

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PROTIKOLÍZNY SYSTÉM PRE MOBILNÉHO ROBOTA

BAKALÁŘSKÁ PRÁCE

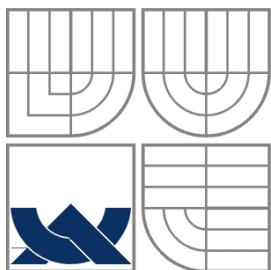
BACHELOR'S THESIS

AUTOR PRÁCE

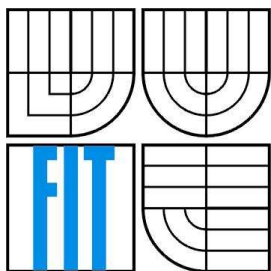
AUTHOR

DANIEL DZURO

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PROTIKOLIZNÍ SYSTÉM PRO MOBILNÍHO ROBOTA

COLLISION AVOIDANCE SYSTEM FOR MOBILE ROBOT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL DZURO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. David Herman

BRNO 2011

Abstrakt

Tato práce pojednává o cíli vytvořit protikolizní systém pro mobilního robota, který je při jízdě často vystavený kolizím s jinými objekty. Takovýmto systémem by se ochránily jeho drahé součásti. Konkrétně bylo potřeba navrhnout systém předčasné reakce a zásahu do systému, který by kolizi dokázal včas zabránit. Jednalo se nejen o použití brzdového systému robota na zastavení či brzdění, ale i o kontrolu předvídání směru vozidla. Pro tento účel byl využit hlavně laserový dálkoměr a ultrazvukové sonary.

Abstract

This bachelor's thesis deals with the anti-collision system for mobile robot which is often liable to the collisions with other objects. Expensive parts of the robot could be saved thanks to the system. The main issue of the thesis was to design the system of premature reaction and its application to the existing system which would help prevent the collision on time. Not only the breaking system was used to stop or slow down the vehicle but also the cooperation with the prediction of the vehicle's direction control was needed. Laser rangefinder and ultrasonic sonar were mainly used for this purpose.

Klíčová slova

protikolizní systém, laserový dálkoměr, sonar, antikolizní systém, robot Quido, lidar, PID regulace, SICK LMS120, brzdící dráha, kritická brzdící vzdálenost

Keywords

collision avoidance, laser radar, sonar, robot Quido, PID regulation, low-cost laser radar, SICK LMS120, breaking distance, critical breaking distance, critical warning distance

Citace

Dzuro Daniel: Protikolizní systém promobilního robota, bakalářská práce, Brno, FIT VUT v Brně, 2011

Antikolízny systém pre mobilného robota

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod Ing. Davida Hermana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Daniel Dzuro

5. mája 2011

Poděkování

Moje velké poděkování patří Ing. Hermanovi za jeho intenzivní pomoc při řešení této bakalářské práce. Svými radami, ochotou a neustálou podporou ve velké míře přispěl k jejímu úspěšnému dokončení.

© Daniel Dzuro, ROK 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Bezpečnosť vozidiel.....	3
2.1 História	3
2.2 Protikolízne systémy automobiliek.....	5
3 Senzorický systém robota	8
3.1 Externé senzory	8
3.2 Interné senzory.....	11
4 Protikolízny systém.....	15
4.1 Vlastný návrh.....	16
4.2 Kinematický model robota	18
4.3 Brzdná dráha.....	21
5 Regulácia a riadenie	24
5.1 PID regulátor	24
5.2 Riadenie rýchlosti	27
5.3 Riadenie natočenia kolies	28
6 Implementácia.....	29
6.1 Knižnica MRPT	29
6.2 Hierarchia tried	29
6.3 Popis tried	30
6.4 Konfigurácia	32
7 Záver	33
Literatúra	34
Zoznam príloh.....	36
Príloha A Ukážka aplikácie	37

1 Úvod

Cieľom tejto bakalárskej práce bolo vytvorenie systému, ktorý by ochránil už existujúceho mobilného robota (ďalej tiež MR) pred zrážkami s inými objektmi. Konkrétne sa jedná o mobilného robota nazvaného *Quido*. S týmto robotom sa podarilo tímu *Roboauto* získať viacero ocenení na rôznych súťažiach. Súčasťou robota je mnoho zariadení, ktoré sú pomerne nákladné a ich zničenie je neprípustné. Bolo potrebné implementovať nielen systém, ktorý by varoval pred hroziacou zrážkou, ale takisto patrične zasahoval do riadenia, aby kolízii dokázal zabrániť.

Tým, že je robot postavený na type podvozku aké majú automobily, dovolili sme si hľadať inšpiráciu v už existujúcich funkčných systémoch. Automobilky ako je Volvo, Mercedes-Benz, BMW, Audi či rad ďalších majú za sebou už niekoľko rokov vývoja tohto systému. Nápady sme hľadali v princípe ich fungovania, aby sme pomocou nášho dostupného hardwaru mohli napodobniť ich funkcionality.

V kapitolách tejto práce sa na začiatku zoznámime s pojmom protikolízny systém, uvedieme si históriu bezpečnosti, do ktorej protikolízny systém nepochybne patrí. Popíšeme si ako a kde sa dnes využíva v praxi, existujúce algoritmy a akým spôsobom sa ho my budeme snažiť implementovať. Postupne sa zoznámime s použitým hardwarovým systémom, sústredíme sa na teoretické predpoklady fungovania celého systému ako je geometria robota, brzdná dráha, odometrické veličiny a iné faktory. Následne sa zameriame na samotný návrh a princíp protikolízneho systému. Implementácia a popis tried je uvedený v poslednej časti práce.

2 Bezpečnosť vozidiel

Od vzniku automobilov išiel vývoj bezpečnosti vozidiel prudko nahor. Mnoho automobiliek vyvíja vlastné systémy ochrany či už pasažierov alebo automobilov ako majetku. Vznikali bezpečnostné prvky ako pásy na sedadlách, airbagy, kontroly mŕtvych uhlov vodiča atď. [1].

Všetky tieto bezpečnostné prvky rapídne redukovali zranenia a následky po dopravnej nehode. Napriek tomu neboli štatistiky optimistické. Výrobcovia automobilov sa preto snažili zamerať na bezpečnosť z trochu iného uhla pohľadu a síce nie redukovaním následkov nehôd, ale priamo jej predchádzaním. Vývojom nových technológií elektroniky sa využívali systémy ako ABS (z nem. *Antiblockiersystem*), ESP (z nem. *Elektronisches Stabilitäts Program*), ASR (z nem. *Antriebsschlupfregelung*) a iné, ktoré boli ešte v minulosti súčasťou výbavy automobilov vyššej kategórie. Nepochybne aj tieto prvky dvihli úroveň bezpečnosti, avšak ešte účinnejšie by bolo zrážku predpovedať a eliminovať ju skôr ako k nej dôjde. Štatistiky ukazujú, že drvivá väčšina dopravných nehôd, cca 88%, je výsledkom nepozornosti vodiča [1]. Iné štatistiky uvádzajú, že 75% všetkých nehôd je v rýchlosti pod 32 km/h. Je to napríklad jazda v kolóne či v meste [2].

Pojem protikolízny systém vznikol prekladom anglického slova *collision avoidance* (ďalej aj CA). Význam tohto slovného spojenia v sebe zahŕňa všetko od monitorovania okolitého priestoru, cez varovania pred možnou zrážkou, až po zásah do systému riadenia. U automobilov sa od protikolízneho systému očakáva, že bude pracovať v situáciách, v ktorých vodič nestihne zareagovať alebo k nej dôjde vďaka jeho neopatrnosti.

2.1 História

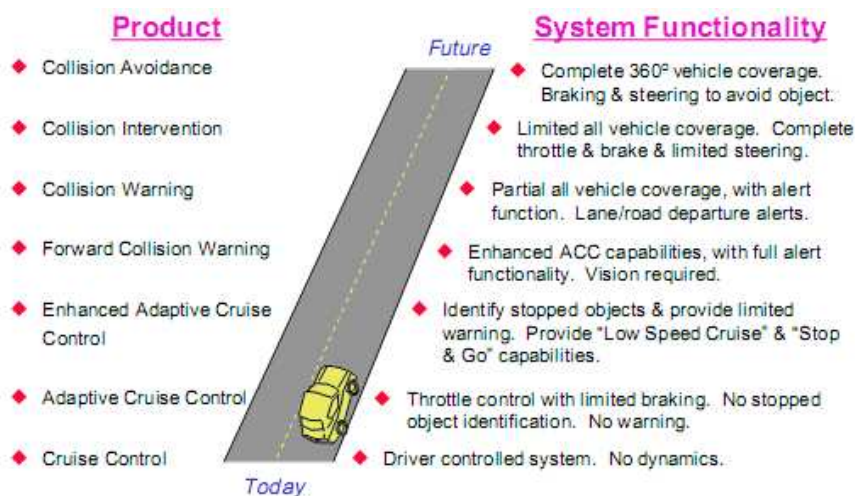
Protikolízny systém je neoddeliteľnou súčasťou autonómneho správania sa automobilu a jeho história siaha až do roku 1933 keď Norman Bel Geddes predstavil na svetovej výstave vozidlo poháňané elektrickými obvodmi vsadenými do vozovky, ktoré bolo riadenie rádiovým [3]. Ďalší významný pokus sa zaznamenal v roku 1977 v strojárskom laboratóriu v meste Tsukuba v Japonsku. Na vyhradenom priestranstve bola dosiahnutá maximálna rýchlosť 30 km/h tak, že vozidlo sledovalo trasu vyznačenú bielymi značkami. V roku 1980, robotická dodávka Mercedes-Benz, ktorá bola vybavená počítačovým videním a vytvorená tímom z Bundeswehr University of Munich v Mníchove, pod vedením Ernsta Dickmannsa, dosiahla rýchlosť 100 km/h na uliciach bez premávky. Následne, Európska komisia začala v priebehu rokov 1987 až 1995 financovanie v hodnote 800 miliónov € do projektu EUREKA Prometheus zameraného na vývoj autonómnych robotov. Zároveň v roku 1980 autonómne pozemné vozidlo (angl. *Autonomous Land Vehicle*, skr. ALV), financované organizáciou DARPA (americká organizácia zaoberajúca sa vývojom technológií pre armádu), predviedlo prvú demonštračnú jazdu, pri ktorej bol použitý laserový radar, počítačové videnie na riadenie vozidla bez prítomnosti vodiča a dosiahlo rýchlosť vyššiu ako 30 km/h [3].

V roku 1994 dvojica robotov VaMP a Vita-2 prešla viac než 1000 km po parížskej trojprúdovej diaľnici v klasickej dopravnej špičke v rýchlosti 130 km/h, hoci bolo potrebných niekoľko korekcií človekom. Tým bola prezentovaná jazda vo voľných prúdoch, jazda v kolóne, prechod medzi pruhmi s predbiehaním iných vozidiel [3].

Významným pokrokom bola predváždzacia jazda v roku 1995, keď autonómne riadený automobil Mercedes-Benz „triedy S“ prešlo 1600 km z Mníchova v Bavorsku, do Kopenhagenu v Dánsku a späť pomocou sledovacieho počítačového videnia a procesorov reagujúcich v reálnom čase. Na diaľniciach v Nemecku robot dosahoval rýchlosť 175 km/h, čo znamenalo, že auto prešlo priemerne 9 km bez zásahu človeka, t. j. 95% riadenia sa vykonávalo autonómne. V klasickej premávke sa dokázalo pohybovať medzi autami a predbiehať ich. V tom istom roku sa na univerzite Carnegie Mellon University podaril projekt nazvaný *No hands across America*. Robot prešiel 5000 km a z toho 98,2% cesty prešiel autonómne. Riadenie natáčania kolies využilo neurónové siete, ale ovládanie plynu a brzd bolo riadené človekom [3].

Projekt ARGO, v rokoch 1996 až 2001, pracoval takisto na modeli autonómneho vozidla. Počas šiestich dní po cestách v severnom Taliansku najazdilo 2000 km, priemernou rýchlosťou 90 km/h, z toho 94% času bolo auto v plnom automatickom móde. Najdlhšia trasa, ktorú prešlo bez ľudského zásahu bolo 54 km. Vozidlo malo iba pomerne lacné čierno-biele kamery a pre detekciu prostredia používalo algoritmus stereoskopického videnia [3].

Organizácia DARPA začala s organizovaním súťaží, ktorej sa zúčastňovali podobné, autonómne riadené vozidlá. V roku 2010 sa uskutočnila testovacia jazda v dĺžke 13000 km. Zo všetkých vozidiel, štyri elektrinou poháňané dodávky prešli z Talianska do Číny, práve na výstavu Shanghai Expo [3]. Z histórie teda vyplýva, že najväčší progres zaznamenal práve vývoj protikolízneho systému, bez ktorého by automobily neboli schopné autonómneho pohybu. Na obrázku 2.1 je znázornený všeobecný vývoj bezpečnostných a asistenčných prvkov v automobiloch.



Obrázok 2.1 História produktov a poskytovaných služieb Collision Avoidance [4]

2.2 Protikolízne systémy automobiliek

Od roku 1998 kladú automobilky na protikolízne a varovné systémy (angl. *Collision Avoidance/Warning*, skr. CW/CA) 3 kritéria vid' [5]:

1. Výsledok varovania systému by mal minimálne rušiť pozornosť vodiča.
2. Systém automatického brzdenia by mal čo najmenej zasahovať do normálnych operácií počas jazdy.
3. Správne fungovanie systému za rôznych jazdných podmienok.

CA sa preto delí do dvoch hlavných kategórií:

- Pokročilé asistenčné systémy pre vodiča (angl. *Advanced Driver Assistance System*, skr. ADAS).
- Systémy predchádzania kolíziám (angl. *Precrash systems*).

ADAS zahŕňajú systémy ako rozpoznávanie značiek pri cestách (angl. *Traffic Sign Recognition*), upozorňovanie na prekroenie rýchlosti (angl. *Intelligent Speed Adaptation*, skr. ISA), vizuálne a zvukové varovanie pred hroziacou kolíziou v prípade prekážky na ceste (angl. *Obstacle and Collision Warning*), varovanie pri vybočení z jazdného pruhu (angl. *Lane Departure Warning*) a autonómny tempomat (angl. *Autonomous cruise control*, skr. ACC), ktorý bez zásahu vodiča spomalí vozidlo ak sa priblíži k inému vozidlu, prípadne zrýchli na požadovanú rýchlosť ak sa prekážke vodič vyhne [1].

Precrash systems operujú v dvoch fázach. Prvou je detekcia nebezpečenstva rôznymi technikami a druhou je značný zásah do riadenia, napríklad automatické brzdenie, keď vodič nedokáže včas zareagovať na varovanie [1]. Z tohto dôvodu automobilky používajú algoritmy, ktoré prevádzajú dva podstatné výpočty:

- Varovanie kritickej vzdialenosti (angl. *warning critical distance*) - je to taká vzdialenosť medzi vozidlami, keď je potrebné varovať vodiča, že sa pri aktuálnej rýchlosti nachádza príliš blízko vozidla nachádzajúceho sa pred ním.
- Kritickej vzdialenosti brzdenia (angl. *braking critical distance*) - je to vzdialenosť medzi vozidlami, pri ktorej je už potrebné využiť brzdový systém vozidla.

Výpočtové aparáty pracujú s relatívnou vzdialenosťou, relatívnou rýchlosťou a rýchlosťou vozidla, aby varovali vodiča alebo zasiahli do riadenia vozidla. Varovanie kritickej vzdialenosti je definované ako funkcia rýchlosti vozidla a relatívnej rýchlosti. Varovanie pre vodiča prichádza vo chvíli, keď vzdialenosť automobilu prekročí vzdialenosť, ktorá je pre kritickú vzdialenosť varovania definovaná (podľa aktuálnej rýchlosti). Kritická vzdialenosť brzdenia je definovaná podobne. Systém začne brzdiť vozidlo v prípade prekroenia kritickej hranice, resp. v poslednej

chvíli. Rozdiel, ktorý rozlišuje algoritmy jednotlivých automobiliek je možné nájsť v matematickom výpočte spomínaných vzdialeností. Každý výrobca alebo architekt takéhoto systému zahŕňa do svojho výpočtu vlastné konštanty a premenné, o ktorých si myslí, že najviac prispievajú k presnosti výpočtu.

2.2.1 Základný princíp

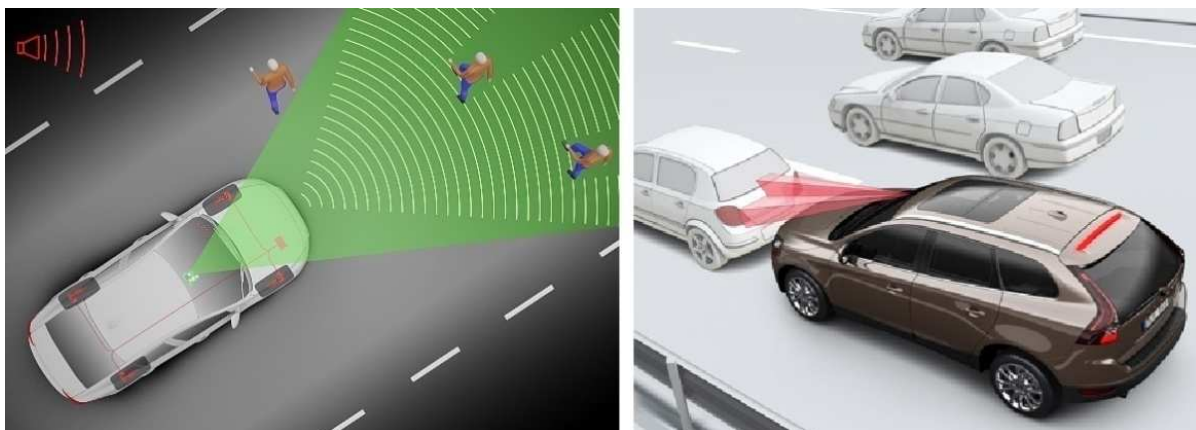
Základnou časťou celého protikolízneho systému sú senzory. Pokiaľ sa jedná o vysoké rýchlosti je potrebný radar s dlhším dosahom pre sledovanie premávky pred vozidlom. Na druhej strane u automobilky Volvo, ktorá prišla so systémom „City Safety“, postačí menej nákladný laserový diaľkomer (angl. *low cost laser radar*, skr. LIDAR). Tento protikolízny systém slúži na ochranu v malých rýchlostiach, napríklad v premávke v meste [2]. Takýto radar sa môže nachádzať v prednej časti vozidla, umiestnený buď v mriežke prednej masky automobilu alebo nad hlavným spätným zrkadlom ako je možné vidieť na obrázku 2.2.



Obrázok 2.2 Umiestnenie diaľkomerov v automobiloch [6] a) v prednej maske b) v hlavnom spätnom zrkadle

Pomocou radaru lokalizuje elektronika automobilu prekážky pred vozidlom a tak bráni či varovaním predchádza ich prípadnému stretu. Názornejšie vysvetlenie podáva ďalší obrázok 2.3, na ktorom je načrtnutá dopravná situácia, v ktorej sa tento protikolízny systém dá využiť.

V počiatkoch sa protikolízny systém automobilov sústredil na zrážky s väčšími objektmi, no následným vývojom sa dosiahla aj ochrana osôb, ktoré buď nedopatrením môžu skončiť v dráhe automobilu alebo ich skrátka vodič prehliadne.

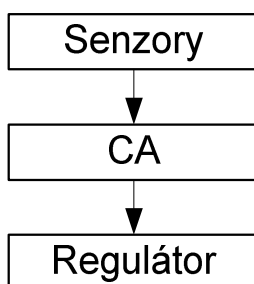


Obrázok 2.3 Princíp snímania objektov pred automobilom a) varovanie pred chodcami [6] b) bežná premávka - „Volvo City Safety“ [7]

2.2.2 Vlastné použite

Princípom akým fungujú jednotlivé protikolízne systémy sme sa aj my chceli inšpirovať a napodobniť takto funkčný celok, ktorý by nám pomohol ochrániť nášho mobilného robota pred možnými zrážkami. Ako bolo spomenuté, automobil dokázal včas varovať vodiča pred nebezpečným približovaním sa. Čo sa týka nášho robota, takisto ho ovláda človek, no rýchlosť robota sa bude upravovať podľa približujúcej sa prekážky. Čím bude objekt bližšie k robotovi tým bude ísť menšou rýchlosťou (v prípade veľkej vzdialenosti pôjde robot maximálnou rýchlosťou). Tým, že sa maximálna možná rýchlosť upravuje podľa najbližšej prekážky, nemusíme potom stanoviť, kedy sa prekážka nachádza v oblasti varovania a kedy v oblasti brzdenia.

Spomenuté *precrash systems* hovorili o dvoch častiach automobilového CA, tzv. pasívnej a aktívnej. Pasívna časť najprv vodiča varuje pred kolíziou a ak vodič patrične nezasiahne, až potom prichádza na rad aktívna časť, ktorá využije brzdomý systém. Po skúsenostiach s implementáciou aktívnej časti, ktorá je pre nás podstatnejšia, nebude ťažké pasívnu časť vytvoriť. Fungovanie systému je znázornené na obrázku 2.4. Časť nazvaná „Senzory“, poskytuje pomocou dostupného senzorického vybavenia potrebné hodnoty veličín. Tie sú potrebné pre časť „CA“ na to aby mohla patrične ovládať riadenie vydávaním pokynov do riadiacej časti „Regulátor“. Každá časť z obrázka 2.4 bude postupne riešená v tejto práci v ďalších kapitolách.



Obrázok 2.4 Časti protikolízneho systému

3 Senzorický systém robota

Senzorický systém je množina zariadení nainštalovaných na robotovi, ktoré poskytujú riadiacej jednotke bližšie informácie, podľa ktorých potom mobilný robot môže prevádzať potrebné výpočty prípadne sa orientovať v priestore. Podľa toho čo tieto senzory sledujú, ich delíme na externé, ktoré snímajú priestor v blízkosti robota a interné, ktoré poskytujú informácie o stave robota.

3.1 Externé senzory

Senzormi snímajúcimi okolie môžu byť napríklad GPS, kamerový systém, kompas, sonary, diaľkomery a všetky tie zariadenia, ktoré pomáhajú s orientáciou a videním robota. Sú v podstate očami a ušami robota.

3.1.1 Prehľad dostupných senzorov

Odvetvie robotiky sa pomerne rýchlo rozmáha a existuje mnoho typov senzorov. Mohli by sme ich rozdeliť do viacerých skupín, napríklad podľa vzdialenosti, ktorú dokáže snímať a tomu odpovedajúcich rozmerov a ceny. Avšak najpodstatnejšie je delenie podľa toho, akú fyzikálnu veličinu využívajú na svoje fungovanie. Najčastejšie je to svetlo, zvuk a magnetizmus. Ich stručný prehľad vid' [14].

Optické senzory

Princípom je vysielanie lúčov svetla a potom ich zachytávanie. Každý z nich je možné deliť ešte podľa počtu lúčov, ktoré vysieľa. Poznáme bodové, ale taktiež aj tie ktoré, vysielajú lúče v istom uhle. Dôležitým parametrom je dosah, ktorý vieme pomocou tohto senzora monitorovať.

Senzory s krátkym dosahom (niekoľko centimetrov) využívajú jednoduchý princíp infračerveného (angl. *infrared*, skr. IR) žiarenia. IR dióda vysieľa lúče svetla a snímač ich zachytáva. Tento princíp závisí na viacerých faktoroch prostredia, ktoré do značnej miery ovplyvňujú presnosť merania. Je to napríklad farba a povrch objektu, od ktorého sa svetlo má odraziť alebo uhol dopadu svetla či iné zdroje svetla (napríklad slnko či iné zdroje svetla).

Senzory so stredným dosahom, sú o niečo komplikovanejšie ako tie predchádzajúce. Svetlo je tak isto emitované, no odrazený lúč sa zameriava pohybujúcim sa objektívom na svetlo-citlivý pás. Faktory ovplyvňujúce presnosť merania sú podobné ako pri senzoroch s krátkym dosahom.

Pre väčší dosah sa využívajú senzory, ktorých princíp spočíva vo vysielaní laserových lúčov. Je to jediný spôsob monitorovania väčšieho dosahu. Meranie týmito senzormi môže byť ovplyvnené dymom či inými súčasťami vo vzduchu.

V neposlednom rade sa na rozpoznávanie prostredia využívajú aj kamery. Sú citlivé na IR žiarenie a zvyčajne je v ich objektívoch umiestnený IR filter. Lacnejšie z nich neobsahujú filter, ktorý robí kameru citlivú na slnečné svetlo. Taktiež sa môžeme stretnúť so stereo videním, ktoré využíva dve kamery umiestnené vedľa seba.

Zvukové senzory

Tento typ senzorov funguje na princípe vysielania ultrazvukových vĺn, ktoré potom zachytáva. Odmeriava čas, ktorý uplynul medzi vyslaním signálu a jeho prijatím.

Delia sa do troch skupín podľa frekvencie vysielaného zvuku. Prvou je infrazvuk, kde je frekvencia zvuku pod hranicou 20 Hz, druhou je ultrazvuk s rozmedzím 20 kHz – 2 MHz vo vzduchu alebo 30 MHz vo vode a tretou je obyčajný zvuk, ktorého frekvencia sa pohybuje medzi 20 Hz a 20 kHz.

Najčastejšie používaným senzorom je ultrazvukový senzor, ktorého frekvencia sa pohybuje pri hranici 40 kHz. Dosah jeho merania je maximálne 4 až 6 metrov a minimálne 2 až 5 cm. Tak ako doteraz uvedené optické senzory tak aj zvukové sú závislé na vlastnostiach odrazových plôch.

Ďalším typom sú magneticky založené senzory, ako je kompas. Aj ten je síce nainštalovaný na našom robotovi, ale nie je potrebný pre fungovanie protikolízneho systému.

Pre správne fungovanie nášho protikolízneho systému môžeme uvažovať snímanie okolia v dvoch odlišných významoch:

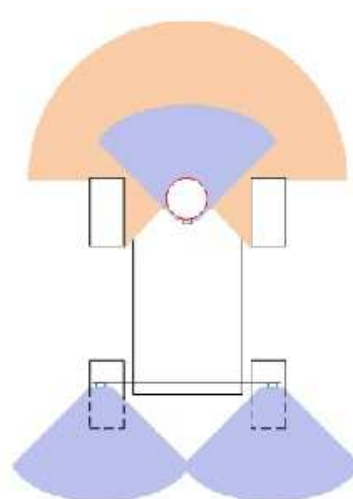
- sledovanie priestoru pre protikolízny systém (väčšia vzdialenosť, niekoľko metrov)
- bezprostredná vzdialenosť pred robotom

3.1.2 Použité senzory

Pri výbere zo spomínaných senzorov zohrala hlavnú rolu vzdialenosť, ktorú dokázali snímať a taktiež rozmery senzorov. Na obrázku 3.1 sú znázornené senzory, ktoré sme si v tejto súvislosti vybrali. Laserový diaľkomer (pre sledovanie väčších vzdialeností, viac ako 1 meter) umiestnený vpredu (označený tmavo-červenou farbou) a ultrazvukové snímače (pre kontrolu bezprostrednej vzdialenosti menšej ako 1 meter), jeden v prednej a dva v zadnej časti robota (označené tmavo-modrou farbou).

Laserový diaľkomer vracia hodnotu vo frekvencii 25 Hz (je možné nastaviť aj 50 Hz), čo znamená, že dáta prichádzajú každých 40 ms a zo sonarov, pri frekvencii 10 Hz, je oneskorenie dát 100 ms. Pri rýchlosti 100 km/h to znamená, že kým príde ďalší údaj z lidar, vozidlo prejde 1,1 m. Pre nášho robota, je to celkom dobrý výsledok, keďže sa dokáže pohybovať maximálne 15 km/h.

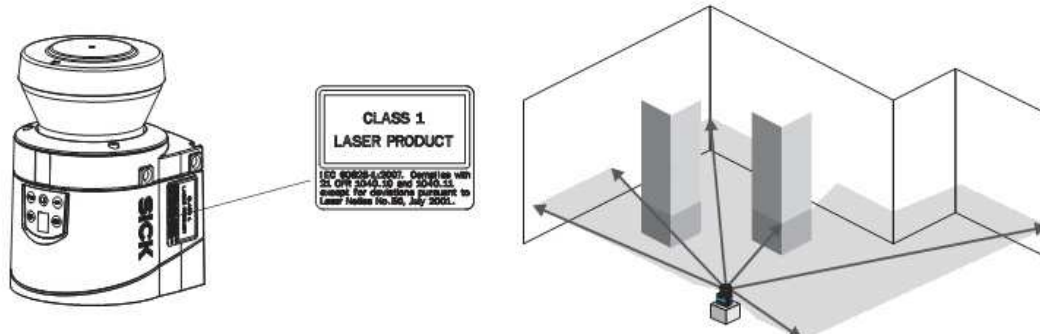
Z obrázka 3.1 sa možno zdá, že snímaná vzdialenosť sonarov je len o čosi menšia než vzdialenosť lidar. Avšak lidar ma dosah cca dvadsaťkrát väčší ako sonar. Obrázok slúži viac na znázornenie uhla snímania ako na vykreslenie veľkosti vzdialenosti, ktorú sníma.



Obrázok 3.1 Umiestenie senzorov

Laserový diaľkometer

Je umiestnený v prednej časti robota, pred prednou nápravou. Slúži na detekciu objektov pred robotom. Konkrétne ide o laserový diaľkometer typu SICK LMS 120 tzv. lidar, ktorý slúži predovšetkým na použitie v interiéroch.



Obrázok 3.2 a) Laserový diaľkometer SICK LMS 120 [6] b) ukážka monitorovania okolia [6]

Ide o elektro-optický laser, ktorý sníma okolitý priestor pomocou vysielaných laserových lúčov. Ak sa lúč odrazí od nejakej prekážky, lidar vráti vzdialenosť k tomuto bodu. Laserové lúče sú vysielané pomocou laserovej diódy a zachytávané pomocou fotodiódy. Vysoký vplyv na odraz má materiál a farba povrchu, od ktorého sa má lúč odraziť a takisto aj uhol, pod ktorým lúč dopadá na povrch. Všetky nastavenia, vlastnosti a riadenie vid' [8]. Niektoré z jeho parametrov sú uvedené v tabuľke 3.1.

Uhol snímania	270°
Uhol medzi emitovanými lúčmi	0,5°
Frekvencia snímania	25/50Hz
Komunikácia	Ethernet, RS-232, CAN
Max. sledovaná vzdialenosť	20 m
Využitie	Systémy ochrany objektov v interiéroch

Tabuľka 3.1 Parametre laserového dial'komera SICK LMS120 [8]

Sonary

Ďalším bezpečnostným prvkom, ktoré boli po čase nainštalované na robotovi sú ultrazvukové sonary. Pracujú na princípe vysielania, odrazu a zachytávania ultrazvukových vln.

Napríklad ak by stála pred robotom lavička v parku a systém by vypočítal, že je šírka medzi nohami lavičky ja dostačujúca, aby robot prešiel tak mu v tom nebude brániť. Čo ak je ale výška robota vyššia ako výška priestoru pod lavičkou? Presne preto je potrebný ultrazvukový sonar, ktorý bude sledovať bezprostrednú vzdialenosť, cca. 1 meter, pred robotom. Sonar, na rozdiel od lidar, ktorý sníma priestor v jednej rovine (dve osy), sníma celý priestor (v troch osiach), vysielaním do všetkých smerov, vie detekovať prítomnosť prekážky a v danej chvíli zareagovať.

3.2 Interné senzory

Napríklad, pre výpočet brzdnej dráhy potrebujeme poznať aktuálnu rýchlosť. Pre určenie smeru potrebujeme vedieť ako sú natočené kolesá. Všetky tieto informácie opisujú aktuálny stav robota a my ich potrebujeme poznať. Pri týchto senzoroach je potrebné zohľadniť frekvenciu príchodu nových dát. Rovnako existuje aj oneskorenie komunikácie hardwaru, ktoré je relatívne a pohybuje sa niekde okolo 2 ms.

3.2.1 Odometrický senzor

Odometria je proces, ktorý popisuje transformáciu dát poskytnutých enkodérmi na zmenu pozície a orientácie robota [15]. Základom odometrie je geometrický tvar robota a z neho príslušné výpočty, no my sme sa zamerali iba na meranie jednej veličiny a tou je rýchlosť. Budeme ju potrebovať pre výpočet brzdnej dráhy vozidla. Frekvencia príchodu hodnôt je 10 Hz, to znamená, že každých 100 ms sa vráti nová hodnota.

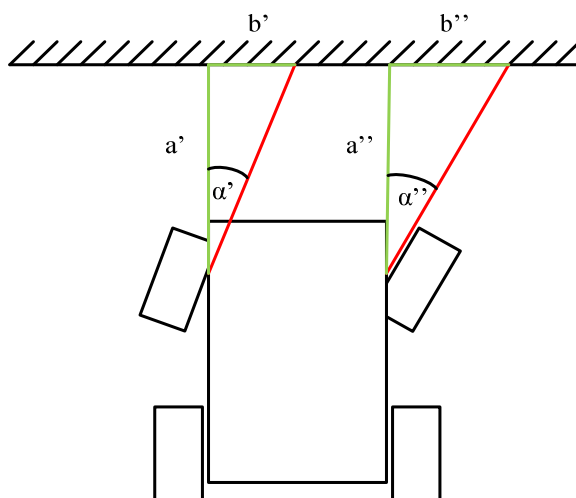
Ako odometrický senzor sme použili analógový myš s rozhraním RS232c. Pomocou adaptéra na USB sme ho pripojili k PC. Princíp je jednoduchý: Fotodióda neustále svieti na koliesko

s pravidelne umiestnenými otvormi, ktoré sa točí v závislosti na pohybe myšou. Prerušované dopadanie svetla na fototranzistor sa spracúva v integrovaných obvodoch a ukladá sa v tzv. tikoch, ktorých hodnotu vysiela myš v časových intervaloch sériovým rozhraním. Koliesko sme nainštalovali na hnací hriadeľ kolies. Vytvorili sme triedu a metódy, ktoré z prichádzajúcich dát zistia počet tikov. Do implementácie sme zahrnuli aj prepočet, keďže údaje z myši priamo neodpovedajú rýchlosti robota. Myš nám vracia počet tikov, ktoré odpovedajú spomínanému prerušovanému svetlu. Prepočet bol triviálny, keďže stačilo iba premerať vzdialenosť a počet tikov, ktorý vrátila myš za túto dráhu.

3.2.2 Natočenie kolies

Na to aby sme vedeli predpovedať trajektóriu robota, potrebujeme prijímať informácie o natočení kolies prednej nápravy. Ovládanie kolies preberajú servomotory, ktoré nám ale nevedia povedať o koľko bolo ktoré koleso otočené. Preto musíme vytvoriť systém, ktorý by nám túto informáciu spoľahlivo podal.

Kolesá sa nachádzajú v zábere laserového diaľkomera, takže ich snímanie diaľkomerom je možné. Postavili sme preto robota kolmo pred prekážku v istej vzdialenosti. Kolesá, na ktorých sme mali provizórne pripevnené laserové ukazovadlo, sme postupne natáčali do viacerých polôh na obe strany. Mali sme k dispozícii dve dĺžky pravouhlého trojuholníka, teda nebol problém vypočítať príslušný uhol každého kolesa a potom vypočítať priemer uhlov oboch kolies.



Obrázok 3.3 Výpočet natočenia kolies

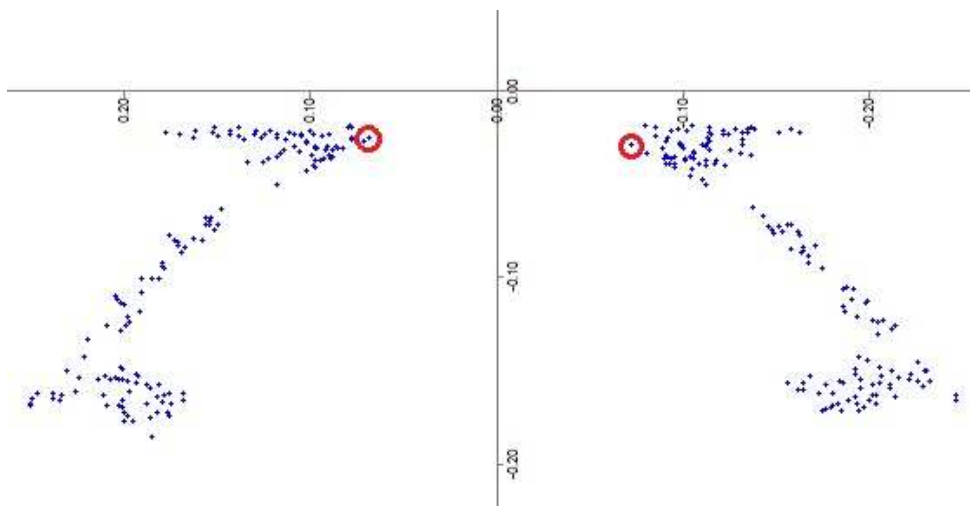
Na obrázku 3.3 je možné spozorovať, že kolesá nie sú natočené o rovnaký uhol. Nie je to chyba, je to zámerne, aby bolo vidieť princíp Ackermanovho podvozku (vysvetlenie viď kapitola 4), aký používa robot. Jednotlivé uhly sa vypočítali pomocou tangensovej vety.

$$\tan \alpha' = \frac{b'}{a'} \Rightarrow \alpha' = \tan^{-1} \frac{b'}{a'} \quad (1)$$

$$\tan \alpha'' = \frac{b''}{a''} \Rightarrow \alpha'' = \tan^{-1} \frac{b''}{a''} \quad (2)$$

$$\alpha = \frac{\alpha' + \alpha''}{2} \quad (3)$$

Pri každom natočení sme si urobili záznam z laserového diaľkomera, v ktorom sa nám uchovali vzdialenosti bodov, pre každé natočenie kolies. Pre každý záznam sme sa snažili nájsť referenčný bod (ďalej RB), podľa ktorého by sme potom spätne rozoznávali natočenie kolies. RB je označený červeným krúžkom v obrázku 3.4.



Obrázok 3.4 Body kolies

Kolesá sme mali priamo a potom sme ich natáčali do niekoľkých polôh na každej strane. V každej polohe sme si odčítali záznam bodov ako je napríklad na obrázku 3.4.

Pre každý záznam sme si našli RB pravého (RBP) a ľavého kolesa (RBL). Išlo nám o vzdialenosti od osi y, ktoré sme od seba odpočítavali. V tabuľke 3.2 vidíme akú funkciu plnil rozdiel RBP a RBL.

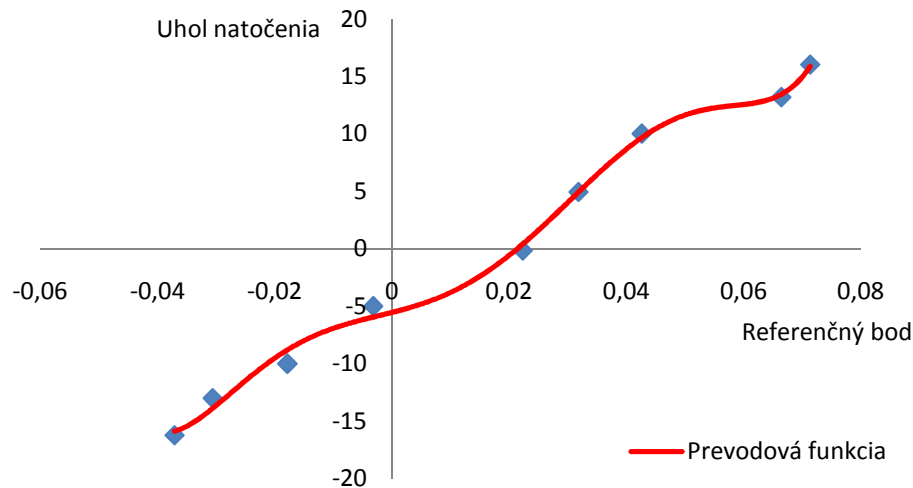
	Význam
RBP - RBL > 0	Kolesá sú natočené vpravo, čím väčší rozdiel tým sú viac natočené
RBP - RBL < 0	Kolesá sú natočené vľavo, čím menší tým sú viac natočené
RBP - RBL = 0	Kolesá sú v priamom smere

Tabuľka 3.2 Rozdiel vzdialeností RBP a RBL

V tomto stave sme poznali vypočítaný uhol natočených kolies a zároveň aj RB, zo záznamov prislúchajúcich daným natočeniam. Rozhodujúci bol práve rozdiel týchto RB ako je uvedený

v tabuľke 3.2. Každý takýto výsledok, ktorý odpovedal rozdielu RBP a RBL, je potrebný ako celkový RB pri danom natočení.

Tieto dvojice bodov (uhly natočenia a im prislúchajúce RB) sme si potom vyniesli do súradnicového systému. Vzniknuté body sme potom preložili polynómom 6-tého stupňa, aby sme dostali rovnicu funkcie, ktorá zodpovedá prevodu RB na uhol natočenia.



Obrázok 3.5 Graf prevodovej funkcie

Počas jazdy robotom, nám laserový diaľkomer vráti body patriace kolesám, protikolízny systém z nich vypočíta RB. Výsledok sa použije v prevodovej funkcii a jej funkčná hodnota bude uhol, pod ktorým sú kolesá natočené. Prevodová rovnica, ktorú sme takto dostali je

$$y = 1 \cdot 10^9 x^6 - 1 \cdot 10^8 x^5 - 2 \cdot 10^6 x^4 + 27683 x^3 + 1986 x^2 + 129,4 x - 5,547 \quad (4)$$

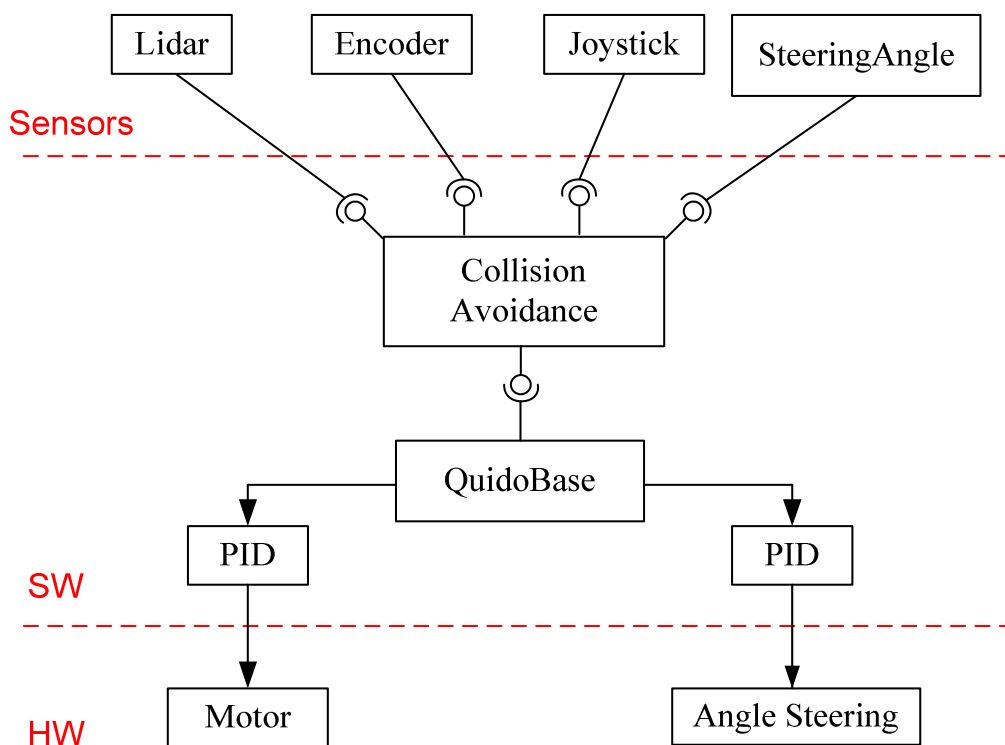
kde hodnota x je vypočítaný RB (z hodnôt z laserového diaľkomera) a y je veľkosť uhla, pod ktorým sú natočené kolesá.

4 Protikolízny systém

V tejto kapitole si popíšeme konkrétny návrh riešenia protikolízneho systému pre mobilného robota. Ak vytvárame nejaký systém je dobré si ho znázorniť diagramom. V našom prípade sme nepostupovali inak. Načrtli sme si ako by mal náš systém fungovať a potom sme postupne riešili fakty ako k danej funkčnosti dôjsť.

Riadenie protikolízneho systému sa skladá z viacerých komunikujúcich súčastí. Na obrázku 4.1 je vidieť rozdelenie jednotlivých komponent do troch vrstiev: sensory, software a hardware. Časť nazvaná „QuidoBase“ riadi výkon motora, natočenie kolies pomocou servomotorov a ďalšie akcie týkajúce sa chodu robota. Na to, aby mohol byť robot chránený pred kolíziami, potrebuje riadiaca jednotka vedieť, kedy a akým spôsobom má ovplyvniť chod robota. Na to jej poslúži časť „CollisionAvoidance“. Z obrázka možno vyčítať, že sleduje senzory nainštalované na robotovi, prevádza výpočty nad dostupnými hodnotami a predáva signály hlavnej riadiacej jednotke „QuidoBase“. Funkcie všetkých blokov sú popísané v tabuľke 4.1 pod obrázkom.

Zmeny v riadení (napr. rýchlosť, brzdenie, natočenie kolies) musia prebiehať čo najrýchlejšie, ale takisto aj čo najplynulejšie. K tomu systém využije softwarovú náhradu PID regulátora.



Obrázok 4.1 Model dátového toku

Názov bloku	Funkcia
Joystick	Všetky riadiace signály z diaľkového ovládania
Lidar	Poskytuje informácie o vzdialenostiach prekážok pred robotom
Encoder	Vracia odometrické dáta, rýchlosť, vzdialenosť
SteeringAngle	Podáva aktuálne informácie o natočení kolies
CollisionAvoidance	Zber všetkých dát a výpočty pre zásah do riadenia
QuidoBase	Základná riadiaca jednotka robota
PID	Softwarová náhrada PID regulátora na ovládanie robota
Motor	Motor poháňajúci robota
AngleSteering	Servomotory pre natáčanie kolies

Tabuľka 4.1 Časti modelu dátového toku

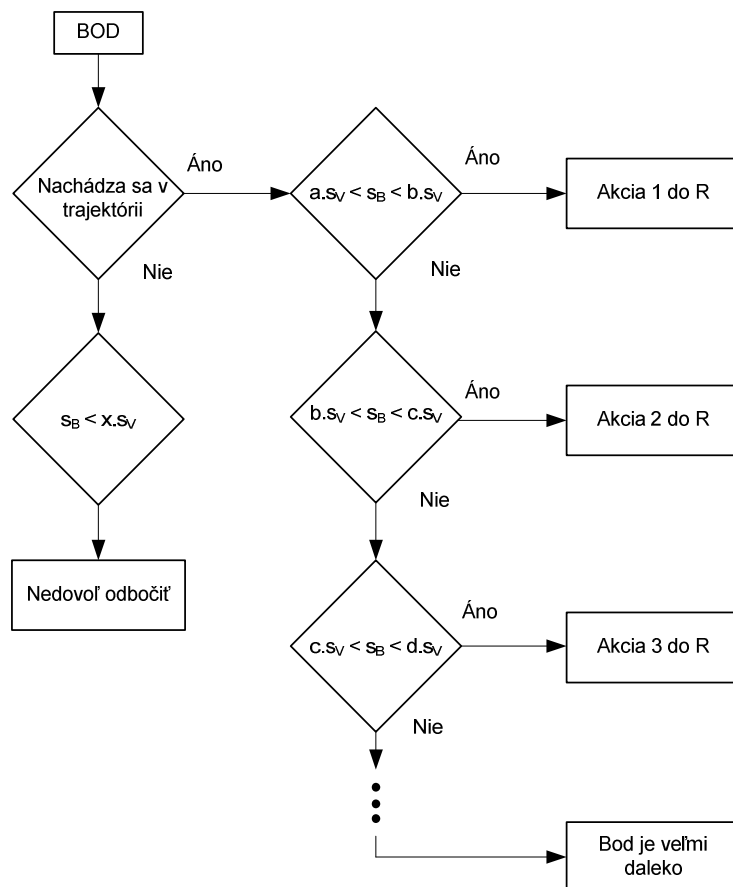
4.1 Vlastný návrh

Náš protikolízny systém pracuje pomerne jednoducho. Zohľadňuje trajektóriu robota, po ktorej by mal ísť podľa aktuálneho natočenia kolies a brzdnú dráhu, ktorú potrebuje aby bezpečne zastavil pred detekovanou prekážkou. Berie do úvahy aj prejazd popri prekážke, ktorá je blízko. Ak sa nenachádza v smere robota, potom tento systém musí dovoliť bezpečný prejazd popri prekážke a nesmie dovoliť odbočiť tak, aby narazil.

CA dostáva informácie o objektoch, presnejšie o ich polohe pred robotom od laserového diaľkomeru v podobe vzdialenosti, ktorú si sama prevádza na súradnice bodu. Podľa nasledujúceho postupu sa potom rozhoduje pre ďalšiu akciu, či prípadný zásah do riadenia:

1. Zoberieme bod a podľa jeho polohy určíme, či sa nachádza v trajektórii robota.
 - a) Ak áno, zaradíme bod do tzv. zóny podľa toho ako ďaleko sa od robota nachádza. V závislosti na zóne sa potom nastaví akcia, napríklad varovanie alebo aktívny zásah do riadenia robota.
 - b) Ak nie, porovnáme jeho vzdialenosť s brzdou dráhou, ktorá je potrebná na zastavenie pri aktuálnej rýchlosti. Ak je vzdialenosť bodu väčšia tak dovoľíme robotu odbočiť. Ak nie, a hrozí zrážka, potom zasiahneme do riadenia tak, že nedovoľíme odbočenie týmto smerom.

Pre lepšie pochopenie je na obrázku 4.2 znázornený stavový diagram, podľa ktorého bude CA pracovať. Počet zón, do ktorých sa body zadelia môže byť ľubovoľný. Záleží len na nás, koľko stupňov ochrany si zvolíme. V tabuľke 4.2 pod obrázkom sú vysvetlené všetky premenné, ktoré sú uvedené v diagrame.



Obrázok 4.2 Stavový diagram protikolízneho systému

Premenné	Význam
a, b, c, d	Koeficienty zón - koeficienty vzdialenosti zón od robota
X	Koeficient tolerancie vzdialenosti bodu mimo trajektórie
R	Regulátor
S_V	Vypočítaná brzdná dráha podľa aktuálnej rýchlosti
S_B	Vzdialenosť bodu podľa laserového diaľkomeru, prípadne podľa výpočtu kružnicového oblúku

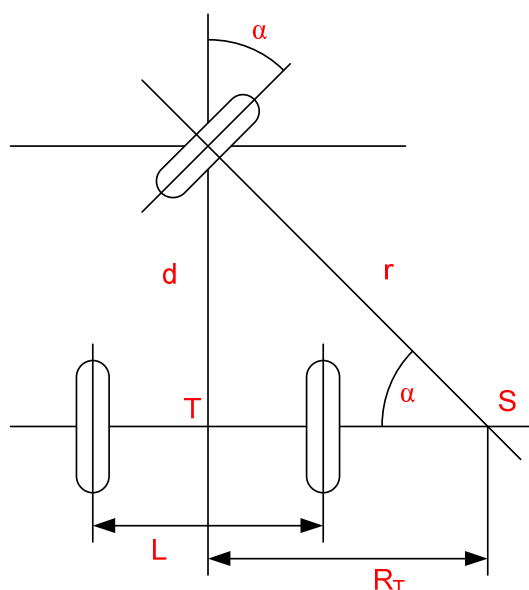
Tabuľka 4.2 Vysvetlivky premenných

4.2 Kinematický model robota

Smer, akým sa bude robot pohybovať budeme určovať podľa natočenia kolies prednej nápravy. Ak sú kolesá v nulovom uhle, znamená to, že robot pôjde po priamke. Ak je uhol natočenia nenulový potom vieme, že trajektória bude v tvare kružnice. Ak poznáme smer, potom je jednoduché určiť, či sa bod nachádza v dráhe robota alebo nie.

Riadenie prednej nápravy, je typu tzv. Ackermannovho podvozku [9]. Ten je charakteristický tým, že sa kolesá prednej nápravy nevychyľujú o rovnaký uhol. Je to z dôvodu, že kružnica, ktorú opisuje ľavé koleso ma väčší polomer ako tá ktorú opisuje pravé koleso, samozrejme ak sú kolesá natočené doprava. V opačnom smere by veľkosti polomerov boli opačné. Tento princíp natáčania kolies bol zavedený z dôvodu prehrievania a ničenia pneumatík tým, že sa jedno z kolies dostávalo do šmyku.

Pre ľahšie počítanie sme si štvorkolesový podvozok zjednodušili na trojkolesový s tým, že riadiace koleso prednej nápravy sa nachádza v strede prednej nápravy a uhol o ktorý je natočené je vypočítaný ako priemer uhlov natočenia pravého a ľavého kolesa. Obrázok 4.3 znázorňuje celú geometriu robota.



Obrázok 4.3 Geometrický model trojkolesového mobilného robota

Systém CA bude fungovať pomocou súradnicového systému. Preto ak zoberieme za počiatok súradnicovej sústavy bod, ktorý je stredom prednej nápravy, tak:

- α - uhol natočenia kolies - vpravo - kladný, vľavo - záporný
- d - rázvor medzi prednou a zadnou nápravou
- r - polomer otáčania

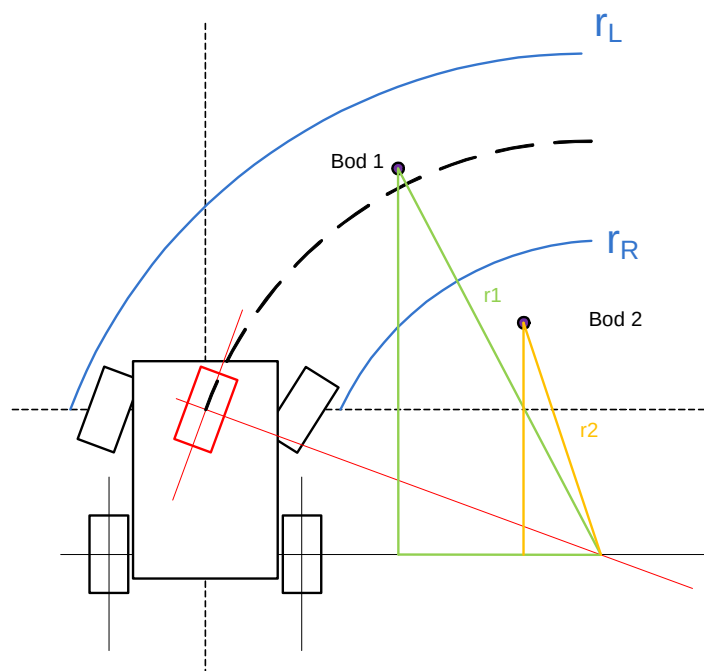
- T - stred zadnej nápravy
- L - rázvor zadnej nápravy, resp. šírka robota
- S - stred otáčania kolies
- R_T - vzdialenosť medzi S a T

Vzťahy pre prepočet neznámych veličín sú:

$$r \sin \alpha = d \sin(90^\circ) \quad (5)$$

$$R_T \sin \alpha = d \sin(90^\circ - \alpha) \quad (6)$$

V predchádzajúcej kapitole sme uviedli, že potrebujeme vedieť trajektóriu robota, aby sme vedeli detekovať prekážky, ktoré mu stoja v ceste. O pár riadkov vyššie sme spomenuli, že budeme uvažovať nad súradnicovou sústavou. Na obrázku 4.4 je znázornený pohľad na vozidlo a jeho dráhu, pri danom natočení kolies.



Obrázok 4.4 Model výpočtu dráhy mobilného robota

Na obrázku je uvedený jeden bod, ktorý sa nachádza v trajektórii robota a jeden, ktorý je mimo jeho dráhu. Premenné r_L a r_R vyznačujú skutočnú dráhu, ktorá započítava aj rozmery MR. Sú to v podstate hodnoty polomerov kružníc s tým istým stredom otáčania ako pri myslenom prostrednom kolese, ktoré sú označené podľa strany, na ktorej sa nachádzajú. Teda r_L označuje polomer ľavej

strany a r_R polomer strany pravej. Tieto polomery sa menia podľa smeru robota. Ak je smer mobilného robota opačný ako na obrázku, t. j. ide do ľavej strany potom sa polomery vymenia. Pre ľahšie určenie polomerov nám budú slúžiť nasledujúce vzťahy:

- Pre uhol $\alpha > 0$, teda smer vpravo:

$$r_L = r + \frac{L}{2} + d_{min} \quad (7)$$

$$r_R = r - \frac{L}{2} - d_{min} \quad (8)$$

- Pre uhol $\alpha < 0$, teda smer vľavo:

$$r_L = r - \frac{L}{2} - d_{min} \quad (9)$$

$$r_R = r + \frac{L}{2} + d_{min} \quad (10)$$

Hodnota d_{min} je odchýlka medzi prekážkou a robotom a zároveň sa do nej môže zahrnúť aj tolerancia zadnej nápravy, ktorá opisuje o čosi menší polomer ako predná.

Pre uhol $\alpha = 0$, čo je priamy smer, nie je potrebné vypočítavať polomery, pretože nám vzniknú iba hranice, ktoré sú rovnobežné so smerom robota. Preto počítame len so šírkou L a odchýlkou d_{min} . Všetky tieto výpočty potom prevádzame do súradnicovej sústavy, kde počítame s konkrétnymi súradnicami jednotlivých bodov. Laserový diaľkomer nám poskytne vzdialenosť bodov pred MR. Po ich prepočítaní na súradnice potom môžeme posúdiť či sa v dráhe MR nachádzajú alebo nie.

Na predchádzajúcom obrázku sú zakreslené príklady bodov v priestore pred MR. Ako bude vedieť MR, že mu stoja v ceste? Jednoducho. Ak máme vzdialenosť každého bodu, potom podľa parametrov laserového diaľkomeru vieme aj určiť, pod akým uhlom sa daný bod nachádza a teda vieme vypočítať jeho súradnice. Musíme však brať do úvahy pozíciu laserového diaľkomera, ktorý nie je umiestnený v strede prednej nápravy, ale je vysunutý o niekoľko centimetrov dopredu. Postačí nám sínusová a kosínusová veta:

$$\text{súradnica } x(y) = \sin \alpha \cdot \text{vzdialenosť bodu} \quad (11)$$

$$\text{súradnica } y(x) = \cos \alpha \cdot \text{vzdialenosť bodu} \quad (12)$$

kde veľkosť uhla α vypočítame:

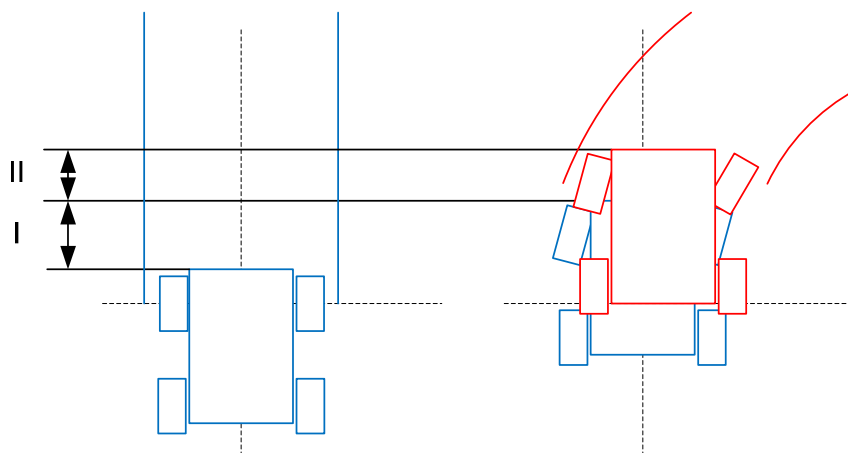
$$\alpha = i \frac{\text{rozsah snímania lidar}}{\text{počet emitovaných}} \quad (13)$$

prítom i je index emitovaného lúča, prislúchajúceho hodnote vzdialenosti toho bodu, pre ktorý chceme súradnice vypočítať. Nesmieme pri tom zabúdať v akom kvadrante sa bod nachádza (preto sú

vo vzorcoch (11) a (12) v zátvorkách uvedené opačné súradnice) a podľa toho upraviť znamienka a správne vypočítať uhol, ktorý odpovedá danému trojuholníkovému vyjadreniu.

Podľa súradníc bodu, ktoré nám poskytne laserový diaľkomer určíme veľkosť polomeru, ktorá je rovná vzdialenosti medzi bodom a stredom otáčania kolies. Ak sa bod nachádza v trajektórii robota, potom sa bude hodnota polomeru tohto bodu pohybovať medzi hranicami trajektórie, teda r_L a r_R .

Kým sa robot dostane do smeru aký od neho požaduje ovládanie, potrvá to istú dobu a zároveň pri istej rýchlosti aj nejakú dráhu. Tieto odchýlky a oneskorenia musia byť vo výpočte zahrnuté. Na obrázku 4.5 sú znázornené dve zóny, ktorými robot prejde než naberie požadovaný smer.



Obrázok 4.5 Oneskorenie riadenia

Zóna I zobrazuje dráhu, ktorú robot prejde vplyvom oneskorenia komunikácie. Ak sa robot pohybuje rýchlosťou v , doba medzi požiadavkou na natočenie kolies a skutočným natočením je t_0 , potom dráha ktorú prejde v zóne I sa rovná $v \cdot t_0$. Zóna II zaznamená už natočenie do strán, ale tým, že servomotory nedokážu otočiť kolesá v jednom momente na požadovaný smer, sa robot bude pohybovať po krivke s meniacim sa polomerom. Až po prejení týchto dvoch zón je možné smer robota pokladať sa požadovaný.

4.3 Brzdná dráha

Pokiaľ ide o protikolízny systém, je dôležité vedieť, že jedným z podstatných veličín vo výpočtoch zohráva vzdialenosť medzi robotom a objektom, ktorý sa nachádza v priestore pred ním. Pre príklad uvediem výpočty automobiliek Mazda a Honda vid' [5].

Algoritmus Mazda pre výpočet kritickej vzdialenosti brzdenia používa tento vzorec:

$$d_{br} = \frac{1}{2} \left(\frac{v^2}{\alpha_1} - \frac{(v - v_{rel})^2}{a_2} \right) + v\tau_1 + v_{rel}\tau_2 + d_0 \quad (14)$$

kde v je CW/CA rýchlosť vozidla, v_{rel} je relatívna rýchlosť medzi vozidlami ($v_{rel} = v - v_{predchádzajúceho}$), α_1 je maximálne brzdné zrýchlenie, α_2 je maximálne brzdné zrýchlenie predchádzajúceho vozidla, τ_1 a τ_2 sú oneskorenia reakcií vodiča a aplikácie výpočtu systému a d_0 je minimálna vzdialenosť medzi dvomi vozidlami.

Algoritmus Honda využíva pre výpočet, kedy má systém vodiča varovať o nebezpečnej blízkosti, definíciu:

$$d_w = 2,2v_{rel} + 6,2 \quad (15)$$

kde je v_{rel} relatívna vzdialenosť medzi vozidlami. Definícia pre výpočet kritickej vzdialenosti brzdenia je:

$$d_{br} = \begin{cases} \tau_1 v_{rel} + \tau_1 \tau_2 \alpha_1 - 0,5 \alpha_1 \tau_1^2 \frac{v_2}{\alpha_2} \geq \tau_2 \\ \tau_2 v - 0,5 \alpha_1 (\tau_2 - \tau_1)^2 - \frac{v_2^2}{2 \alpha_2} \frac{v_2}{\alpha_2} < \tau_2 \end{cases} \quad (16)$$

kde konštanty v , v_{rel} sú konštanty ako u predchádzajúcej automobilky. Hodnota α zobrazuje maximálne brzdné zrýchlenie a v je rýchlosť. Dolný index 1 označuje naše vozidlo a index 2 vozidlo, nachádzajúce sa pred naším vozidlom, τ_1 udáva oneskorenie systému a τ_2 čas brzdenia.

Výpočty každého výrobcu by sa dali zovšeobecniť do dvoch rovníc:

$$d_{w,scaled} = d_w f(\mu) g(driver) \quad (17)$$

$$d_{br,scaled} = d_{br} f(\mu) g(driver) \quad (18)$$

kde je vidieť, že výpočet vzdialenosti, varovania/brzdenia, sa skladá z koeficientu pre prípad varovania/brzdenia a potom funkcie vypočítavajúcej vplyv trenia vplyvom koeficientov šmykového trenia pre daný typ vozovky a funkcie zahrňujúcej ľudský faktor pri riadení vozidla.

Je dôležité si uvedomiť rozdiel medzi dvomi typmi vzdialenosti, s ktorými sa tu stretneme.

1. Vzdialenosť nameraná - skutočná vzdialenosť medzi robotom a objektom. Mení sa v čase. Jej hodnotu získame meraním, pomocou laserového diaľkomera umiestneného v prednej časti robota.
2. Vzdialenosť vypočítaná - brzdná dráha - je to vzdialenosť, ktorá je potrebná pre zastavenie robota pri momentálnej rýchlosti a aktuálnych podmienkach prostredia. Jej hodnotu prepočítavame podľa vzorca:

$$s = vt_0 + \frac{v^2}{2a} + d_{min} \quad (19)$$

Tieto dve hodnoty sa prehodnocujú počas celého behu CA systému. Vzájomne sa porovnávajú, aj s nami určenými toleranciami a podľa výsledkov porovnania sa vykonávajú zásahy do riadenia.

Na vzorci (19) je zrejmé, že sa skladá z troch častí: dráha, ktorú prejde robot kým sa zmena prejaví na pohybe robota od jej výpočtu, samotný výpočet brzdné dráhy podľa aktuálnej rýchlosti a vzdialenosť, s ktorou má robot pred prekážkou počítať.

V tomto výpočte je viac neznámych hodnôt resp. premenných, ktoré sú pre výpočet potrebné. Hodnotu v , nám poskytne odometria. Čas t_0 získame postupným skúšaním. Brzdné zrýchlenie sme určili pomocou akcelerometra skúšobnými jazdami (je určené konštantou v konfiguračnom súbore, viď kapitola 6) a odchýlka je ľubovoľná, no ak tam nebude, resp. bude rovná 0, potom riskujeme, že ak výpočet nebude úplne presný, hrozí zrážka.

Mobilný robot dokáže pomocou laserového diaľkomeru určiť vzdialenosť prekážky od robota, ktorú môžeme porovnávať s vypočítanou brzdou dráhou a tak robiť príslušne zásahy do riadenia robota.

5 Regulácia a riadenie

Náš robot je primárne riadený diaľkovým ovládaním typu joypad od firmy Sony. Ide o hybrid medzi gamepadom a joystickom, keďže ovládanie má tvar gamepadu, ale sú na ňom aj malé páčky, ktoré plnia úlohu malého joysticku. A práve tie používame na riadenie smeru vozidla, teda natočenie kolies a na reguláciu plynu.

Riadenie rýchlosti robota, t. j. ovládanie vpred a vzad bolo upravené. Doteraz sa pomocou páčky na joypade riadil výkon motora a nie jeho rýchlosť. To znamená, že ak bola páčka v jednej pozícii, tak bol požadovaný príslušný výkon motora bez ohľadu na podmienky okolia. V prípade, že sme išli s robotom do kopca a páčka ovládania bola v tej istej pozícii, výkon motora sa nezmenil, ale jeho rýchlosť klesla.

V súčasnosti sa páčkou pre pohyb vpred a vzad ovláda rýchlosť. Robot plynule reguluje výkon motora, aby dodržal požadovanú rýchlosť. Ak si zoberieme podobný príklad ako v predchádzajúcom odseku, tak ak robot pôjde hore kopcom a jeho rýchlosť klesne, automaticky regulátor upraví výkon motora, tak aby sa robot pohyboval rýchlosťou požadovanou z diaľkového ovládania.

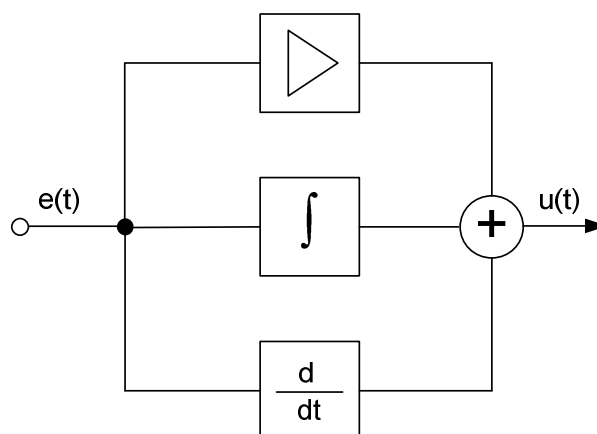
V tomto prípade prichádzame k momentu, keď poznáme veličiny namerané, napríklad senzormi a veličiny požadované, buď ovládaním alebo výpočtom riadiacej jednotky. Týmto nám vzniká sústava, ktorej sa hovorí regulačná. Jej úlohou je v čo najkratšom čase dosiahnuť aby sa aktuálna hodnota istej veličiny zmenila na požadovanú. K tomu je potrebný regulátor, ktorý túto zmenu prevedie. V našom prípade sme použili tzv. PID regulátor viď [10] a [11].

5.1 PID regulátor

PID regulátor je známy v regulačnej technike. Regulátor ako taký sa stará o regulovanie regulačnej sústavy bez zásahu človeka. V našom prípade ho využijeme na reguláciu rýchlosti a natočenia kolies prednej nápravy.

Tým, že diaľkovým ovládaním regulujeme požadovanú rýchlosť a nie priamo výkon motora, musíme tento výkon regulovať podľa požadovanej rýchlosti. Máme preto aktuálnu hodnotu výkonu a potom požadovanú, ktorú potrebujeme pre zaistenie alebo udržanie rýchlosti (ak ide o cestu do kopca).

Podobne je to aj pri natáčaní kolies. Pokiaľ z ovládania príde pokyn na zmenu smeru robota, ide nám o čo najrýchlejšie natočenie kolies do požadovanej polohy. Samozrejme toto natočenie je najprv overené samotným protikolíznym systémom a následne ak je povolené nám vznikajú opäť dve veličiny - aktuálny uhol natočenia a požadovaný. V oboch prípadoch nám vzniká sústava, ktorú potrebujeme vhodne regulovať. Použitý PID regulátor sa skladá z viacerých zložiek a jeho vnútorná schéma je vyznačená na obrázku 5.1.



Obrázok 5.1 Schéma PID regulátora

Povedzme, že si výpočet CA vyžiada úpravu rýchlosti. Potom budeme mať rýchlosť aktuálnu $y(t)$ a požadovanú $w(t)$. Medzi týmito hodnotami vzniká regulačná odchýlka $e(t)$ pomocou vzťahu:

$$e(t) = w(t) - y(t) \quad (20)$$

V závislosti na $e(t)$, ktorá je privádzaná na vstup PID regulátora sa mení akčná veličina, ktorou potom zasahujeme do systému. Z obrázku môžeme vyčítať, že PID regulátor obsahuje tri zložky, ktoré rôznymi spôsobmi ovplyvňujú výstupnú akčnú veličinu.

1. **P - proporcionálna zložka** - proporcionálny zásah je úmerný regulačnej odchýlke podľa vzťahu:

$$u(t) = Ke(t) \quad (21)$$

K - zosilnenie, ktoré riadi zložku P

Ak by sme použili iba proporcionálne chovanie sa regulátora bez ostatných dvoch zložiek, vznikla by nám trvalá regulačná odchýlka. Zväčšením zosilnenia K , by sme mohli zmenšiť trvalú regulačnú odchýlku, ale vzniká tak nebezpečenstvo nestabilného regulačného obvodu, kde môže regulovaná veličina kmitavo alebo nekmitavo neobmedzene narásť a tak poškodiť zariadenie. Aby sme tomu predišli pridáva sa do tohto regulátora aj integračná zložka.

2. **I - integračná zložka** - integračný zásah je úmerný dobe, počas ktorej existuje regulačná odchýlka:

$$u(t) = u_0 + \frac{1}{T_I} \int_0^t e(\tau) d\tau \quad (22)$$

T_I - integračná časová konštanta, ktorá riadi zložku I

Použitím integračnej zložky zmizla trvalá regulačná odchýlka, pretože regulátor mení akčnú veličinu neustále kým nenájde stav nulovej regulačnej odchýlky. Zmenšovaním T_I rastie podiel integračnej zložky a kmitavosť rastie taktiež. Je možné ho ovplyvniť pridaním ďalšej a to derivačnej zložky regulátora.

3. **D - derivačná zložka** - spôsobuje, že akčný zásah bude úmerný rýchlosti zmeny regulačnej odchýlky podľa vzťahu

$$u(t) = T_D \frac{de(t)}{dt} \quad (23)$$

T_D - derivačná časová konštanta, ktorá riadi zložku D

Táto zložka slúži na kompenzáciu zmien regulovanej veličiny v predstihu, teda napomáha tlmeniu zákmitov regulačného obvodu. Princíp je v tom, že hneď ako sa začne hodnota veličiny meniť k novej hodnote, derivačná zložka začne pôsobiť v opačnom smere teda od žiadanej hodnoty.

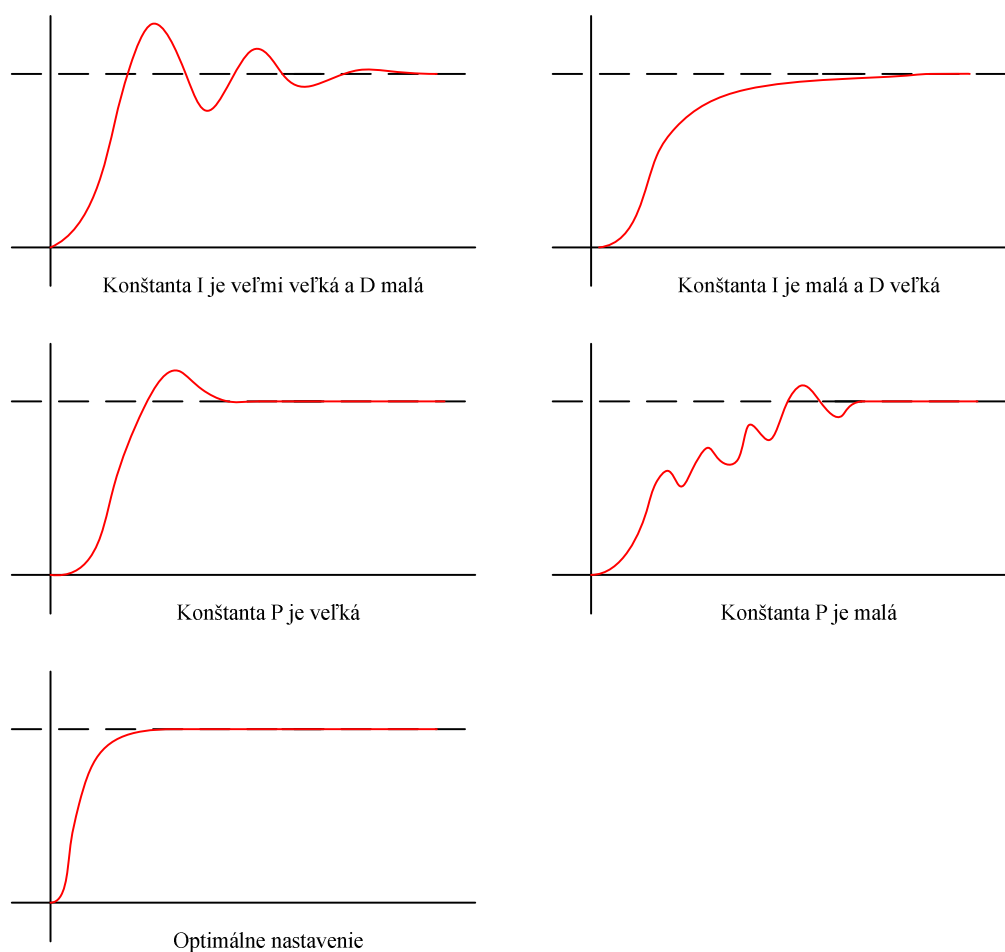
Zložky P, I, a D sa sčítajú v celkovú akčnú veličinu a spoločne tak zasahujú a riadia priebeh regulačného pochodu. Celkovo môžeme pomocou konštánt P, I a D zapísať vzťah pre PID regulátor takto:

$$u(t) = Pe(t) + I \int_0^t e(t)dt + D \frac{de(t)}{dt} \quad (24)$$

Nastaviť optimálne hodnoty konštánt pri jednotlivých zložkách je pomerne zložité. Aby sme aspoň približne vedeli, či sú jednotlivé zložky nastavené správne poslúžia nám tieto pomocné grafy na obrázku 5.2. Ako bolo na začiatku spomínané, PID regulátor sme využili v dvoch rolách. Hodnoty jednotlivých zložiek P, I a D sú uvedené v tabuľke 5.1.

Použitie / Zložka	P	I	D
Rýchlosť	4,00	1,00	2,50
Natočenie kolies	0,50	0,25	0,02

Tabuľka 5.1 Použité konštanty PID regulátora



Obrázok 5.2 Optimalizácia konštánt P, I a D

5.2 Riadenie rýchlosti

V prípade regulácie rýchlosti budeme uvažovať nad tromi stavmi. Prvým bude aktuálna rýchlosť, vrátená odometrickým senzorom, druhým bude rýchlosť požadovaná a tretím rýchlosť dovolená.

Funkcionalita protikolízneho systému zaručuje, že každá požadovaná hodnota vychádzajúca z ovládania musí byť schválená protikolíznym systémom.

Pre lepšie pochopenie si predstavme príklad: Robot ide istou rýchlosťou. Páčka na ovládanie je stále posunutá hore. To znamená, že požadovaná rýchlosť má byť buď rovnaká (ak ide o maximálnu rýchlosť) alebo sa má zvyšovať. Pred tým, než riadiaca jednotka vyšle signál o požadovanej rýchlosti do regulátora, *collision avoidance* vďaka lidarovi skontroluje priestor pred robotom.

V prípade, že sa podľa natočenia kolies nachádza prekážka v dráhe robota, prepočíta sa maximálna rýchlosť, akou môže robot ísť aby bezpečne zastavil a nenarazil. Ak sa v trajektórii robota nenachádza žiadna prekážka, stane sa požadovaná rýchlosť dovolenou a riadiaca jednotka vyšle

signál pre motor, aby zaistil požadovanú hodnotu rýchlosti. Aby sa požadovaná rýchlosť dosiahla čo najrýchlejšie a čo najkontinuálnejšie, využije sa spomínaný PID regulátor.

5.3 Riadenie natočenia kolies

PID regulátor sa podobne využije aj pri nastavovaní natočenia kolies. Pri regulácii natočenia sa podobne ako pri regulácii rýchlosti vyskytujú tri stavy. Uhol o ktorý sú kolesá natočene (získané lidarom), uhol o ktorý chceme kolesá natočiť (požadovaná hodnota) a uhol, ktorý môžeme kolesá natočiť (dovolená hodnota).

Opäť ide o rozhodnutie, kedy sa požadovaná hodnota stane dovolenou. Môže sa stať, že robot prechádza popri nejakej prekážke, ktorá momentálne nie je v jeho dráhe, preto protikolízny systém dovolí chod robota. Ak by sa ale neopatrnosťou človeka, ktorý riadi robota stalo, žeby chcel zmeniť natočenie kolies tak, aby sa robot hnal k prekážke, malo by sa tomu v predstihu zabrániť.

Príklad pre kontrolu natočenia je nasledujúci: Robot sa pohybuje priamo popri nejakej prekážke (napríklad stene), ktorá sa nachádza po ľavej strane. Joypad vráti zmenu natočenia vľavo. Prekážka môže byť pomerne ďaleko, ale aj blízko. V tomto prípade, protikolízny systém vyráta aká veľká je brzdná dráha, vo vzťahu k aktuálnej rýchlosti, k tejto prekážke. Laserový diaľkomer nám takisto vráti vzdialenosť k tejto prekážke. Ak je brzdná dráha k tejto prekážke väčšia ako jej vzdialenosť, potom sa požadovaná hodnota natočenia ignoruje. V opačnom prípade sa požadovaná hodnota stane dovolenou a servomotory natočia kolesa požadovaným smerom.

6 Implementácia

V tejto kapitole si konkrétne popíšeme zostavenie programu do tried, ktoré boli potrebné. Systém, sme implementovali pomocou prostredia Qt4 s použitím kompilátora Microsoft Visual C++ 2008. Hlavnou podpornou knižnicou, z ktorej sme odvodzovali aj vlastné triedy bola knižnica MRPT vid' nasledujúca podkapitola. Vychádzali sme z už navrhnutej aplikácie, ktorá nám slúžila na vizualizáciu dát a výpočtov, ktoré sme prevádzkali a jej náhľad je uvedený v prílohe A.

6.1 Knižnica MRPT

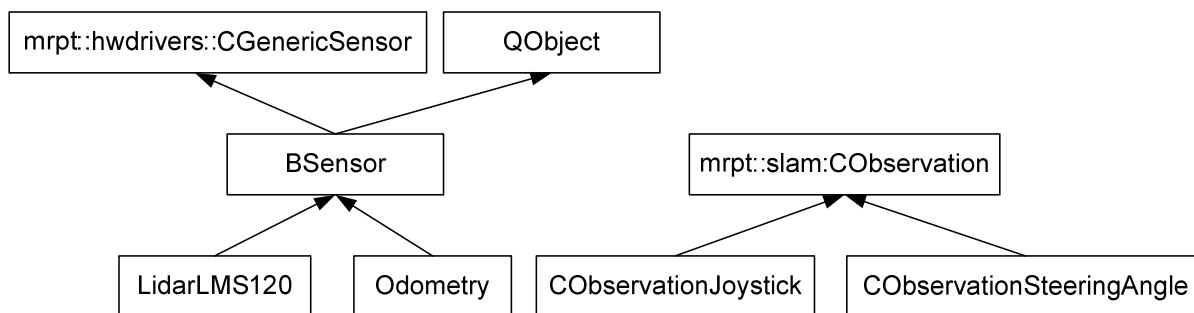
Jedným z nástrojov pri vytváraní tejto bakalárskej práce bola knižnica MRPT (Mobile Robot Programming Toolkit) [12]. Je postavená na jazyku C++ a zahrňuje v sebe knižnice pre správu 3D geometrie, funkcie hustoty pravdepodobnosti, rôzne filtre, 3D vizualizácie a iné. Takisto pomáha vytvárať súbory s dátami, ktoré nepochybne potrebujeme pre analýzu či už odometrie, pohybu, jazdných vlastností alebo iba jednoduché zapamätanie si trasy po ktorej prešiel. Tieto zaznamenané dáta, tzv. raw dáta môžu byť uložené ako súbor sledovaní. Tie je možné potom prechádzať a analyzovať pomocou vizualizačného prehrávača RawLogViewer, ktorý je takisto súčasťou knižnice MRPT. Knižnica MRPT je voľne dostupná, stále vyvíjaná a na jej stránkach je možné sledovať aktuálny zoznam podporovaných senzorov a hardwarových zariadení.

6.2 Hierarchia tried

Implementovali sme niekoľko potrebných tried. Niektoré z nich, napríklad LidarLMS120 a Odometry boli vytvorené dedením. Dedili sme ich z tried prístupných z knižnice MRPT. Na obrázku 6.1 je vidieť celú hierarchiu dedenia pre naše triedy. Triedy, ktoré sme vytvárali, ale nie sú zobrazené na obrázku platí, že nebola dedičnosť potrebná.

Na spomenutom obrázku je prvá uvedená dedičnosť senzorov laserového diaľkomera a odometrického senzora. V knižnici MRPT je vytvorených niekoľko tried senzorov, preto sme využili niektoré z nich, odvodili sme si vlastnú triedu BObject a z nej sme vytvorili triedy pre senzory s potrebnými vlastnosťami, ktoré sme potrebovali.

Druhou dedičnosťou na obrázku sú pozorovania. Pre senzory existuje v knižnici MRPT trieda, ktorá obsluhuje senzory a ich činnosť je založená na pozorovaniach. Vytvorili sme vlastné odvodené triedy pre pozorovania na ovládanie (CObservationJoystick) a natočenie kolies (CObservationSteering-Angle).



Obrázok 6.1 Dedičnosť tried a) triedy pre senzory b) triedy pre pozorovania

6.3 Popis tried

Každému hardwarovému zariadeniu, či už je to odometrický senzor, laserový diaľkomer, sonary alebo diaľkové ovládanie (joypad) prislúcha aj komunikujúca trieda, ktorá dáta z toho zariadenia spracováva a využíva pre ďalšie účely.

Hlavným cieľom bolo vytvoriť program protikolízneho systému (triedu CollisionAvoidance). Ten však sám o sebe bez iných častí nie je použiteľný. Na to, aby bol využiteľný, potrebuje dáta z prostredia. Tie mu poskytujú jednotlivé senzory, ktoré majú takisto svoje obsluhujúce triedy. Pomocou takéhoto spojenia dostaneme funkčný celok. V tejto podkapitole si podrobne popíšeme triedy, ktoré navzájom spolupracujú a tak sa podieľajú na celkovom fungovaní *collision avoidance*.

6.3.1 Trieda CollisionAvoidance

Centrálnou funkciou celého protikolízneho systému je trieda CollisionAvoidance. Zbiera informácie z dostupných senzorov, robí výpočty a následne aktívne zasahuje do riadenia robota.

Na začiatku je pre výpočet potrebné nastaviť niekoľko parametrov. Sú to geometrické parametre robota ako šírka, rázvor náprav a umiestnenie laserového diaľkomera. K týmto parametrom sú potrebné aj tolerancie, na ktoré sa pri výpočte musí brať ohľad. Sú to odchýlky vzdialenosti pred prekážkou, okolie robota aj časové oneskorenie, ktoré je medzi výpočtom a skutočným zásahom do riadenia robota.

6.3.2 Trieda LidarLMS120

Na začiatku je potrebné nadviazať spojenie s laserovým diaľkomerom. To sa uskutoční po nastavení parametrov ako sú IP, porty, frekvencia snímania a iné nastavenia, ktoré sú potrebné pre spojenie s diaľkomerom. Komunikácia prebieha pomocou TCP/IP spojenia prenosovou rýchlosťou 100 Mbit/s, ktoré podporuje plný aj polovičný duplex. S lidarom sa komunikuje pomocou správ, ktoré podporujú požiadavky na namerané hodnoty, nastavovanie parametrov komunikácie a zistenie stavu v ktorom sa lidar nachádza. Skontroluje sa či je spojenie s diaľkomerom uskutočnené a začnú sa odosielať príkazy

pre diaľkomer. Následne sa dekodujú prichádzajúce reťazce a informácie sa uložia do príslušných premenných [8].

Dáta z lidarů prichádzajú vo forme scanov (pozorovaní). V jednom pozorovaní je viacero údajov, ktoré odpovedajú jednému obrazu okolia. Okrem časových údajov sa tam nachádza 1082 hodnôt, ktorých hodnota zodpovedá vzdialenosti k tej prekážke, ku ktorej bol vyemitovaný laserový lúč. Jednou z často používaných metód, ktoré sú k dispozícii je metóda, ktorá konvertuje jeden vektor vzdialenosti z pozorovania, do troch vektorov súradníc (x, y, z). Hodnoty súradníc vo vektore osi z sú nulové, pretože laserový diaľkomer sníma priestor v jednej rovine (stačia súradnice x a y).

6.3.3 Trieda Odometry

Trieda pre odometrický senzor, teda trieda poskytujúca údaje o rýchlosti robota bola odvodená podobne ako laserový diaľkomer, vid' obrázok 6.1.

Po nastavení portu, na ktorom je tento senzor pripojený je potrebné nastaviť parametre prenosu. Vytvorí sa spojenie, ktoré sa skontroluje. Hneď ako je komunikácia zabezpečená, senzor začne posielat' dáta vo forme trojbytových blokov.

Na začiatku každého validného bloku sa nachádza synchronizačná jednotka, ktorá nám hovorí, že ďalšie informácie sú na miestach ako predpokladáme. Z každého bloku tak môžeme vyčítať počet tzv. tikov, ktorý nás informuje o pohybe senzora po osi x a osi y. Pozície jednotlivých bitov v celom bloku sú znázornené v tabuľke 6.1.

Byte	P7	P6	P5	P4	P3	P2	P1	P0
1.	-	1	LB	RB	Y7	Y6	X7	X6
2.	-	0	X5	X4	X3	X2	X1	X0
3.	-	0	Y5	Y4	Y3	Y2	Y1	Y0

Tabuľka 6.1 Pozície bitov v blokoch, P - pozícia, LB – left button, RB – right button [13]

Jednotlivé bity je treba správane pospájať, aby sme dostali posun po jednotlivých osiach. Posun po osi x bude v bitovom čísle vyzerat' X7X6X5X4X3X2X1X0 a po osi y Y7Y6Y5Y4Y3Y2Y1Y0, ale taktiež nesmieme zabúdať kontrolovať každý blok na synchronizačnú jednotku. Pretože nám ide o sledovanie rýchlosti, postačí nám jeden smer, teda jedna os.

6.3.4 Trieda PID

Pre riadenie výkonu motora, či už išlo o brzdenie alebo zvyšovanie výkonu sme potrebovali regulátor, ktorý by nám túto zmenu patrične zaistil. Pomocou matematického vyjadrenia fungovania PID

regulátora sme potom implementovali triedu, ktorá nám príslušný zásah do riadenia vypočíta. Na rozdiel od hardwarového regulátora, kde sú konštantami elektrické súčiastky, sme v softwarovej verzii iba menili hodnoty premenných a tak získali patričné chovanie regulátora.

6.4 Konfigurácia

V mnohých triedach, či už tých, ktoré sme implementovali my alebo v tých, ktoré sme len používali, vystupuje mnoho konfiguračných premenných. Môžu nimi byť premenné opisujúce prostredie, v ktorom sa robot nachádza (brzdné zrýchlenie) i rozmery robota (pre protikolízny systém) atď.. Tak isto sa tam nachádzajú aj frekvencie pod akými majú zo senzorov dáta prichádzať, resp. konfigurácie elektronických zariadení ako je lidar alebo sonary, takisto aj potrebné konštanty pre PID regulátory. Aby sme nemuseli stále hľadať danú triedu a modifikovať hodnoty v zdrojovom kóde, je vytvorený konfiguračný súbor *quido_config.txt*, v ktorom sa tieto konštanty a premenné dajú ľahko modifikovať.

Takýto konfiguračný súbor je rozdelený na sekcie, ktorá začína názvom v hranatých zátvorkách a pod ňou sa nachádzajú inicializácie jednotlivých premenných. Schéma konfiguračného súboru a príklad jednej sekcie je v nasledujúcej tabuľke:

Schéma sekcií	Príklad sekcie
[názov sekcie1] názov_premennej_11 = hodnota názov_premennej_12 = hodnota [názov sekcie2] názov_premennej_21 = hodnota názov_premennej_22 = hodnota	[CollisionAvoidance] stop_front_distance = 0.4 stop_rear_distance = 0.4 time_delay_control = 0.1 breaking_acceleration = 1.2 speed_tolerance = 1.0 wheelbase_ = 0.53 robotWidth_ = 0.5

Tabuľka 6.2 Schéma konfiguračného súboru

7 Záver

Robotika je rozmanitá veda, ktorá zasahuje do mnohých oblastí technického sveta. Ak sa chceme v tejto oblasti pohybovať, mali by sme mať základy elektrotechniky a programovania. Ich kombinácia vytvára úspešný predpoklad efektívnej práce.

Zadanie práce, ktorým bolo vytvorenie protikolízneho systému sme splnili. Robot autonómne rozoznáva objekty nachádzajúce sa v jeho dráhe, v ktorej sa predpokladá, že bude pokračovať. Dosah laserového diaľkomera, ktorý je očami robota, je síce iba 20 metrov, no pre rýchlosti ktorými sa bude robot pohybovať to stačí. Pre bezprostrednú vzdialenosť používa robot sonary. V prípade, že sa priblíži k robotovi prekážka, ktorú laserový diaľkomer nezachytí, sonar túto hrozbu lokalizuje a využije brzdný systém robota na jeho rýchle zastavenie. Robot nielenže v dostatočnom predstihu brzdí, ale takisto aj kontroluje riadenie z diaľkového ovládania. Dokáže predvídať smer robota pri žiadanom natočení kolies a ak hrozí kolízia, toto natočenie neprevedie.

Práca sa preto vypracovávala v dvoch štádiách. Jedným bolo sledovanie okolitého priestoru pomocou laserového diaľkomera a dolovanie informácií z prichádzajúceho dátového toku z diaľkomera. Nad nimi sme potom vytvárali metódy, ktoré produkovali nejaký stav podľa, ktorého sa upravovalo riadenie. Druhým štádiom bol prevenčný systém kolízie, ktorý dokázal predvídať smer robota, podľa žiadaného natočenia kolies. Bolo potrebné si uchovať posledný obraz z laserového diaľkomera a podľa neho sa potom riadiť. Medzitým sme pomerne rýchlo doimplementovali systém sonarov vpredu a vzadu, ktorý sníma bezprostrednú vzdialenosť okolia robota.

Vytvorený systém zohľadňuje iba statické objekty a prekážky. Ďalším zaujímavým vylepšením by preto mohla byť implementácia rozpoznávania pohyblivých objektov. Vlastnosti pohybujúceho sa objektu pred robotom, by potom mohli byť takisto využité na predvídanie jeho smeru. Takýmito detailmi by postupne mohlo dochádzať k stále dokonalejšiemu protikolízному systému.

Vypracovanie tejto bakalárskej práce určite rozšírilo moje vedomosti o tom ako to vo svete robotiky chodí. Veľmi potešujúce bolo vidieť zmeny na správaní sa robota po niekoľkých dňoch práce. Pevne verím, že som svojou prácou aspoň trochu prispel tímu *Roboauto*, ktorého cieľom je vytvoriť autonómne riadené vozidlo.

Literatúra

- [1] Konstantinidis, E.I., Patoulidis, G.I., Vandikas, I.N. *Development of collaborative vehicle collision avoidance system* [online]. Posledná modifikácia: 16. Augusta 2010. [cit. 2011-3-10]. Dostupné z WWW: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5548022>>. ISBN 978-1-4244-7866-8.
- [2] Thatcham. *Thatcham research news: Collision Avoidance* [online]. [cit. 2011-3-11]. Dostupné z WWW: <http://www.thatcham.org/research/pdfs/research_news_vol3_issue2.pdf>.
- [3] Driverless car [online]. 2011-4-1 [cit. 2011-04-19]. Wikipedia. Dostupné z WWW: <http://en.wikipedia.org/wiki/Driverless_car>.
- [4] Windmann, G.R., Bauson, W.A., Alland, S.W. *Development of collision avoidance systems at delphi automotive systems* [online]. IEEE International Conference on Intelligent Vehicles. [cit. 2011-3-10]. Dostupné z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.601&rep=rep1&type=pdf>>.
- [5] Seiler, P., Bongsob, S., Hedrick, J.K. *Development of a Collision Avoidance System* [online]. University of California-Berkeley. [cit. 2011-3-15]. Dostupné na WWW: <http://vehicle.me.berkeley.edu/Publications/AVC/pseil_sae98.pdf>.
- [6] Volvo V60 New 2011 model | Active and passive safety for V60 [online]. 2009 [cit. 2011-04-19]. Carnewsandreviews.info. Dostupné z WWW: <<http://carnewsandreviews.info/volvo-v60-safety-systems.html>>.
- [7] Continental sensor system in the new Volvo XC60 [online]. 2008 [cit. 2011-04-19]. Enewsautocar.com. Dostupné z WWW: <<http://www.eneautocar.com/continental-sensor-system-in-the-new-volvo-xc60/2008/03/31/0210>>.
- [8] Operating instructions LMS100/111/120/151 [online]. 2008, 18. marca 2011 [cit. 2011-04-19]. Robotsinsearch.com. Dostupné z WWW: <<http://robotsinsearch.com/file/SICK/LMS123-10000%20Security/LMS123-10000-operating%20instructions.pdf>>.
- [9] RcTek – Radio Controlled Model Car Handling – The Ackerman Steering Principle [online]. 2009 [cit. 2011-04-19]. Rctek.com. Dostupné z WWW: <http://www.rctek.com/technical/handling/ackerman_steering_principle.html>.
- [10] Plynulá regulace PID [online]. 2003-2006 [cit. 2011-04-19]. Valter.by1.cz. Dostupné z WWW: <<http://valter.by1.cz/plynula-regulace-pid>>.
- [11] Automa: Co znamená PID [online]. 2003 [cit. 2011-04-19]. Odbornecasopisy.cz. Dostupné z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=28768>.
- [12] The Mobile Robot Programming Toolkit [online]. 2010 [cit. 2011-04-19]. Mrpt.org. Dostupné z WWW: <<http://www.mrpt.org/About>>.

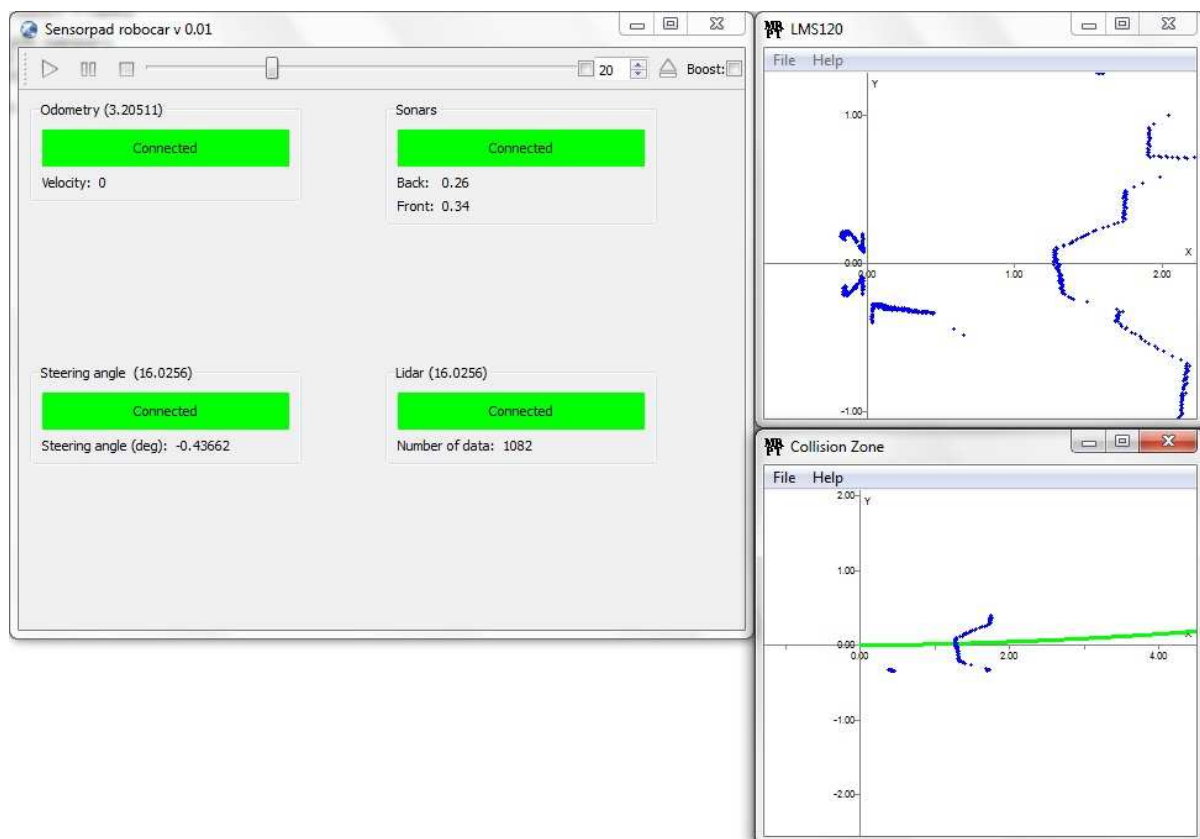
- [13] *PC mouse info* [online]. 1995-1998 [cit. 2011-04-19]. Epanorama.net. Dostupné z WWW: <<http://www.epanorama.net/documents/pc/mouse.html>>.
- [14] Robotics. In *Wikipedia : the free encyclopedia* [online]. Wikipedia Foundation. [cit. 2011-04-19]. Dostupné z WWW: <http://en.wikibooks.org/wiki/Robotics/Real_World_Sensors>.
- [15] *Odometrie (Robotika.cz > Guide)* [online]. 2005 [cit. 2011-05-03]. Robotika.cz. Dostupné z WWW: <<http://robotika.cz/guide/odometry/en>>.

Zoznam príloh

Príloha A. Ukážka aplikácie

Príloha B. CD so zdrojovými kódmi, technickou správou a spustiteľnou aplikáciou

Príloha A Ukážka aplikácie



V okne s názvom „LMS120“ sa vykresľujú všetky body vôkol robota a v okne „Collision Zone“ sa zobrazuje výsledok výpočtu, ktorý selektuje body nachádzajúce sa v trajektórii robota. Taktiež vykresľuje predpokladanú dráhu, po ktorej sa bude robot pohybovať pri aktuálnom natočení kolies.