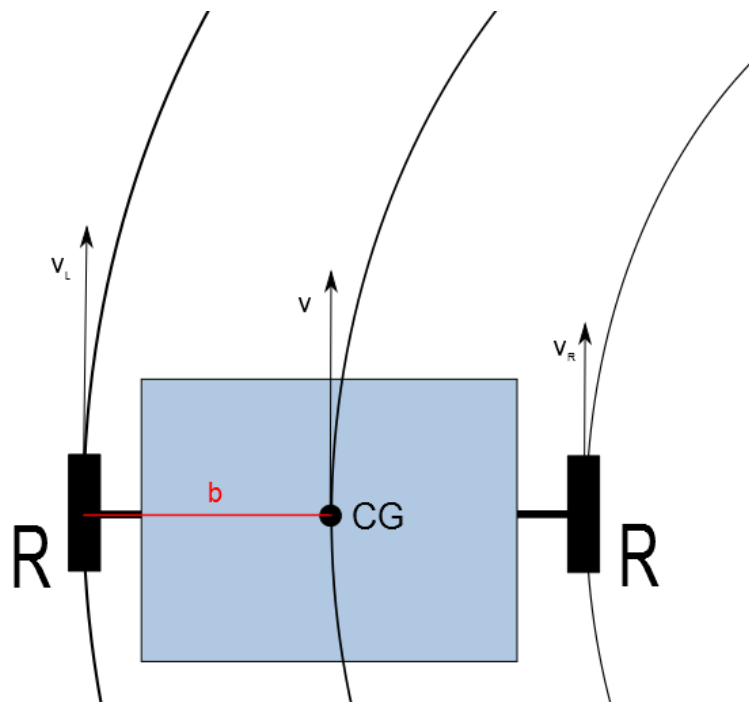


Obr. (54): Oskulační kružnice ve vybraném bodu.

Je nutné dodat, že při takto počítaném aproximačním polynomu má osculační kružnice se středem napravo od křivky poloměr se záporným znaménkem; naopak osculační kružnice se středem nalevo od křivky má kladný poloměr. Toto je důležité pro návrh řízení (viz následující podkapitola 5.5).

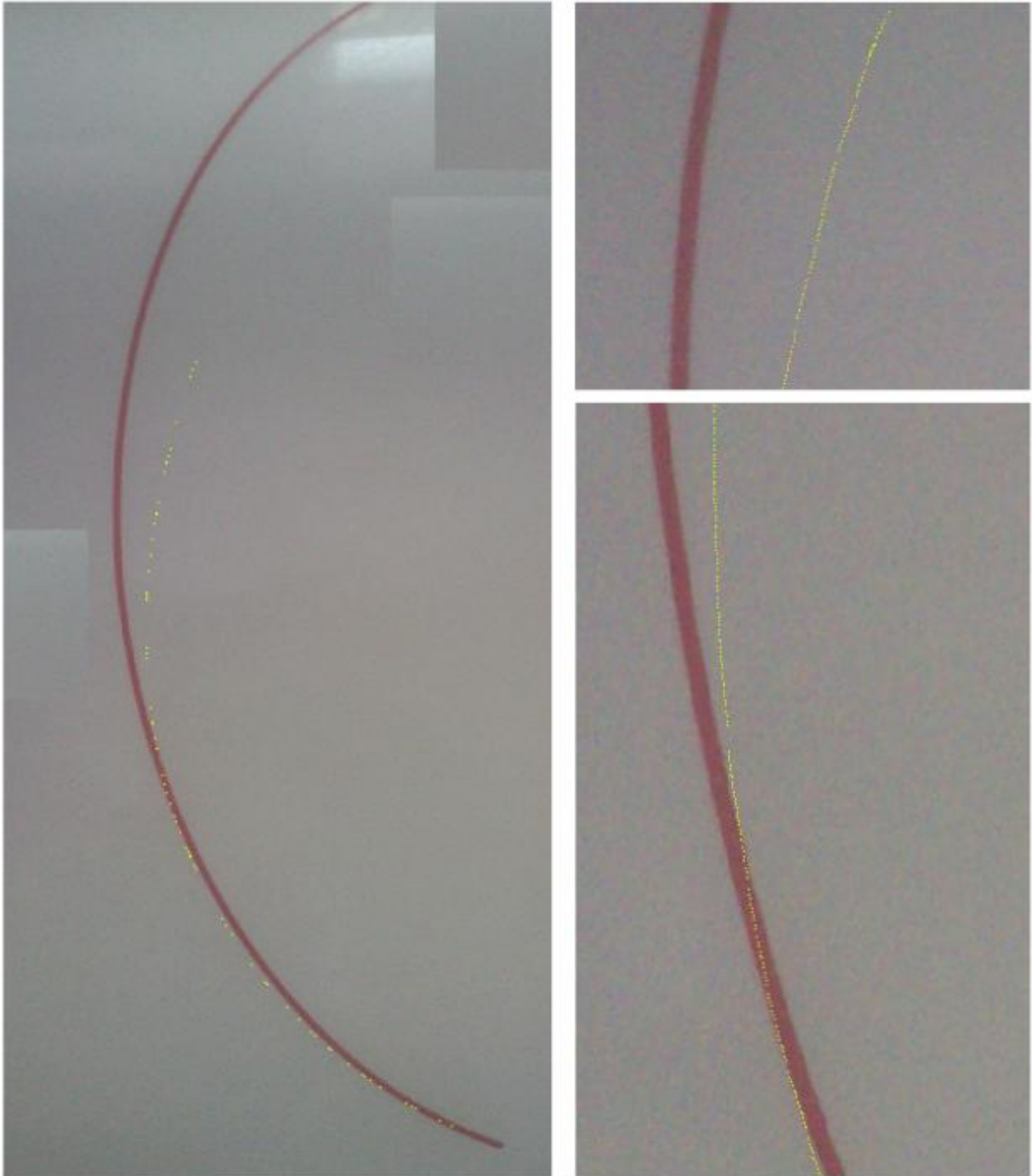
5.5 Návrh řízení

Hlavní ideou pro návrh řízení bylo využít oskulační kružnice. Získaný reálný obrázek z kamery se naprahuje, naprahovaná část se proloží polynomem 4. řádu a v daném bodu (v tomto případě v „prostředním bodu“ polynomu) se vypočítá poloměr příslušné oskulační kružnice. Podle znaménka u získaného poloměru je možné s jistotou určit, kam má robot zatáčet. Potom se znalostí konstrukčních parametrů robotu (vzdálenost těžiště od kol a poloměru kol) není problém podle 3.1 dopočítat lineární rychlosti kol, ze kterých je možno získat potřebné úhlové rychlosti kol. Uvedený výpočet realizuje funkce *Rizeni.m*. V programu je zabudovaná pojistka v případě, že poloměr oskulační kružnice je příliš velký ($|l| > 10^6$ px). V tomto případě je kružnice považována za přímku a robot se pohybuje přímočaře. Celá situace je znázorněna na následujícím obrázku (Obr. (55)):



Obr. (55): Idea návrhu řízení pomocí oskulačních kružnic.

Předpokládalo se, že navržené řízení pomocí aproximace naprahané čáry polynomem, následného výpočtu oskulační kružnice ve středu polynomu a přepočítání lineárních rychlostí kol na rychlosti úhlové, tedy bez regulátoru, nebude dostatečně účinné. Výsledky simulací jsou znázorněny níže (Obr. (56)):

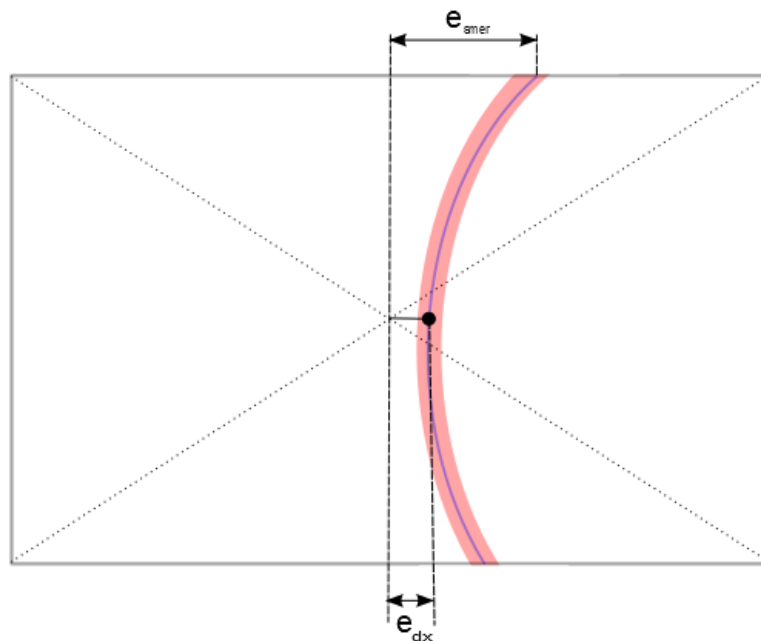


*Obr. (56): Výsledek simulace pro řízení bez regulátoru.
(Poznámka: Žluté body označují těžiště robotu v jednotlivých krocích simulace.)*

Robot sice nakonec z čáry sjel, ale poměrně dlouho se na ní udržel. To vedlo k úvaze regenerovat celé nové řízení pomocí regulátoru, ale jen opravit to původní založené na oskulačních kružnicích. Byla definována regulační odchylka e_{dx} jako x-ová vzdálenost mezi bodem, ve kterém se konstruuje oskulační kružnice („prostřední bod“ aproximovaného polynomu), a geometrickým středem obrázku. Tato regulační odchylka se ukázala být nedostačující, a proto byla definována ještě druhá regulační odchylka e_{smer} jako x-ová vzdálenost mezi posledním viditelným bodem čáry (aproximovaného polynomu) a vertikální osou procházející středem obrázku.

$$\begin{aligned} e_{dx} &= x_{stred} - x_{OSK} \\ e_{smer} &= x_{stred} - x_{END} \end{aligned} \quad (73)$$

Grafické znázornění konstrukce regulačních odchylek je možné lépe pochopit z následujícího obrázku (Obr. (57)):



Obr. (57): Konstrukce regulačních odchylek.

Pro takto definované odchylky (73) byl vytvořen následující algoritmus:

- jestliže $e_{dx} > 0$ nebo $e_{smer} > 0$, pak oprav rychlost pravého kola, levé ponechej
 \Rightarrow čára je nalevo od středu obrázku, pravé kolo musí zrychlit
- jestliže $e_{dx} < 0$ nebo $e_{smer} < 0$, pak oprav rychlost levého kola, pravé ponechej
 \Rightarrow čára je napravo od středu obrázku, levé kolo musí zrychlit

Tento postup je příkladem využití vizuální zpětné vazby popsané v podkapitole 3.5.

Pro opravu úhlových rychlostí jednotlivých kol byl použit PD regulátor popsaný v podkapitole 3.3. Přepínání mezi regulátory zajišťuje funkce *porovnaní.m*, která realizuje výše uvedený algoritmus.

Parametry všech regulátorů byly voleny experimentálně (metodou pokus - omyl). Jedná se totiž o nelineární systém a veškeré pokusy o zapojení sofistikovanějších metod selhaly (viz Závěr).

Konkrétní hodnoty parametrů PD regulátorů byly zvoleny pro obě regulační odchylky e_{dx} a e_{smer} stejně. Jediným rozdílem je zařazení konstant K_{dx} a K_{smer} , které násobí chyby vstupující do regulátorů a tím udávají poměr, jakým se mají jednotlivé regulátory podílet na řízení a jak velký akční zásah mají vygenerovat. Tyto konstanty vlastně „váží důležitost“, která je přiřazována jednotlivým regulátorům.

Hodnoty parametrů PD regulátorů (74):

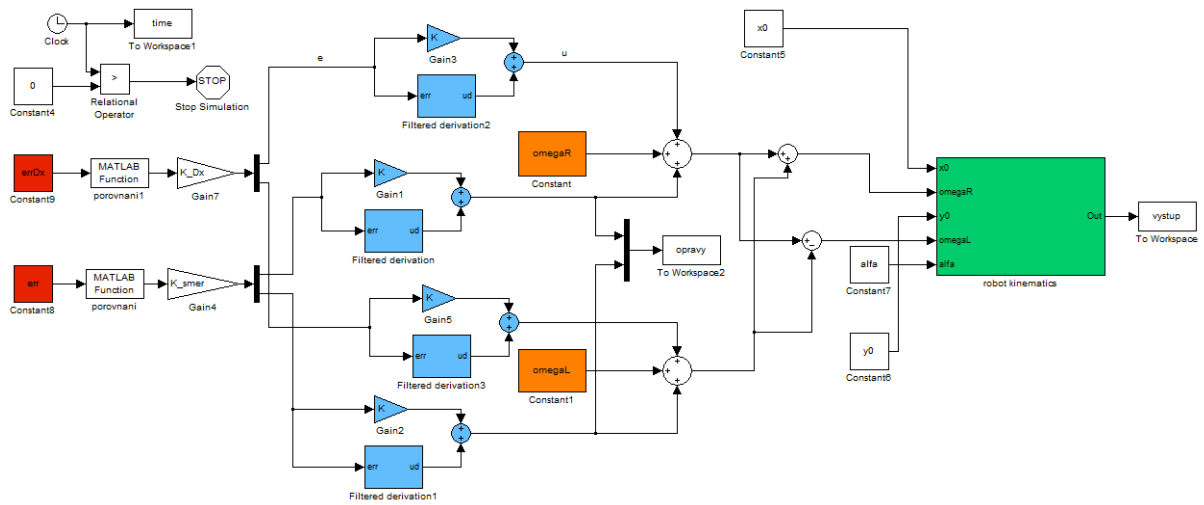
$$\begin{aligned} K &= 5 \cdot 10^{-4} \\ T_D &= 5 \cdot 10^{-2} \\ \tau &= \frac{T_D}{5,2} = 9,6 \cdot 10^{-3} \\ K_D &= KT_D = 2,5 \cdot 10^{-5} \\ (K_{dx} &= 1, K_{smer} = 1,2) \end{aligned} \tag{74}$$

V algoritmu řízení je ještě zakomponována úprava aktuální rychlosti vozíku v závislosti na velikosti poloměru oskulační kružnice. Tato závislost je lineární a platí:

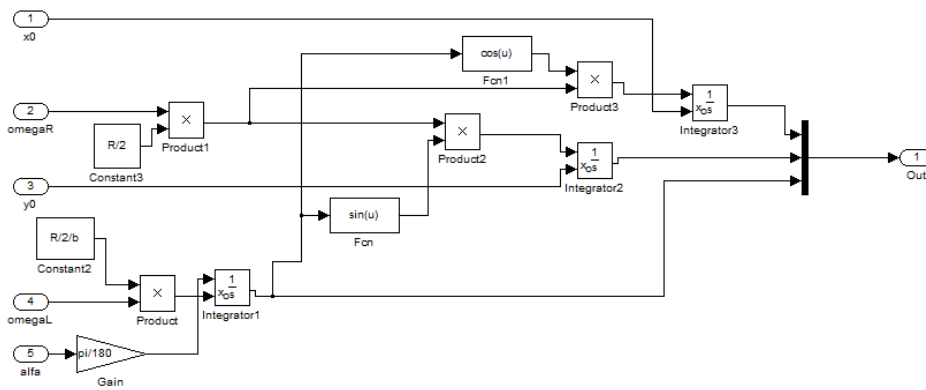
$$v = \begin{cases} v_{max}, & |l| > 200 [px] \\ k \cdot |l|, & jinak \end{cases} \tag{75}$$
$$k = 0,5 \cdot 10^{-4}$$

kde konstanta k je vztažena k poloměru oskulační kružnice v pixelech.

Použitá schémata v programu Matlab/Simulink vypadají následovně (Obr. (58,59)):

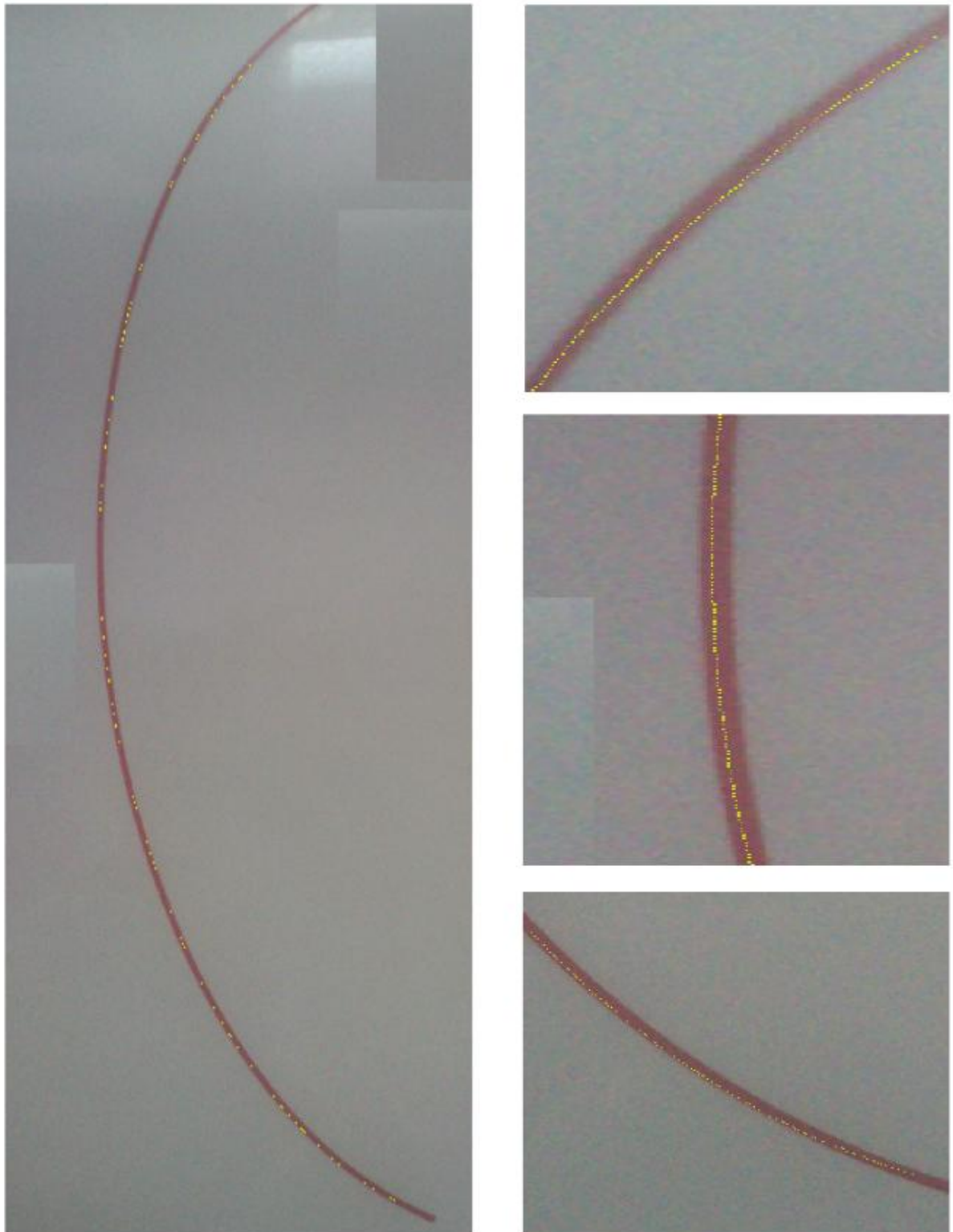


Obr. (58): Použité řídicí schéma.



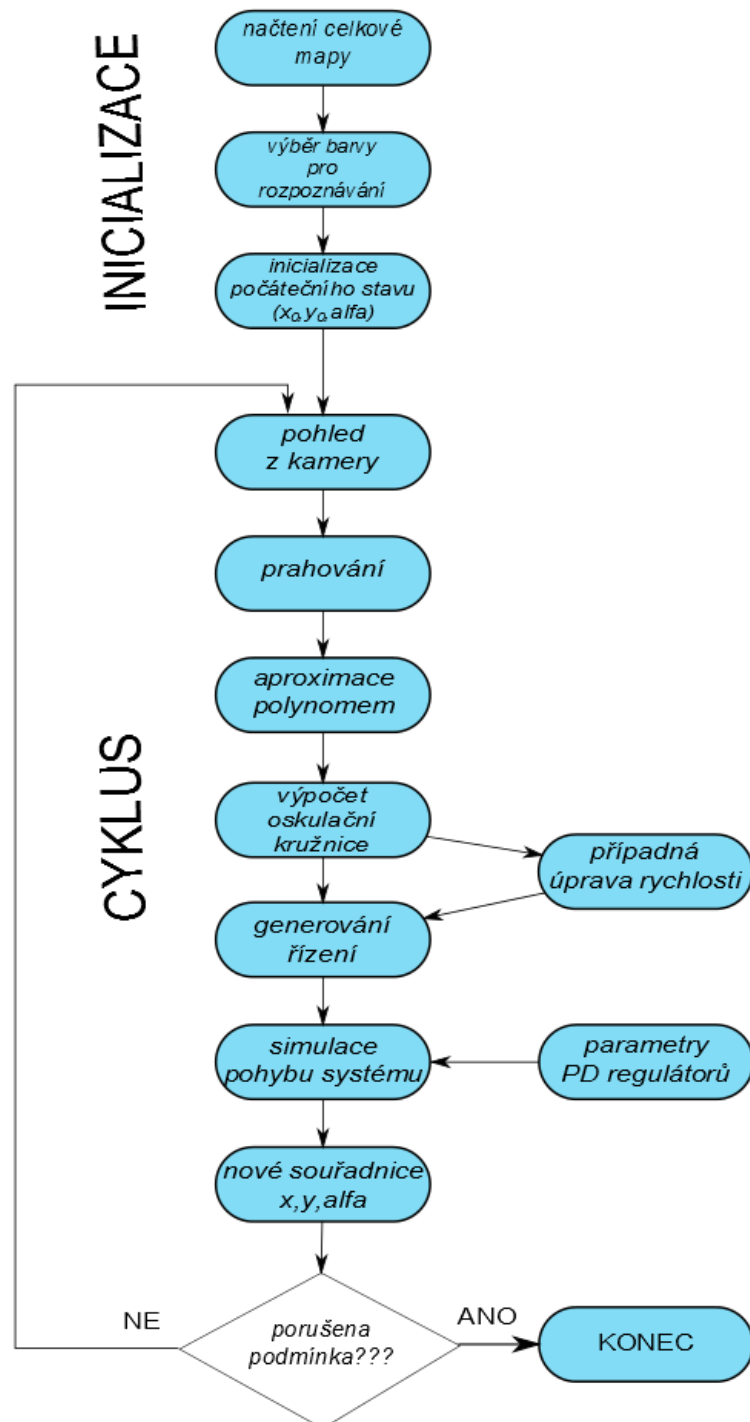
Obr. (59): Subsystém robot kinematics.

Výsledek simulace s PD regulátory (Obr. (60)):



Obr. (60): Simulace s PD regulátory.

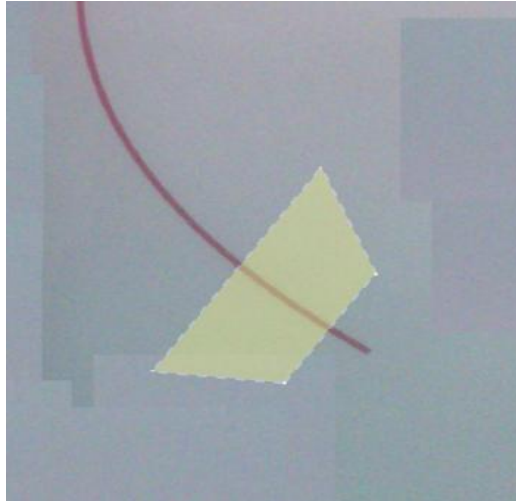
Zde je pro přehlednost uveden celý algoritmus rozpoznávání a řízení pomocí diagramu (Obr. (61)):



Obr. (61): Celý algoritmus rozpoznávání a řízení.

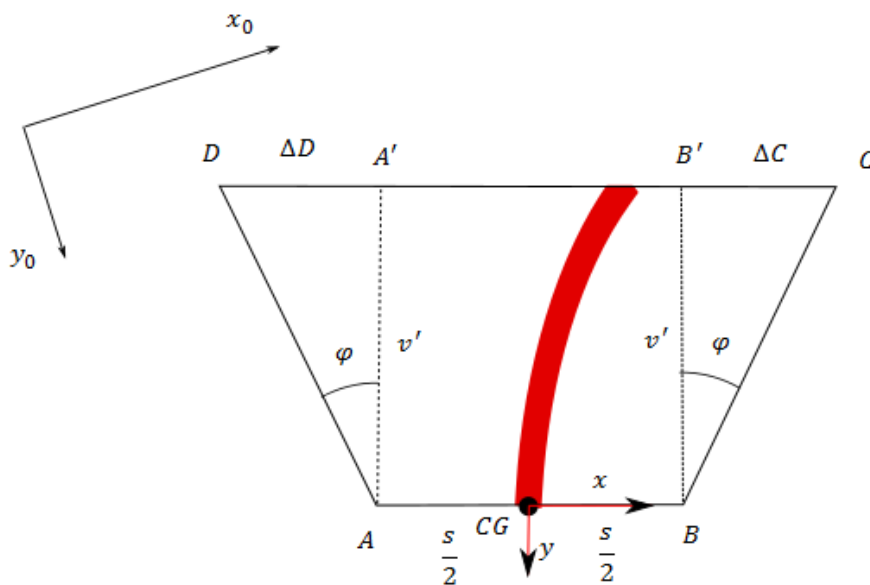
5.6 Rozšíření na perspektivní pohled z robotu

Nyní se předpokládá, že je kamera umístěná tak, že směřuje dopředu. Střed dolního okraje pořízeného snímku je považován za souřadnice těžiště robotu. Z celkové mapy je tedy vybrána oblast ve tvaru rovnoramenného lichoběžníku, která je znázorněna na následujícím obrázku (Obr. (62)):



Obr. (62): Vybraná oblast pro perspektivní pohled.

Níže jsou popsány některé důležité body pro transformaci (Obr. (63)):



Obr. (63): Obecný geometrický popis použitý pro transformaci.

kde:

- $\frac{s}{2}$ - je polovina šířky dolního okraje snímaného obrázku (totožná s 2D pohledem - Obr (47))
- v' - je jakýsi „dohled“ kamery, v této práci je zvoleno $v' = 150[p\mu]$
- φ - reprezentuje úhel, do kterého je ještě kamera schopna zachytit prostor po stranách, v této práci je zvoleno $\varphi = 30^\circ$
- CG - je těžiště o souřadnicích $[x_1; x_2]$

Pro transformaci je použit princip popsáný v 3.4. Opět se předpokládá, že je známa poloha těžiště a rotace vzhledem ke světovému souřadnému systému (x_0, y_0) . Nejprve jsou vypočítány souřadnice vrcholů v souřadném systému lichoběžníku (x, y) :

$$A = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -\frac{s}{2} \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \frac{s}{2} \\ 0 \end{bmatrix}$$

(76)

$$C = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \frac{s}{2} + v' \operatorname{tg} \varphi \\ -v' \end{bmatrix}$$

$$D = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} -\frac{s}{2} - v' \operatorname{tg} \varphi \\ -v' \end{bmatrix}$$

Nyní je použit vztah (77) pro transformaci souřadnic bodu ze souřadného systému lichoběžníku (s.s. 1) do světového souřadného systému (s.s. 0):

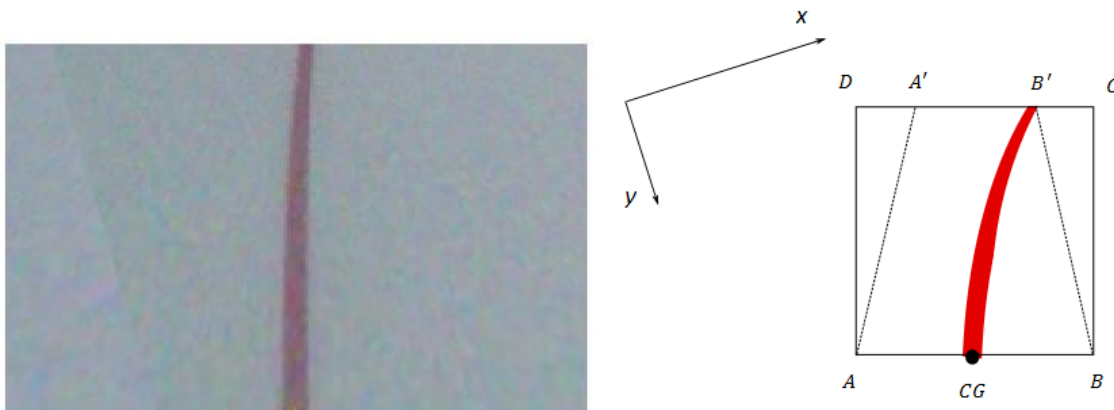
$$P^0 = r_{0,1}^0 + R_1^0 \cdot P^1 \quad (77)$$

kde:

- P^0 - jsou souřadnice bodu v s.s. 0
- P^1 - jsou souřadnice bodu v s.s. 1
- $r_{0,1}^0$ - je vektor posunutí počátku s.s.1 vůči počátku s.s. 0; v tomto případě $r_{0,1}^0 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
- R_1^0 - je matice rotace popisující natočení s.s. 1 vůči s.s. 0, sloupce této matice jsou směrové vektory os s.s. 1 vyjádřené v souřadnicích s.s. 0; v tomto případě

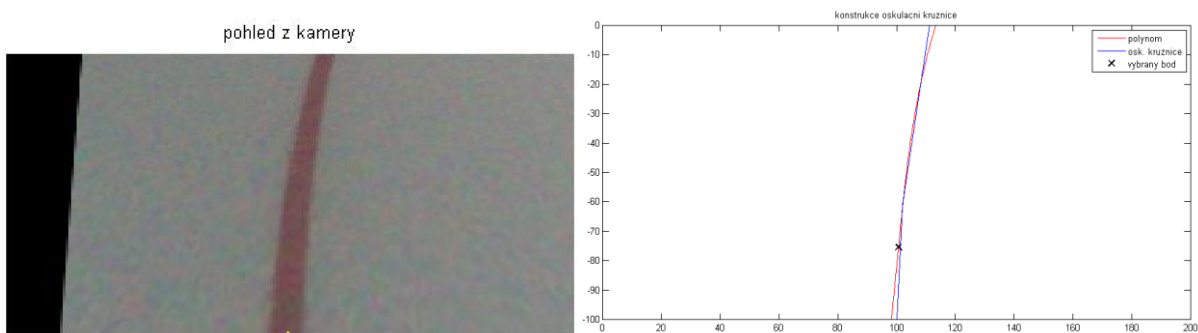
$$R_1^0 = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \text{ (levotočivý systém!)}$$

Stejným způsobem jako je popsáno v podkapitole 5.2 se uvedená oblast transformuje na obdélník. Výsledkem je obrázek, který působí dojmem perspektivního pohledu do prostoru. To je dáno zkosením původně kolmých úseček AA' a BB' z Obr. (63) blíže ke středu nového obrázku. (Obr. (64)) Tento způsob simulace pohybu je daleko bližší reálnému využití než způsob popsáný v podkapitole 5.2.



Obr. (64): Výsledný perspektivní pohled z kamery a geometrický náčrtek.
(Poznámka: Žlutý bod na dolním okraji obrázku reprezentuje těžiště robotu.)

Změnou pohledu z imaginární kamery musely být upraveny i některé dílčí kroky v celém algoritmu. Není například rozumné počítat oskulační kružnici ve středu obrázku, když už v něm neleží těžiště robotu. Nabízí se použít pro výpočet prostřední bod dolního okraje obrázku, nad kterým je uvažováno těžiště, avšak aproximovaný polynom může být v tomto bodu velmi nepřesný (viz Obr (54)) a vypočtená oskulační kružnice může dokonce i implikovat řízení na jinou stranu, než by bylo potřebné. Řešením může být výpočet oskulační kružnice ve čtvrtině výšky obrázku (Obr. (65)), kde je polynom již přesnější a chyba způsobená vzdáleností od těžiště relativně malá.

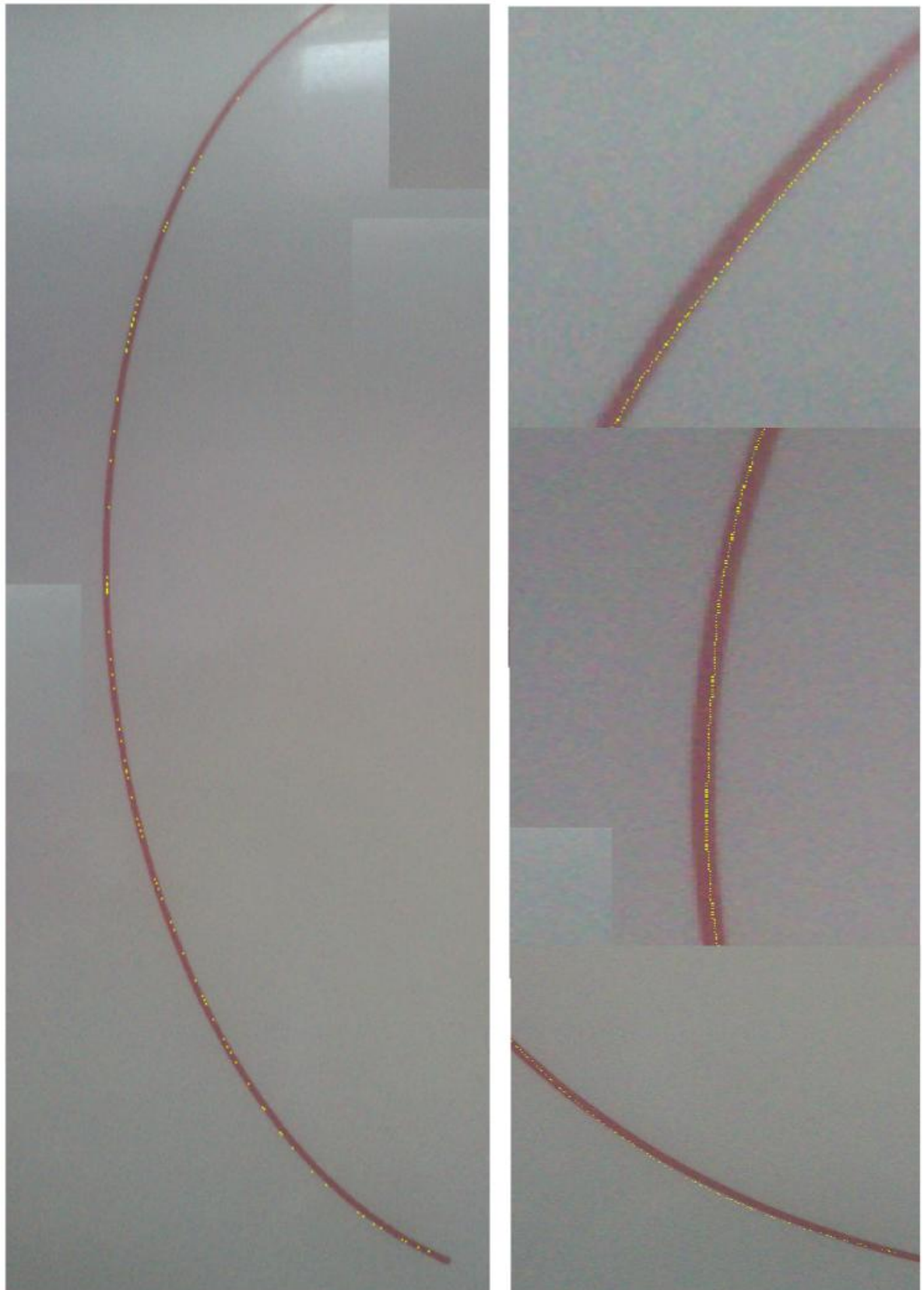


Obr. (65): Konstrukce oskulační kružnice pro perspektivní pohled z robotu.

Změněny musely být i parametry použitých PD regulátorů, a to konkrétně následujícím způsobem (78):

$$\begin{aligned}
 K &= 5,5 \cdot 10^{-4} \\
 T_D &= 5 \cdot 10^{-2} \\
 \tau &= \frac{T_D}{5,2} = 9,6 \cdot 10^{-3} \\
 K_D &= K T_D = 2,75 \cdot 10^{-5} \\
 (K_{dx} &= 1,2; K_{smer} = 1,0)
 \end{aligned}
 \tag{78}$$

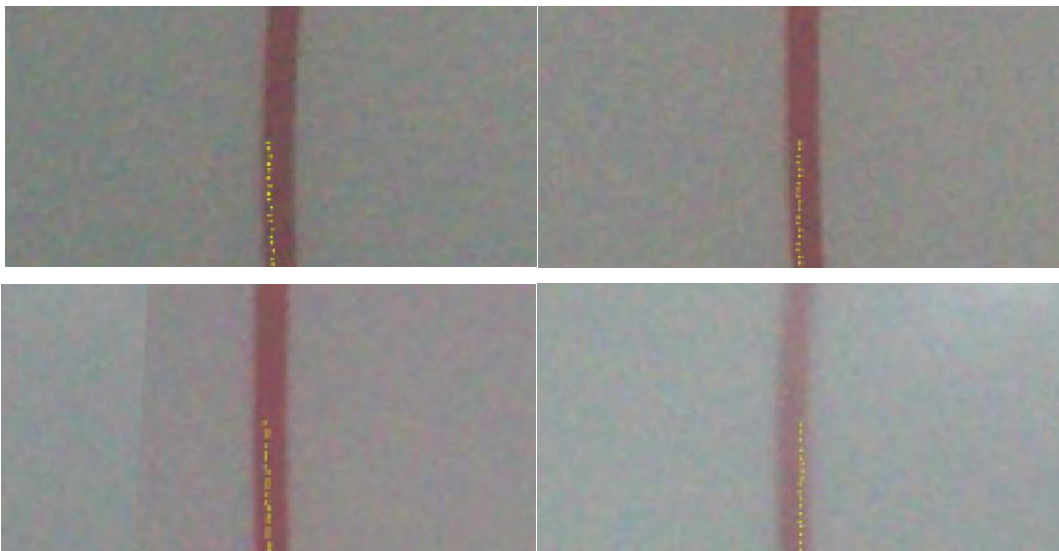
Výsledek simulace s perspektivním pohledem z kamery (Obr. (66)):



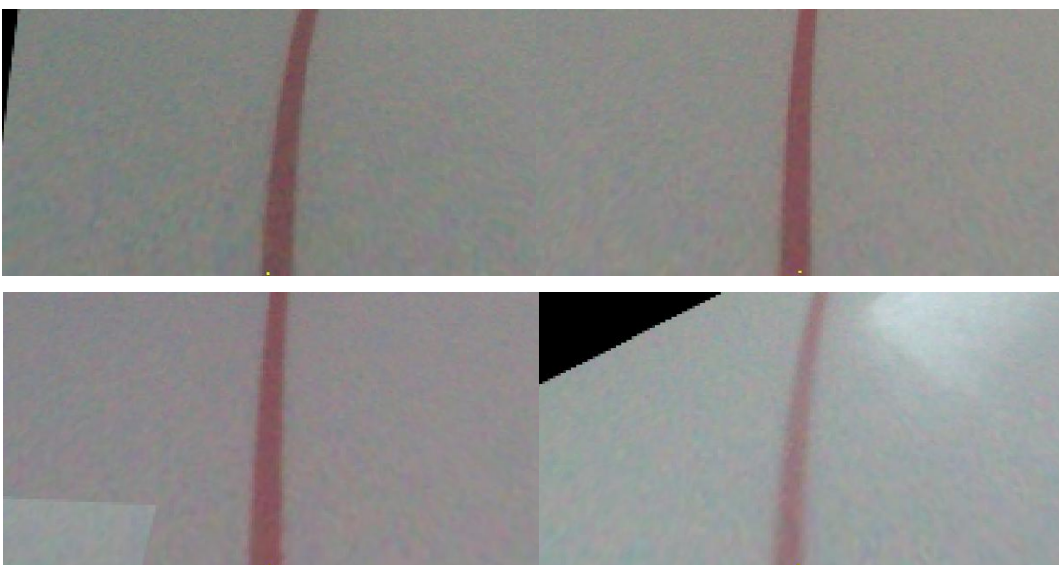
Obr. (66): Simulace s PD regulátory. (3D pohled z kamery)

5.7 Tvorba videozáznamu

Jelikož by bylo dobré zkontrolovat pohyb robotu přímo z pozice „jeho řidiče“, součástí práce na přiloženém CD je i několik videozáznamů ve formátu AVI. Jednotlivé obrázky pořízené z robotu (tenzory) se poskládají za sebe a vytvoří obtížně představitelnou čtyřrozměrnou matici. Ta se použije jako vstup pro funkce *immovie* a *movie2avi* a tím vznikne kontrolní video. Protože některé přehrávače prohazují kanály barevného modelu RGB, videa se ukládají jak ve formátu RGB, tak ve formátu BGR. Níže jsou uvedeny ukázky z pořízených videí s vnímáním okolí ve 2D i 3D. Na obrázcích (Obr. (67,68)) jsou dobře patrné změny odstínu čáry vlivem měnícího se okolního prostředí:

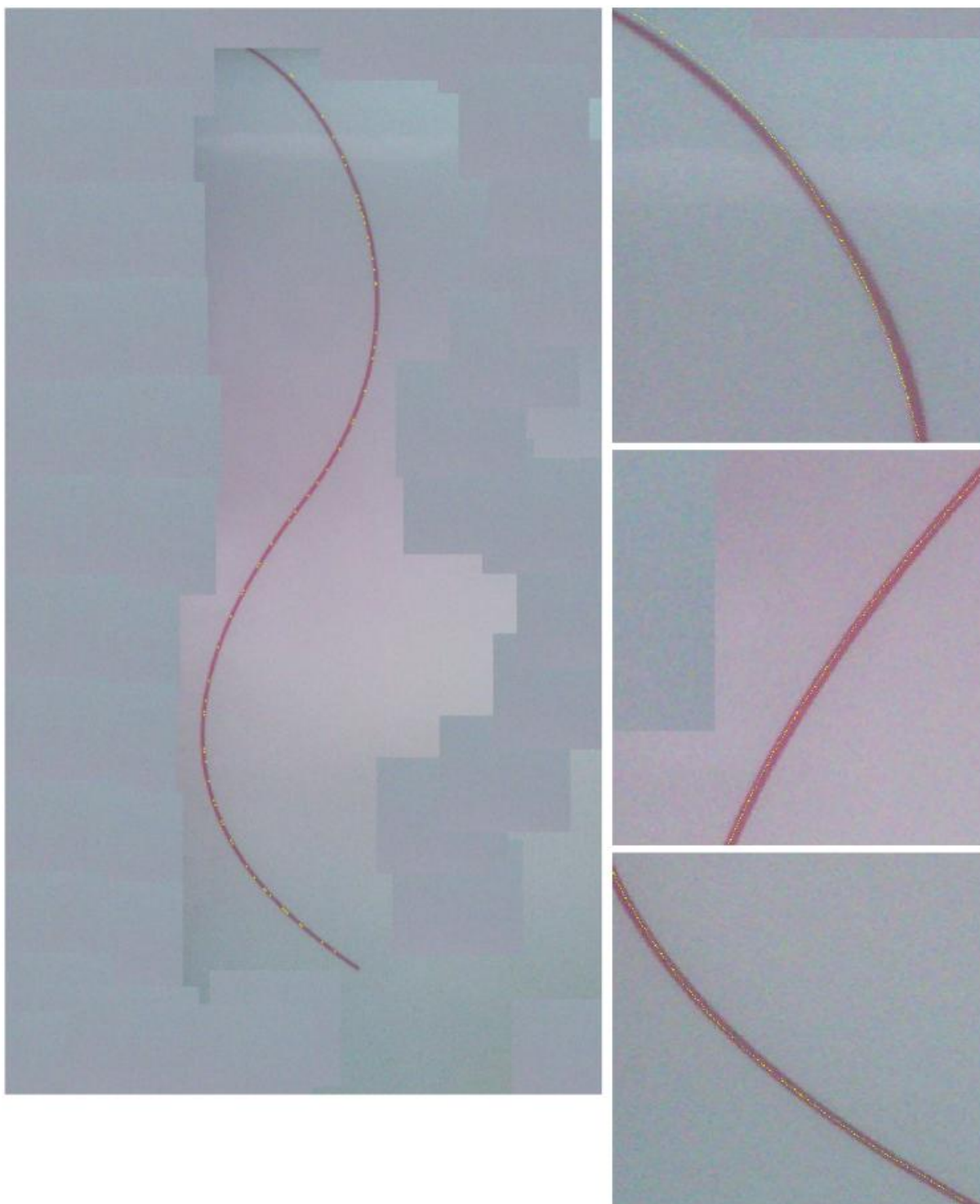


Obr. (67): Ukázky z přiloženého videa. (2D)

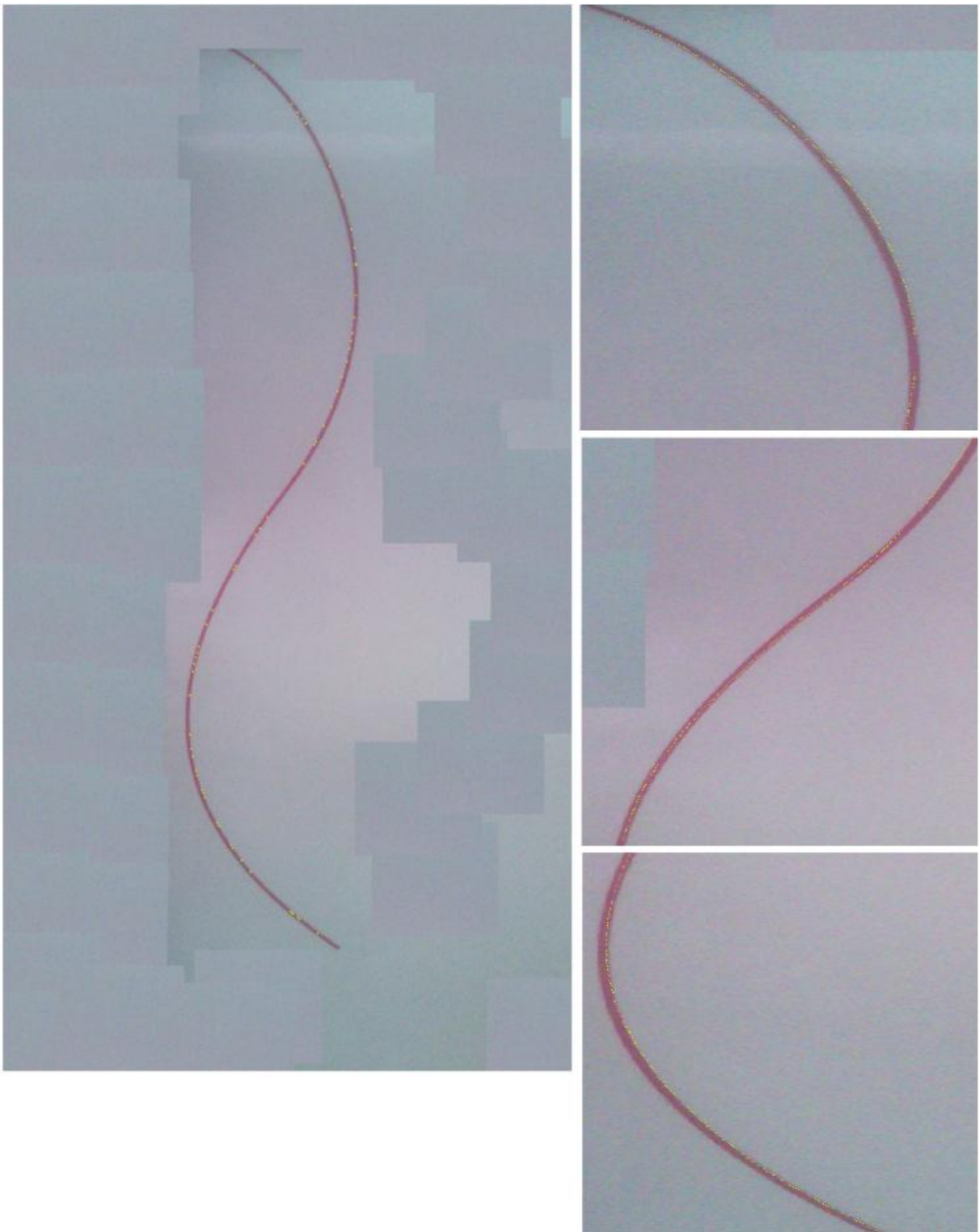


Obr. (68): Ukázky z přiloženého videa. (3D)

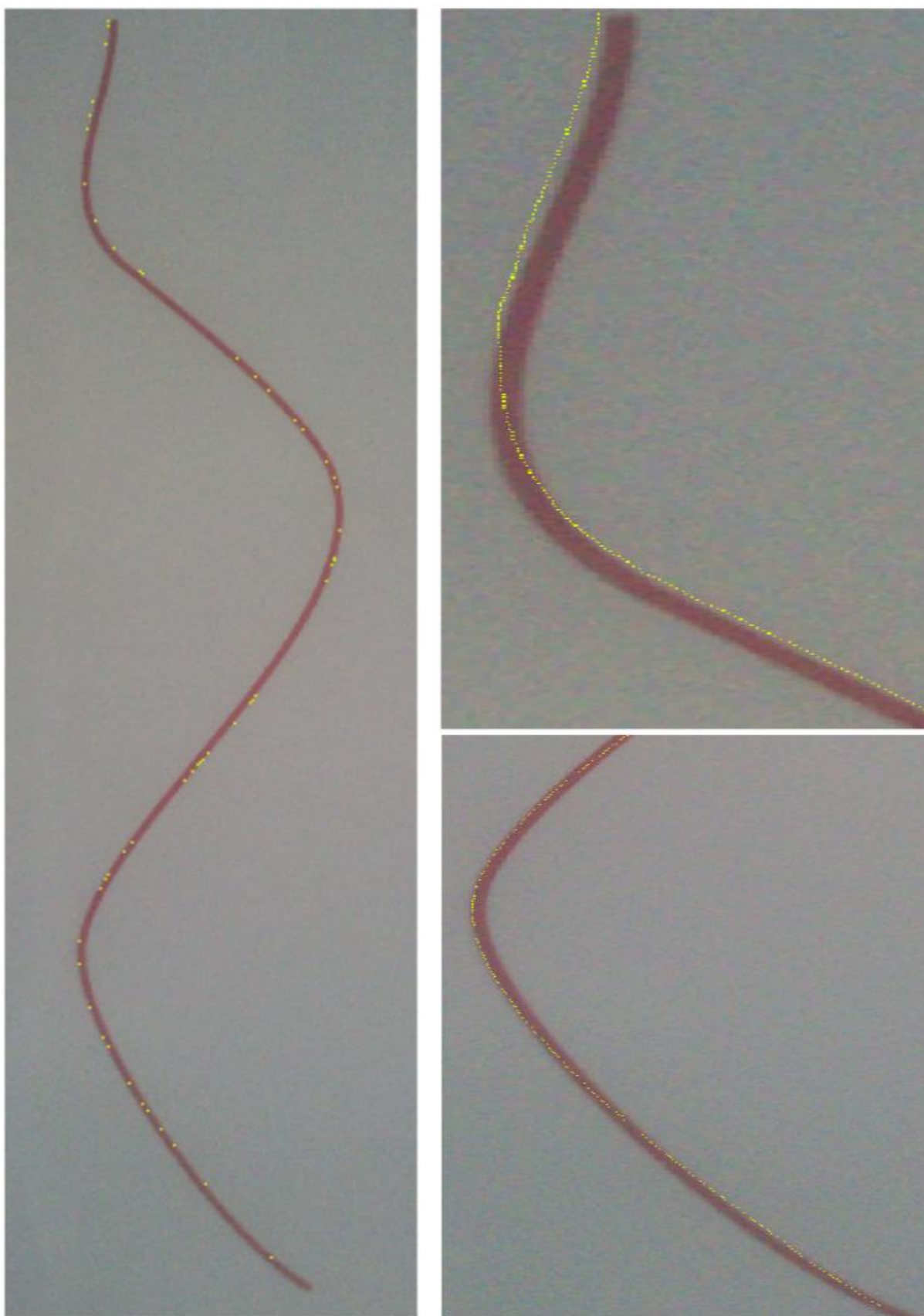
5.8 Testování na různých křivkách



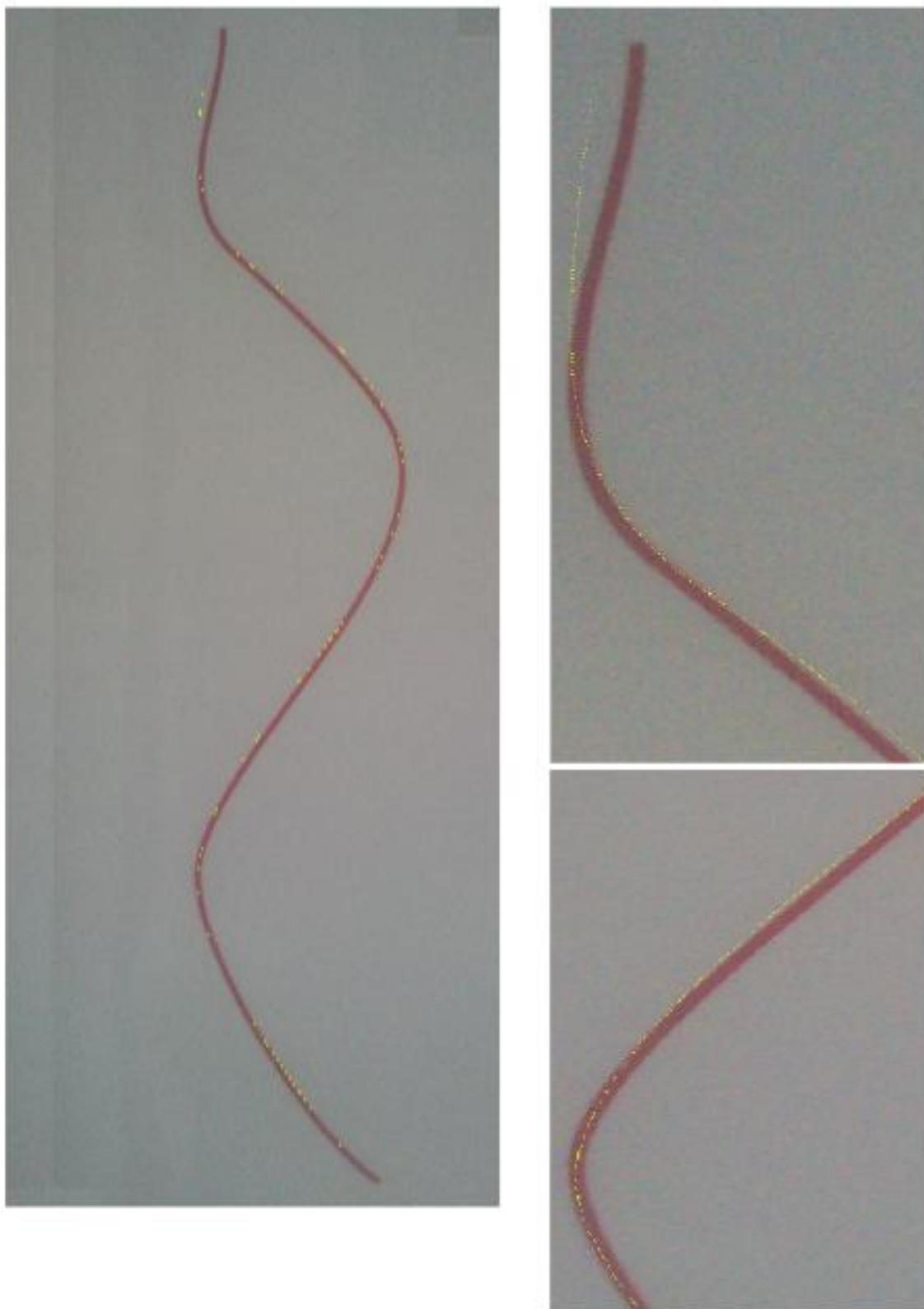
Obr. (69): Testovací křivka 2. (pohled kamery 2D)



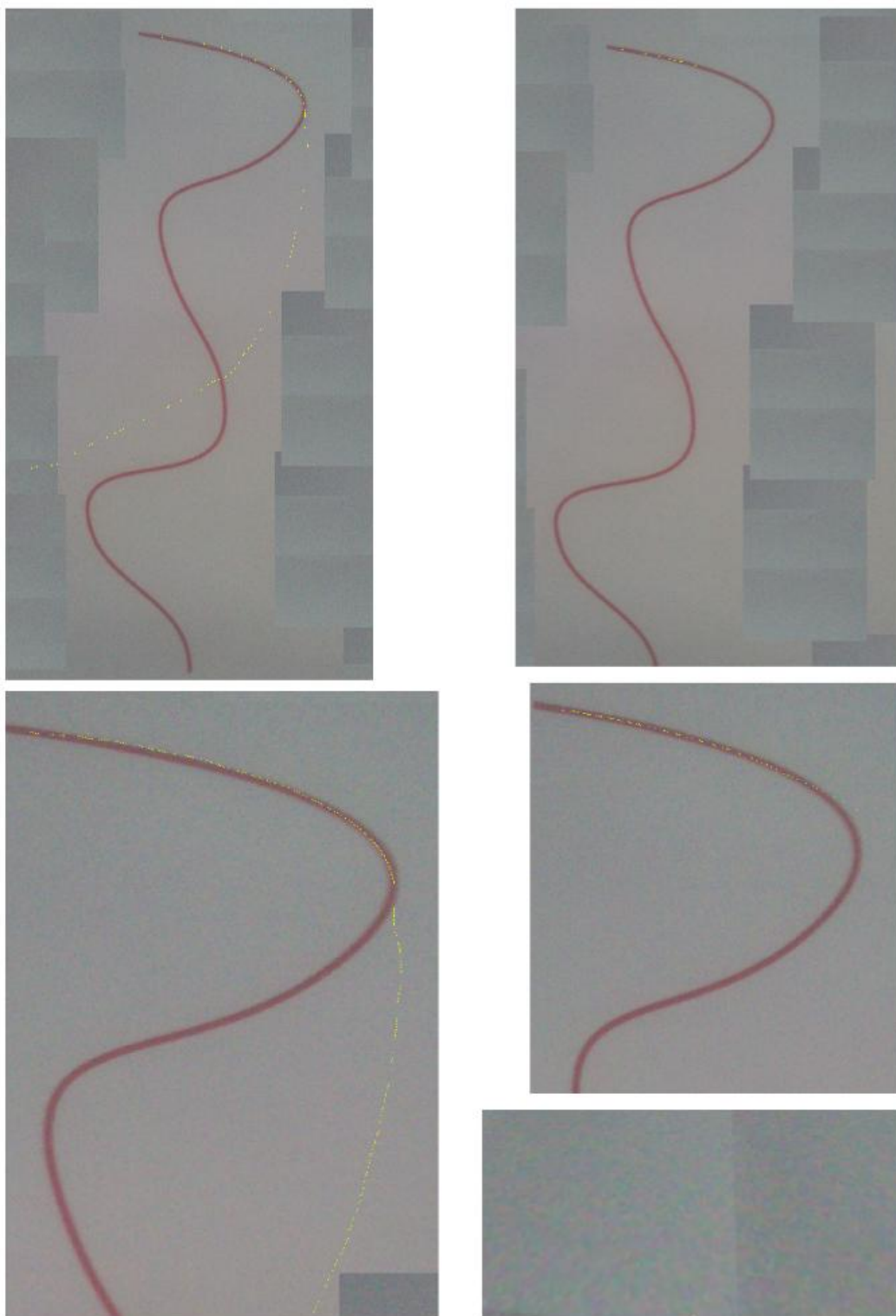
Obr. (70): Testovací křivka 2. (pohled kamery 3D)



Obr. (71): Testovací křivka 3. (pohled kamery 2D)



Obr. (72): Testovací křivka 3. (pohled kamery 3D)



*Obr. (73): Testovací křivka 4: pohled kamery 2D - vlevo, pohled kamery 3D - vpravo,
(vpravo dole záběr z 3D kamery, který způsobil zastavení algoritmu)*

Z uvedených příkladů vyplývá, že robot dokáže sledovat testované křivky s poměrně uspokojivými výsledky. Jak je zřejmé z Obr (73), nedokáže projet zatáčku s příslušnou oskulační kružnicí o poloměru asi 1cm, což vzhledem k daným definovaným rozměrům robotu není špatný výsledek.

Vizuální kontrola pohybu robotu je dobrým mechanismem pro verifikaci vytvořeného algoritmu řízení, avšak pro lepší představu je zde navíc vytvořeno kritérium (79) ve tvaru:

$$J = \frac{1}{N} \sum e_{dx}^2 \quad (79)$$

kde

- N - je počet kroků algoritmu
- e_{dx} - je regulační odchylka definovaná v (73)

Hodnoty kritérií pro testované křivky ((60), (66), 5.8) jsou uvedeny v Tabulce (2):

	testovací obr. č. 1 (Obr. (60))		testovací obr. č. 2 (Obr. (69))		testovací obr. č. 3 (Obr. (71))		testovací obr. č. 4 (Obr. (73))	
	2D	3D	2D	3D	2D	3D	2D	3D
J	5,5337	13,6140	9,2292	10,1521	20,5933	68,8162	>100	>100

Tabulka (2): Hodnoty kritérií pro testované křivky.

Obecně tedy platí, že výsledky s 3D pohledem z kamery jsou o něco horší, než výsledky z kamery simulující 2D pohled, což je do jisté míry logické.

Závěr

Uvedená diplomová práce nastiňuje jeden z možných způsobů řízení mobilního kolového robotu s dvěma nezávisle řízenými koly za účelem sledování libovolně barevné křivky, která představuje svar potrubí. Daná křivka by z podstaty problému měla být spojitá, hladká a poloměry křivosti jednotlivých oblouků by neměly být příliš malé.

Pro rozpoznání křivky je použito nejprve modelování barvy pomocí Gaussovy křivky získané z obrázku ve formátu HSV, poté prahování pomocí věrohodnostní funkce. Stěžejní části řídicího algoritmu jsou oskulační kružnice, na jejichž základě jsou generovány požadované úhlové rychlosti jednotlivých kol tak, aby robot co nejlépe kopíroval danou trajektorii. Pro zlepšení kvality řízení je použita vizuální zpětná vazba. Vzniklé regulační odchylky jsou korigovány pomocí PD regulátorů.

Jelikož nebyl k dispozici reálný model kolového robotu, na kterém by bylo možno navržený algoritmus otestovat, je v této práci vytvořena jakási ohraničená virtuální 2D realita, ve které se může robot volně pohybovat. Výřezy z takto získané virtuální reality, které simulují pohled z imaginární kamery umístěné na robotu, jsou pak vstupem pro rozpoznávání křivky. Navíc je zde implementováno rozšíření o perspektivní pohled z kamery, tedy vlastně o náhražku 3D vidění z robotu.

Použitý algoritmus řízení nepředpokládá dokonalou znalost umístění robotu v prostoru ani znalost parametrického popisu celé sledované křivky. Pro generování akčních veličin jsou použity pouze vstupy z imaginární kamery.

Parametry regulátorů byly voleny experimentálně. Snahou autora bylo zapojit do řešení problému sofistikovanější metody, avšak ty v naprosté většině nepřinesly uspokojivé výsledky. V literatuře se vyskytují například metody řízení založené na nelineární zpětné vazbě. Použití nelineární zpětné vazby na nelineární systém je však velice problematické a zaručit v jakémkoliv smyslu stabilitu systému je v podstatě nemožné, a to i pro opakované simulační experimenty, natož pak pro reálné použití. Dalším způsobem řízení kolových robotů, které by sledovaly danou křivku, může být například řízení založené na fuzzy logice nebo neuronových sítích. Tyto metody jsou také popsány v některých článcích, avšak jejich základ je heuristický. Ve velkém množství akademických publikací se navíc předpokládá, že

se robot pohybuje v kartézském souřadném systému a je známa jeho přesná poloha a natočení, což je ale pro účely této práce naprosto nereálný předpoklad. Návrh řídicího algoritmu pomocí aproximace naprahané čáry polynomem a následná konstrukce oskulační kružnice v daném bodu polynomu se zdá v porovnání s výše uvedenými metodami akademicky korektnější, i přestože jsou regulační odchylky korigovány PD regulátory s experimentálně získanými parametry.

Další kroky výzkumu by nejspíše vedly k otestování uvedeného algoritmu na reálném zařízení. Návrh řízení je také zatížen několika předpoklady, jejichž oslabením by se docílilo robustnějšího řešení problému. Například povrch, na kterém se robot pohybuje, by nemusel být dokonale hladký, mohly by se uvažovat třecí a odporové síly nebo prokluz kol. Osy kol by také nemusely procházet těžištěm robotu apod. Změny by mohly nastat i v algoritmu rozpoznávání čáry. Například by mohl být implementován nějaký zaváděcí algoritmus, díky němuž by robot najel na čáru v případě, že by na ní od začátku nestál. V uvedeném řešení se také předpokládá, že z robotu je vždy vidět jen jedna barevná křivka. Není ošetřen případ, kdy by bylo na obrázku z imaginární kamery viditelných více čar stejné barvy. V neposlední řadě není vůbec diskutována možnost, že jsou trubky svařeny například kolmo do sebe. V tomto případě by byl svar pro robota dané architektury nepřístupný a musely by být provedeny některé dílčí úpravy. Například by robot mohl být vybaven ramenem, na jehož konci by bylo umístěno další ultrazvukové zařízení, které by takto nedostupný svar kontrolovalo. Všechny tyto úpravy však přesahují možnosti této práce.

Celkové řídicí schéma stejně jako použitý způsob rozpoznávání čáry vycházejí ze známých a poměrně hojně používaných metod využívaných v automatickém řízení, resp. počítačovém vidění. Těžiště a hlavní přínos práce tkví především ve způsobu propojení vizuální části s částí řídicí pomocí vizuální zpětné vazby, způsobu řízení robotu, jehož základem jsou oskulační kružnice, a vytvořením virtuální reality ve 2D i 3D, kterou lze použít pro testování libovolných architektur kolových robotů bez nutnosti konstrukce prototypových zařízení.

Použitá literatura

- [1] Villela, V.J.G., Parkin, R., Parra, M.L., González, J.M.D., Liho, M.J.G.: *A wheeled mobile robot with obstacle avoidance capability*, ISSN 1665-7381, Mexico, 2004
- [2] Grepl, R.: *Modelování mechatronických systémů v Matlab/SimMechanics*, BEN, 2007
- [3] Mathworld Wolfram [online], dostupné na webových stránkách: <http://mathworld.wolfram.com>
- [4] Šonka, M., Hlaváč, V., Boyle, R.: *Image Processing, Analysis and Machine Vision*, ISBN 0-534-95393-X, PWS / International Thomson Press, 1999
- [5] Wikipedia [online], dostupné na webových stránkách: <http://en.wikipedia.org>
- [6] Papastavridis, J. G.: *Analytical Mechanics: A Comprehensive Treatise on the Dynamics of Constrained Systems; for Engineers, Physicists, and Mathematicians*, Oxford University Press, Oxford, ISBN 0-19-512697-1, 2002
- [7] Bláha, L.: *Stabilizace a automatické řízení pohybu dvoukolky*, diplomová práce, 2007
- [8] Švejda, M.: Přednášky z předmětu KKY/URM, 2013, dostupné na webových stránkách: <http://home.zcu.cz/~msvejda/URM/>
- [9] Melichar, J.: Přednášky z předmětu KKY/LS1, KKY/LS2, 2011, dostupné na webových stránkách <http://www.kky.zcu.cz/cs/courses>
- [10] Schlegel, M.: Přednášky z předmětu KKY/PS, 2013
- [11] Stránky předmětu KIV/ZPG, dostupné na webových stránkách <http://herakles.zcu.cz/education/zpg/cviceni.php>
- [12] Malis, E.: *Survey of vision-based robot control*, ENSIETA European Naval Ship Design Short Course, INRIA, Sophia Antipolis, France, 2002
- [13] Horáček, L. a kol.: *Přehled současných manipulátorů pro nedestruktivní zkoušení a jejich příslušenství*, ÚJV Řež a.s., Plzeň, 2009
- [14] Google [online], dostupné na webových stránkách <http://www.google.cz>