# Návrh nelineárneho regulátora pre diferenciálny podvozok

Mám nelineárny systém diferenciálneho podvozku mobilného robota opísaný diferenciálnymi rovnicami.

Za stavové premenné systému, ktoré chcem riadiť (stabilizovať) považujem len súradnice X a Y.

Uhol natočenia theta nie je potrebné nijako špecificky riadiť, jeho hodnota môže vo výsledku ľubovoľná. Samozrejme musí byť taká, aby celý systém bol stabilný a konvergoval do rovnovážneho stavu (poloha musí konvergovať k 0,0).

Vstupom do systému sú signály vt a omega – translačná rýchlosť a rýchlosť rotácie. Výstup X aj Y sú vlastne integrátory nelineárne závislé od uhla theta.

Cieľom riadenia je stabilizovať tento systém do polohy 0,0 vychádzajúc z ľubovoľných počiatočných podmienok, a to riadením vstupov vt a omega. S použitím Lyapunovovej teórie stability by som chcel navrhnúť nelineárne riadenie. K dispozícii mám samozrejme spätnú väzbu od všetkých stavových veličín (poloha,uhol).

Pre splnenie podmienok použitia Lyapunovovej teórie stability je nutné uvažovať stabilizáciu do bodu (0,0). Všeobecná stabilizácia do iného ľubovoľného bodu sa dá dosiahnuť rovnakým spôsobom ale s použitím transformačných vzťahov pre súradnice, ktoré posúvajú stred súradnicového systému do žiadaného bodu.

Aby sa to ešte trochu skomplikovalo vstupy budú musieť byť mierne modifikované. Nemám totiž k dispozícii priamo možnosť meniť rýchlosť otáčania omega ale iba polomer zatáčania R.

Rýchlosť rotácie potom závisí aj od translačnej rýchlosti . Translačnú rýchlosť vt viem nastavovať priamo a môže byť ako kladná, tak aj záporná.

Potom sa sústava diferenciálnych rovníc pozmení (vt a R sú signály akčného zásahu)

## Riešenie:

Moje riešenie vychádza z obmedzených znalostí teórie stability a návrhu nelineárneho riadenia.

Ako prvé som transformoval problém stabilizácie súradníc X,Y na problém stabilizácie kartézskej vzdialenosti polohového vektora v zmysle vzdialenosti polohy robota od žiadanej polohy. Eliminujem tým tak jednu rovnicu.

### Stabilizácia vzdialenosti robota:

Potom časová derivácia – diferenciálna rovnica vzdialenosti robota:

Za časové derivácie súradníc viem dosadiť známe výrazy z diferenciálnych rovníc systému:

Rovnica pre uhol ostáva nezmenená

Problém ostáva, že uhol nechcem stabilizovať do nuly, nechcem ho stabilizovať vôbec, chcem aby bol presne taký, že vzdialenosť robota od žiadanej polohy bude konvergovať čo najrýchlejšie k nule.

Preto zatiaľ riešim stabilitu nezávisle len pre rovnicu vzdialenosti:

Zvolím Lyapunovovu funkciu v kvadratickom tvare:

Potom:

Derivácia LF pozdĺž trajektórie systému:

Ak chcem aby bolo záporne semidefinitné a tým pádom bola vzdialenosť *z*stabilná, musím akčným zásahom zabezpečiť aby výraz bol záporný. Najjednoduchšie to spravím takto:

Potom:

Kde *m* je konštanta zosilnenia.

Akčný zásah má pre akúkoľvek polohu robota a uhol natočenia to správne znamienko a veľkosť. Problém je, že bez riadenia polomeru zatáčania R vzdialenosť nikdy nekonverguje k 0.

Konverguje len k minimálnej vzdialenosti aká je pre daný natvrdo nastavený polomer zatáčania R možná. Čo je vlastne v poriadku, pretože nič iné sa pri riadení len translačnej rýchlosti čakať nedá.

### Stabilizácia uhla natočenia robota:

Otázka je, ako riadiť správne polomer zatáčania R ?

Žiadanú hodnotu uhla natočenia robota nemám explicitne definovanú.

Jediný zmysel dáva určiť si žiadanú hodnotu uhla natočenia tak, že to bude vždy opačná hodnota k uhlu (akoby fáza komplexného čísla) ktorý dostaneme transformáciou súradníc polohy robota X,Y do polárnych súradníc (atan(y/x)). Chcem teda smerovať robota vždy opačným smerom aký smer má jeho polohový vektor.

Uhol polohového vektora (fáza polohového vektora) robota označím ako alfa:

Označím potom chybu natočenia robota (výchylka uhla natočenia) ako rozdiel aktuálneho natočenia a fázy polohového vektora.

Pre dynamiku regulačnej odchýlky uhla natočenia robota bude platiť nasledujúca rovnica:

Túto diferenciálnu rovnicu môžem teraz chcieť stabilizovať k nule, teda požadujem nulový rozdiel uhlov (natočenia robota a smerového uhla).

Za ešte dosadím známy výraz z pôvodného systému.

Následne viem ešte dosadiť už známu translačnú rýchlosť robota, ktorá vyplynula z riešenia stabilizácie vzdialenosti  Lyapunovovou rovnicou.

Ostáva určiť časovú deriváciu uhla polohového vektora:

X a Y sú časovo premenlivé signály a preto musíme derivovať nasledovne:

Dynamika odchýlky uhla natočenia bude:

Zvolíme kandidáta na Lyapunovovu funkciu v tvare:

Potom:

Derivácia LF pozdĺž trajektórie systému bude:

Celý problém stabilizácie výchylky natočenia robota stojí a padá na znalosti samotnej výchylky natočenia. Prirodzene nie je problém ju počítať ako:

Kde všetky veličiny poznám. Problém je práve s arkusom tangensom, ktorý je síce pekné použiť ale pri posielaní podielu do funkcie sa mi stratí informácia o kvadrante.

Aj toto by sa dalo riešiť použitím funkcie atan2, ktorá ako argument neberie podiel zložiek ale obe zložky nezávisle – atan2(y,x).

Aj napriek tomu vzniká ďalší problém. Konkrétne to, že atan2 má nespojitosť v oblasti kedy výstupný uhol prechádza a skáče nespojito medzi hodnotami a . Čo je nepoužiteľné, pretože potrebujem spojitý výstup v celej pracovnej oblasti súradníc.

Aj keby som tento problém vyriešil, potrebujem dosiahnuť zápornú semidefinitnosť derivácie Lyapunovovej funkcie pozdĺž trajektórie systému pre zaručenie stability.

Ako ale akčným zásahom R zabezpečiť zápornú semidefinitnosť tohto výrazu ? Je to pomerne komplikovaná záležitosť.

Preto som vymyslel obchádzku pomocou aproximácie výchylky uhla.

### Aproximácia chyby(výchylky) natočenia robota:

Ak poznám uhlovú rýchlosť (rotácie) polohového vektora robota (už som ju odvodil).

Môžem tvrdiť, že jej znamienko, a v princípe aj veľkosť (bez prenásobenia vt) je úmerná práve výchylke natočenia (chybe natočenia robota).

To prakticky znamená, že keď polohový vektor (v zmysle X,Y) rotuje okolo žiadanej polohy, tak sme nesprávne natočení (kolmo k cieľu). Naopak ak robot smeruje správne k cieľu, polohový vektor robota nerotuje ale iba sa zmenšuje alebo zväčšuje (ideme priamo na cieľ).

Preto je znamienko rotácie polohového vektora rovnaké ako znamienko chyby uhla natočenia a ich veľkosti sú si úmerné, samozrejme bez uvažovania absolútnej veľkosti člena, ten sa správa ako zosilnenie rotácie.

Potom výraz pre Lyapunovou rovnicu

prejde to tohto tvaru:

Čo po úprave bude:

A platí:

Kde výraz

Je vždy kladný a v podstate zastupuje druhú mocninu výchylky

Potom ostáva zabezpečiť, aby aj výraz

Bol vždy záporný

Najjednoduchšie to realizujem:

Kde viem ľahko vypočítať aj bez použitia atan2, ako bolo už ukázané:

Kde *k* je konštanta zosilnenia

### Kód pre regulátor:

Simulačná schéma na overenie:





Kód regulátora napríklad v Matlabe bude:

function [vt,R] = robot\_regulator(X,Y,X\_,Y\_,fi)

m=0.5;

k=1;

vt=-m\*((X-X\_)\*cos(fi)+(Y-Y\_)\*sin(fi));

z\_squared=(X-X\_)^2+(Y-Y\_)^2;

delta=((X-X\_)\*sin(fi)-(Y-Y\_)\*cos(fi))/z\_squared;

R=k/delta\*abs(vt);

end

### Simulačné výsledky:







