

Technická Univerzita v Košiciach, Fakulta Elektrotechniky a Informatiky
Katedra Kybernetiky a Umelej Inteligencie

Záverečná správa k zadaniu z predmetu Hybridná Výpočtová Inteligencia
Využitie HVI pri navigácii robotov

Dominik Horňák

Ivan Vojtko

Štefan Labbancz

Richard Halčín

Igor Marga

2018

Úvod

Vo svete je riadenie a navigácia mobilných robotov veľmi modernou a populárnou témou. S príchodom a rozvojom Umelej Inteligencie sa otvorili možnosti pre vyriešenie rôznych problémov v tejto oblasti. Jedná sa o riadenie autonómnych vozidiel a tým defacto odstránenie ľudského faktoru, ktorý často zlyháva z ciest alebo o riadenie iných špecializovaných strojov (robotov) ako sú špecializované vojenské roboty, automatizácia stavebnej techniky, dopravné roboty v rôznych skladoch (Amazon) alebo továrňach (Volkswagen) a mnoho iných oblastí. A toto je len výber pár príkladov z množstva rôznych oblastí kde tieto technológie vedia nájsť uplatnenie.

V prvej kapitole je uvedené znenie zadania, ciele a podmienky. Druhá kapitola popisuje analýzu problému. Taktiež spomína riešenie podobných problémov a články, ktorými bolo toto zadanie inšpirované. Do veľkej miery popisuje technickú stránku samotného Neuro-Fuzzy systému.

V tretej kapitole je uvedený aktuálny stav a uplatnenie podobných projektov v reálnej praxi. Je tu uvedených pár príkladov z reálneho života, kde je jasne vidno, že táto problematika je veľmi aktuálna a zaujímavá.

Štvrtá kapitola popisuje postup riešenia problému. Kroky, ktoré sme museli urobiť aby sme sa dostali od stanovenia cieľa po samotnú implementáciu systému na reálnom robotovi. V tejto kapitole sú taktiež načrtnuté problémy ktoré prispeli k nie veľmi pozitívnemu výsledku tohto zadania.

Piata kapitola hovorí o experimentoch, testovaní následnom postupe pri opravovaní (odstraňovaní) problémov a výsledkoch jednotlivých experimentov. Taktiež tu je stručne zhodnotený hlavný dôvod prečo má zadanie negatívne výsledky a sú tu stručne navrhnuté možné kroky, ktoré sa mohli urobiť inak aby sa tomuto výsledku predišlo.

V závere je uvedená kapitola o administratívnych veciach spojených s riešením zadania, ktorý člen tímu čo robil a aké mal kompetencie.

Zadanie

Navrhnuť a implementovať systém pre riadenie mobilného robota s využitím prostriedkov Hybridnej Výpočtovej Inteligencie. Konkrétny cieľ zadania je v réžii každej vývojovej skupiny.

Podmienky

- Systém musí využiť prostriedky Hybridnej Výpočtovej inteligencie, tj. kombinácia rôznych metód výpočtovej inteligencie, FALCON a mnohé iné fantázií sa mädzé nekladú
- Z dôvodu, že táto technológia je používaná v zadaniach často nemôže byť použitá technológia ANFIS
- Pracovať na zadaní v skupine a priebežne
- Vypracovať záverečnú správu k riešeniu zadania

Cieľ

Implementovať systém kombinovaný z Neurónovej siete a Fuzzy regulátora určený pre navigáciu mobilného robota (konkrétne použitý TurtleBot 2) v miestnosti. Robot by sa mal dostať do určeného cieľového bodu a nenaraziť po ceste do žiadnej prekážky.

Analýza

Prvým krokom pri riešení problému bolo jasne si zvoliť cieľ zadania, rozdeliť si úlohy v tíme a preskúmať rôzne možnosti ako by sme sa vedeli posunúť od stanovenia cieľa k výsledku.

Náš tím sa skladal z piatich členov a rozdelili sme si roly nasledovne:

- **Manager** – Dominik Horňák
- **Analytik** – Igor Marga
- **Prezentér** – Štefan Labbancz
- **Programátor #1** – Ivan Vojtko
- **Programátor #2** – Richard Halčín

Preskúmanie cieľa pochopiteľne padlo na tímového analytika, no keďže prehľadanie internetu nieje krátkodobá záležitosť rozhodli sme sa, že mu s tým pomôžeme a prehľadávali sme možné spôsoby riešenia nášho problému spoločne.

Hľadaním riešení sme dospeli k trom možným článkom, kde bol riešený podobný problém, z ktorých sme čerpali ďalšiu inšpiráciu. Pozn.: odkazy na články sú na konci v kapitole Použitá literatúra.

V článku [2] sa autori zamerali na riadenie mobilného robota s diferenciálom, čo nebolo v našich podmienkach možné zreprodukovať a taktiež používali technológiu ANFIS, ktorá bola pre nás obmedzením. Preto sme sa rozhodli touto cestou neísť.

Článok [3] zase využíva kombináciu Neurónovej siete a Fuzzy systému a nevyžaduje robota s diferenciálnym pohonom kolies, no tento systém sa nám zdal ako zbytočne príliš zložitý čiže sme sa tomuto prístupu opäť vyhli.

Najpriateľnejšou cestou v našich podmienkach však bolo riešenie v článku [1]. Tu sa riešil problém riadenia dvojkolesového robota použitím kombinácie Neurónovej siete a Fuzzy systému podobne ako v článku [3], no systém bol omnoho jednoduchší a bližší ľudskému chápaniu sveta. V tomto článku však autori pracovali len v simulačnom prostredí čo nám dalo priestor na obohatenie ich práce o vyskúšanie systému na reálnom robotovi. Samozrejme, vzhľadom na časové obmedzenie semestra a iné okolnosti vznikli počas práce problémy o ktorých sa bližšie zmienim pri vyhodnotení práce.

Štruktúra systému

Po vzore článku [1] náš systém sa skladal zo štyroch samostatne funkčných celkov.

Neurónová sieť mala štyri vstupné neuróny konkrétne vzdialenosť od prekážky kolmo vľavo, vzdialenosť od prekážky kolmo vpravo, vzdialenosť od prekážky pred robotom a nakoniec uhol natočenia robota od cieľovej pozície v priestore. Za vstupnými neurónmi boli umiestnené dve skryté vrstvy kde v prvej bolo desať neurónov a v druhej boli neuróny tri. Nakoniec bol na výstupe jeden neurón ktorého výstupná hodnota mala reprezentovať uhol ktorým by sa robot mal reálne natočiť v danej situácii. (Príklad: stena predomnou je vzdialená 35cm, naľavo mám prekážku vzdialenú 35cm ale napravo prekážku nemám resp. je veľmi ďaleko a moje natočenie od cieľa je -15° tak aj napriek tomu že by k cieľu by som sa najlepšie dostal pri natočení -15° na výstupe sa očakáva 90° - natočenie vpravo aby som sa vyhol prekážke). Samotná neurónová sieť bola trébovaná klasickým backpropagation algoritmom s využitím knižnice Tensorflow (www.tensorflow.org).

Toto bola v teórii pekná myšlienka no nie veľmi úspešne vykonaná, dôvody obzrejším v časti s vyhodnotením.

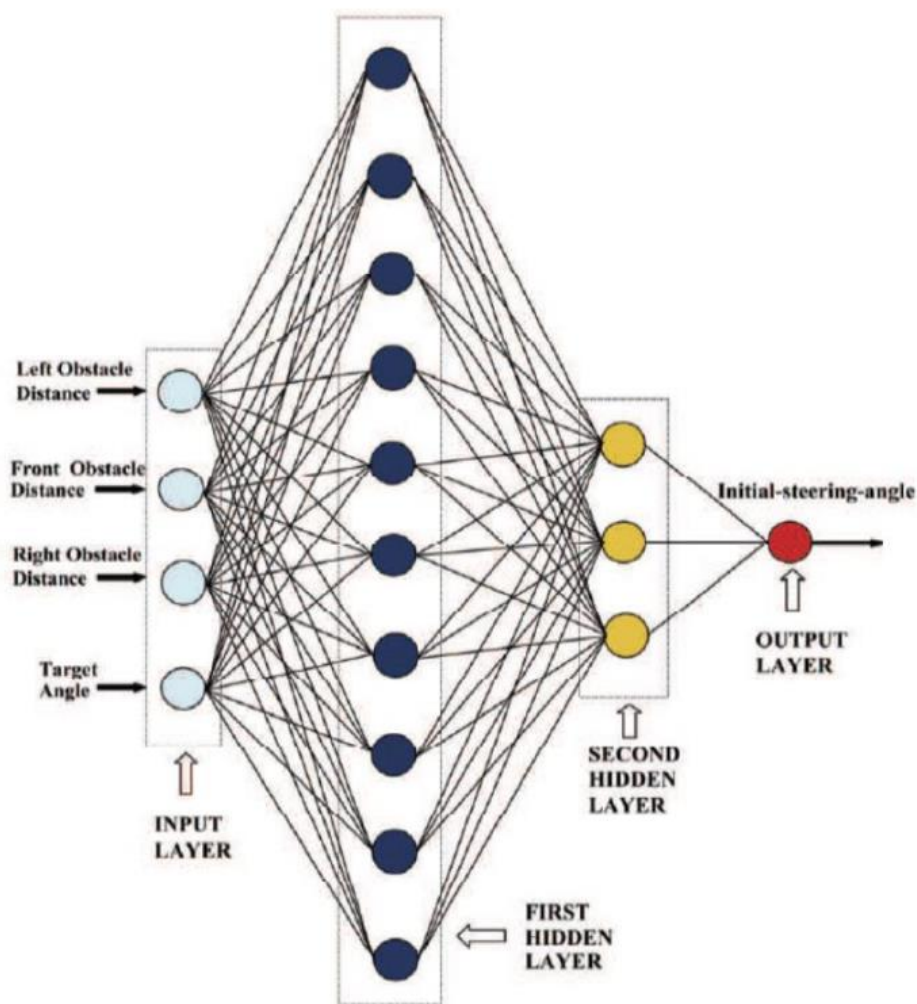


Image 1: Štruktúra použitej neurónovej siete

Fuzzy regulátor bol umiestnený hneď za neurónovou sieťou a jeho úlohou bolo premeniť výstupný uhol z neurónovej siete na uhlové rýchlosti pravého a ľavého kolesa robota. Štrukturálne mal tento regulátor podobne ako neurónová sieť štyri vstupy, tri vzdialenosti zo senzorov ale štvrtým vstupom bol práve výstup z neurónovej siete. Výstupom z fuzzy regulátora boli spomínané uhlové rýchlosti pre pravé a ľavé koleso, ktoré išli do modulu pre riadenie robota (API Turtlebot 2 v systéme ROS).

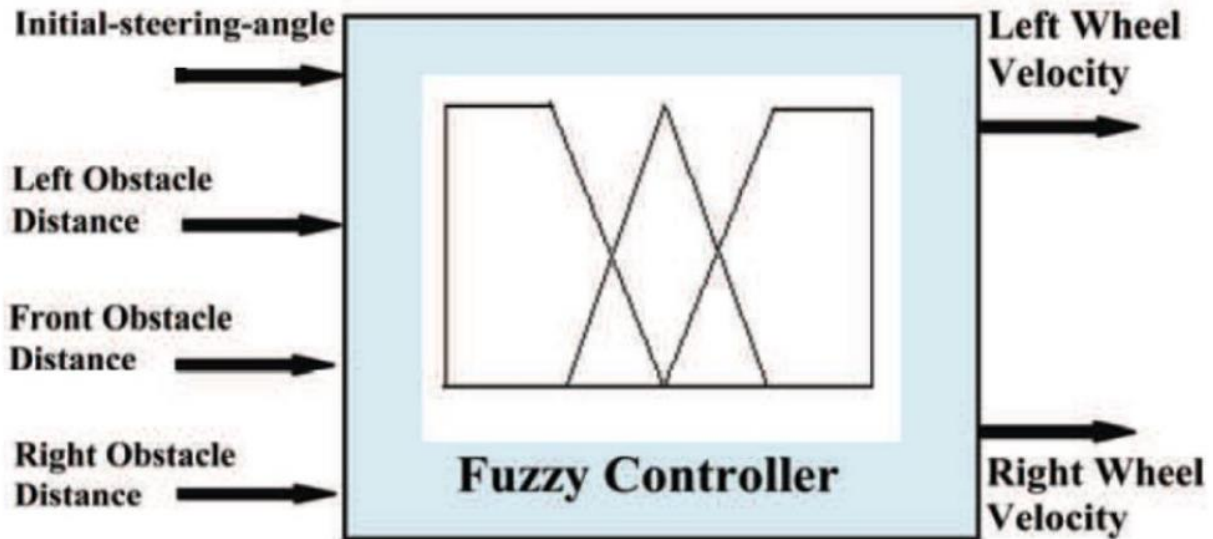


Image 2: Náčrt vstupov a výstupov fuzzy regulátora

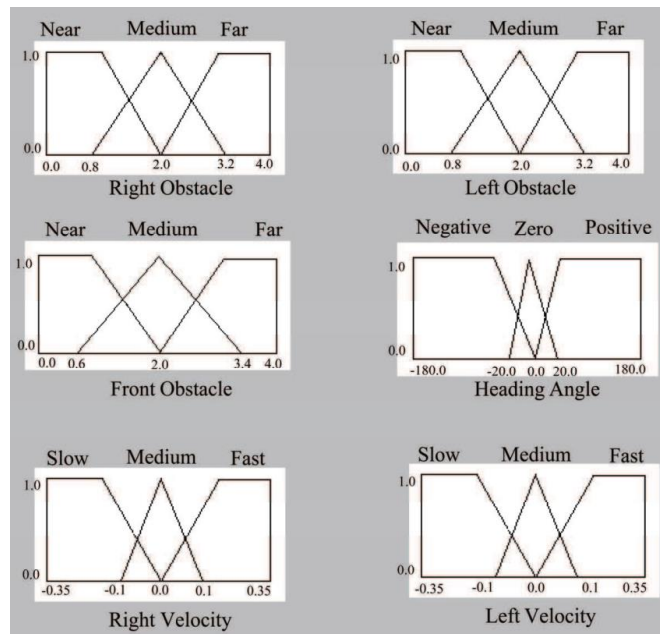


Image 3: Fuzzy funkcie príslušnosti

Fuzzy pravidlá boli v tvare:

IF LeftObstacle IS Far AND FrontObstacle IS Near AND RightObstacle IS Med THEN LeftVelocity IS Slow AND RightVelocity IS Fast

Hardvér (Arduino a senzory) bol použitý na získavanie vzdialeností od prekážok. Senzory fungovali na báze vysielania ultrazukových vln a merania času od vyslania vlny po jej vrátenie sa k senzoru. Z toho bola počítaná vzdialenosť od prekážky. Tieto výpočty však prebiehali priamo na doske senzoru a nebolo potrebné tieto details implementovať. Celý zber dát riadil jednočipový mikropočítač Arduino Nano. Úlohou Arduina bol zber dát a ich posielanie po sériovej linke cez USB port do hlavného programu, kde tieto dáta boli použité ako vstupy do Neurónovej siete a Fuzzy regulátora.

Dáta zo Arduina nám prichádzali už prevedené na centimetre a vo vhodnom formáte aby sme vedeli jednoducho rozlíšiť ktorá vzdialenosť patrí ktorému senzoru.

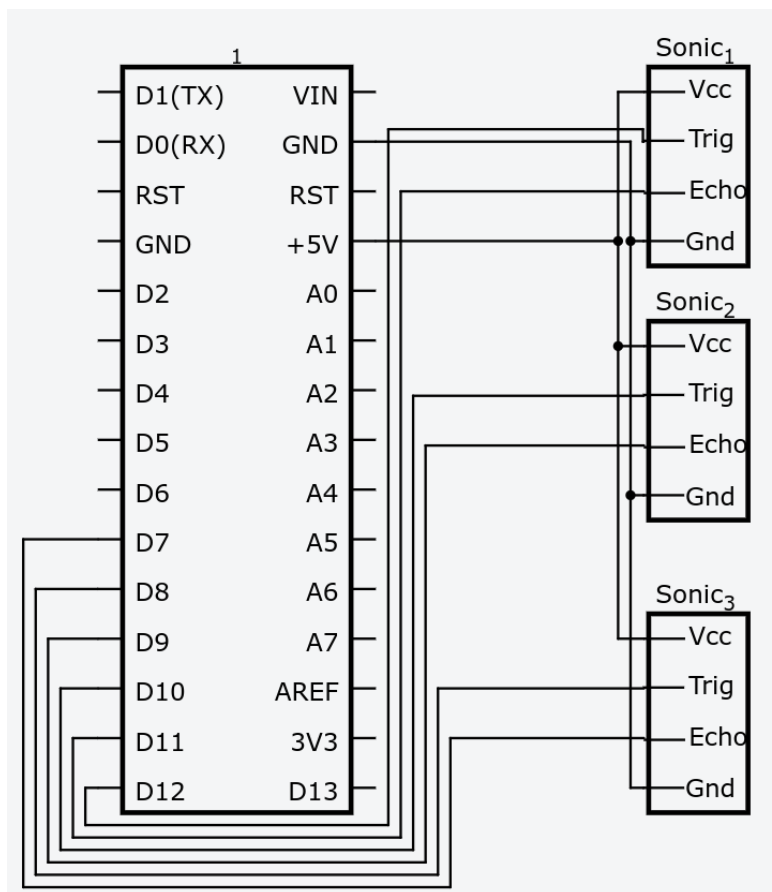


Image 4: Schéma zapojenia Arduino a senzory



Image 5: Umiestnenie senzorov na robotovi

Turtlebot a ROS je technicky taktiež hardvér, ale vzhľadom na charakter toho, že sa jednalo prakticky o samostatný modul je to v architektúre systému samostatná jednotka. Tento modul bol prakticky hotový modul, ktorý má OpenSource zdrojové kódy a je prakticky už implementovaný. ROS je operačný systém, ktorý nám poskytoval jednoduché API pre ovládanie jednotlivých kolies na robotovi. O samotnom robotovi sa dá bližšie dočítať na stránkach ROS-u (ros.org) alebo stránkach projektu pod ktorým sa robot vyvíja a vyrába (turtlebot.com).



Image 6: Obrázok ako samotný robot vyzerá

Aktuálny stav

Vo svete je riadenie a navigácia mobilných robotov veľmi modernou a populárnou témou. S príchodom a rozvojom Umelej Inteligencie sa otvorili možnosti pre vyriešenie rôznych problémov v tejto oblasti. Jedná sa o riadenie autonómnych vozidiel a tým defacto odstránenie ľudského faktora, ktorý často zlyháva z ciest alebo o riadenie iných špecializovaných strojov (robotov) ako sú špecializované vojenské roboty, automatizácia stavebnej techniky, dopravné roboty v rôznych skladoch (Amazon) alebo továrňach (Volkswagen) a mnoho iných oblastí. A toto je len výber pár príkladov z množstva rôznych oblastí kde tieto technológie vedia nájsť uplatnenie.

Mediálne asi najviac skloňovaným príkladom takéhoto využitia mobilnej robotiky sú autonómne autá s ktorými prišli Google, Tesla, no postupne sa do závodu o to kto bude prvý s dokonalým autonómnym vozidlom sa postupne zapojili aj tradičné automobilky akými sú Ford, Volkswagen, Mercedes a iné. No veľa úspechov v tejto oblasti má práve spomínaný Google, ktorý už tohto roku spúšťa pilotnú prevádzku plne autonómnej taxi služby na Floride v USA pod značkou Waymo. Na Google sa ale doťahuje Tesla so svojím Autopilotom, ktorý je štandardnou výbavou u všetkých modelov tejto automobilky. Ľudia berú túto technológiu veľmi pozitívne aj napriek tomu, že má svoje nedostatky (ako bolo možné vidieť v nedávnej fatálnej nehode, kde príčinou bol práve autopilot). Aj kvôli takýmto prípadom sú občas ľudia skeptickí voči tejto technológii. Ale s dostatkom času si myslím, že sa tieto nedostatky dotiahnu a môže sa to vyvinúť veľmi zaujímavým spôsobom.

Zo špecializovaných uplatnení spomeniem len Amazon, kde vyvinuli pre interné účely špeciálnych robotov, ktoré presúvajú tovar v skladoch pre optimálne umiestnenie. Tieto roboty sú plne autonómne a nevyžadujú žiaden zásah človeka.

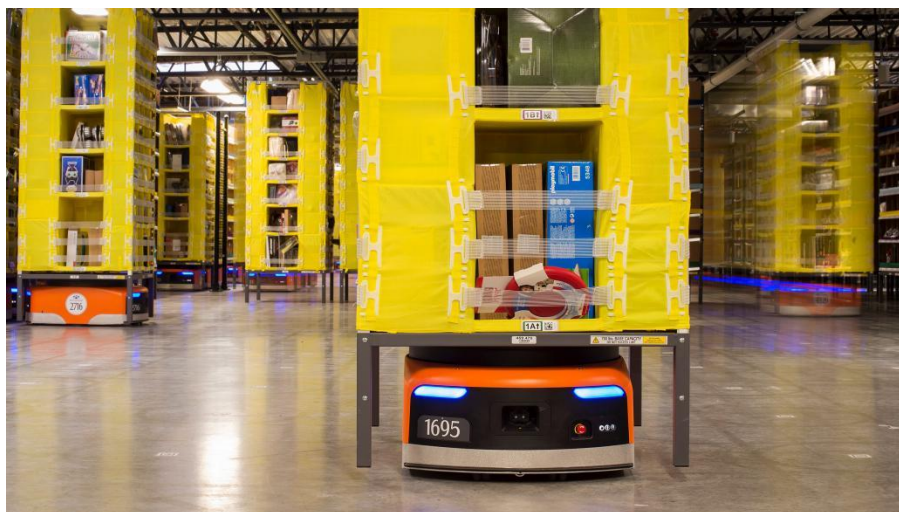


Image 7: Autonómny robot v sklade Amazonu

Postup pri riešení

Problém bol riešený tímovou spoluprácou. Všetci členovia tímu si poctivo plnili svoje úlohy, aj keď občas nie úplne na čas podľa dohodnutých časov, no to sa v takýchto projektoch dá očakávať.

Prvým krokom pri riešení úlohy bola analýza (popísaná v predošlých častiach) a následný návrh (dohodnutie sa) ako budeme problém riešiť. V tejto diskusii sme sa dohodli na ciele, čo vlastne od tohto zadania chceme, a približnej štruktúre ako chceme aby systém fungoval. Na tieto veci bolo potrebné preštudovať a prediskutovať článok [1] ktorým bola táto práca inšpirovaná. Výsledkom tejto diskusie bolo, že pre ďalší postup je potrebné mať hardvér (senzory, arduino....) aby sme vedeli aké dáta vlastne zo senzorov môžeme čakať. Tu prišla prvá komplikácia, keďže sme za senzormi čakali niekoľko týždňov, čo posunulo celé riešenie problému o tento čas. Nebol to veľký problém, ale vo výsledku by stratený čas mohol pomôcť.

Po vyskladaní hardvéru a preskúmaní dát ktoré sme zo senzorov mohli očakávať sme si určili obmedzenia:

min vzdialenosť 3cm - toto bolo posunuté do 15cm pretože 3cm je príliš blízko steny

max vzdialenosť 500cm - tu sme pri testovaní prišli na to, že je to zbytočne veľa a trénovacie dáta tým pádom boli zbytočne rozsiahle, čo malo negatívny vplyv na výsledok

presnosť senzorov +/- 0.5cm - veľmi kvalitné senzory

Z týchto údajov sme mali dostatok informácií pre vytvorenie trénovacích dát. Tu naskytol problém č. 2 trénovacie dáta použité v článku dostupné neboli. Preto sme museli improvizovať a vytvoriť si ich manuálne, čo spôsobilo asi najväčší dopad na výsledok práce. Tvorba dát prebiehala štýlom:

- vygenerovali sa situácie vo forme <distanceLeft, distanceRight, distanceFront, targetAngle> z rozsahu senzorov (stupeň 0° berieme že cieľ je priamo pred robotom)

- k týmto "situáciám" sa následne manuálne dopĺňala informácia o tom ktorým smerom sa na základe ľudského uváženia robot mal natočiť

Toto manuálne "olabelovanie" trénovacích dát sme robili všetci, každý na vlastnej náhodne vygenerovanej vzorke 500 situácií (spolu cca 2500 situácií). Tu vstúpil do hry ďalší faktor, každý z nás uvažuje inak a riešil situácie vlastným spôsobom, čo mohlo spôsobiť nekonzistenciu v dátach a následné problémy. Taktiež ako bolo spomenuté dáta boli generované na plnom rozsahu senzorov tj. 3cm - 500cm, čo bolo pre laboratórne podmienky zbytočne veľa a zbytočne to zvýšilo priestor hodnôt ktoré bolo

potrebné množstvom tréovacích dát pokryť pre správne naučenie neurónovej siete. V priestoroch laboratória sme len ťažko dosiahli vzdialenosť vyššiu ako 200cm.

V dátach sa tiež vyskytovali hodnoty -1 na pozíciách sensorov, tieto hodnoty znamenali vzdialenosť veľmi veľkú (nemerateľnú čiže viac ako 500cm) alebo veľmi malú (menej ako 3cm ale túto možnosť sme vylúčili tým, že sme robota nepúšťali do situácií kde by sa priblížil k stene bližšie ako 15cm). Toto mohlo spôsobiť ďalšie problémy pri tréovaní, práve kvôli tomu kde je hodnota -1 umiestnená na číselnej osi vzhľadom na hodnotu 500 a vzhľadom na hodnotu 15. Do budúcnosti, by bolo lepším riešením nahrádzať hodnotu -1 zo senzoru hodnotou 500cm a v tréovacích dátach takúto hodnotu vôbec neuvažovať.

Po vygenerovaní dát prišla na rad najkrajšia časť a to tréovanie neurónovej siete, ktoré bolo vďaka frameworku tensorflow veľmi jednoduché. Použitá bol klasický DNNRegressor, ktorý je prakticky klasickou implementáciou backpropagation algoritmu. Z tohto tréovania sa uložil model, ktorý sa následne na robotovi už len načítal a používal pre predikciu hodnoty uhla.

Programovanie a tréovanie neurónovej siete bolo robené paralelne s programovaním fuzzy regulátora pre ktorý bola použitá knižnica scikit-fuzzy z frameworku scipy. Pravidlá a tvary funkcií príslušnosti boli v článku veľmi dobre popísané, preto sme ich len prevzali a implementovali do nášho programu.

Nakoniec zostávalo už len dať jednotlivé moduly dokopy a implementovať algoritmus prepočtu nových súradníc cieľa vzhľadom na odometrické dáta z robota. Samotný program bol implementovaný multithreadovo tj. každý modul bežal vo vlastnom threade a upravoval spoločný priestor hodnôt z ktorých sa pravidelne vypočítavali nové hodnoty uhlových rýchlostí použitím Neuro-Fuzzy systému. Týmto multithreadovým prístupom bol vyriešený problém nesynchronizácie získavania dát z odometrie (poskytované ROS-om) a dát zo sensorov (poskytované Arduino cez serial).

Experimenty a výsledky

Testovanie zadaní (hlavne tých s robotmi) je tou najkrajšou časťou nášho odboru a nebolo tomu inak ani u rohto zadania. Prebiehalo prakticky celý deň, pretože sme sa stretli s mnohými problémami, ktoré počas návrhu a implementácie nebolo možné predvídať. Jedným z problémov bola napríklad chybná inštalácia ROS systému na robotovi, čo zabralo veľmi dlhú dobu pri opravovaní.

Experimenty boli vykonávané štýlom:

- umiestnili sme robota do stredu miestnosti
- spustili sme skript a prikázali sme robotovi aby šiel do pozície <50cm, 50cm> (uhlopriečne cca 62cm)
- sledovanie správania sa robota

Prvé experimenty boli neúspešné z dôvodu chybné implementácie niektorých častí (neošetrenie niektorých prípadov, napr chybný formát dát zo seriovej komunikácie a iné). Tieto problémy boli veľmi rýchlo odstránené, keďže sme mali tím kvalitných programátorov.

Následne sme pred ďalšími experimentami testovali jednotlivé moduly či fungujú tak ako majú (pre istotu). Testovali sa senzory pridávaním rôznych výpisov hodnôt do output-u hlavného skriptu, ktorý ovládal robota a ukladaním prekážok do ciest sensorov. Týmto testom sme prišli na to, že senzory fungujú v poriadku. Ďalej sa testovali výsledky odometrických výpočtov (nových target súradníc). Toto prebiehalo tak, že sme robota posúvali vstavanými funkciami na posun o určitú vzdialenosť počas toho ako bežal náš skript, ktorý periodicky prepočítaval nové súradnice. Tu sme zistili, že z dôsledkov nepresnosti odometrie vzniká chyba približne 8cm na 100cm prejednom, čo je veľmi dobrá presnosť pre tieto účely.

Testovať modul fuzzy regulátora a neurónovej siete samostatne nebolo možné, keďže tieto systémy boli navrhnuté tak aby fungovali dobre v reálnom čase a nie po jednotlivých vzorkách ale vzhľadom na jednoznačnosť pravidiel vo fuzzy regulátore sme označili tento modul za funkčný. Jediný modul, ktorý sa nedal označiť ako funkčný bola neurónová sieť, pretože nebolo možné oskúšať jej real-time fungovanie a vzhľadom na spôsob, ktorým boli tvorené dáta pre jej tréning sme sa zhodli, že ak bude niekde problém bude práve tam.

Po uistení sa, že jednotlivé moduly fungujú a pár last-minute opravách sa nám podarilo systém spojiť a robotom pohnúť. Robot však nešiel do cieľa ale krúžil skoro náhodným pohybom v jeho okolí približne 150cm každým smerom. Toto nebolo

očakávané správanie a preto sa začala investigácia, toho kde mohol nastať problém. Trénovacie dáta v neurónovej sieti boli robené tak, že ak mala neurónová sieť na všetkých vstupoch zo sensorov hodnotu -1 (prekážka je ďaleko), tak výsledný uhol je vlastne uhol smerom k cieľu. Preto pre "odstavenie" neurónovej siete sme prelepili senzory páskou aby posielali hodnotu -1 a došli sme k záveru, že samotný fuzzy regulátor je schopný dostať robota do cieľa, no nie optimálnym spôsobom (video ukážka je na webe projektu, jedná sa o jeden z mnohých pokusov, ktorý vyšiel)

Týmito zisteniami sme sa zhodli na tom, že problémy, ktoré nastali boli zapríčinené takmer jednoznačne spôsobom, ktorým boli tvorené trénovacie dáta. Tento problém by bolo možné odstrániť ak by sme napríklad mali k dispozícii čas stratený počas čakania na senrozy alebo úplnou zmenou prístupu k tvorbe daných dát (napríklad ovládanie robota človekom a zber dát práve počas tohto "chodenia" po miestnosti a obchádzania prekážok).

Uplatnenie v praxi

Ako už bolo spomenuté v aktuálnom stave, v praxi má táto tématika široké uplatnenie, aj keď možno nie priamo u dvojkolesových robotov ale všeobecne v navigácii je to veľmi “hot-topic” vo svete Kybernetiky, Umelej Inteligencie ale aj IT všeobecne.

Ako takú pikošku, čo bolo aj naším akýmsi nespĺneným cieľom tohto zadania, uvediem projekt robený práve na tomto robotovi (turtlebot), kde robot bol naprogramovaný aby prišiel k človeku s kávovarom. (<https://www.youtube.com/watch?v=ni49uYq4cF4>) Čiže akýsi robot nosič kávy, čo je v našom odbore veľmi dôležité uplatnenie.

Trochu administratívnych vecí

Z administratívneho pohľadu tohto zadania som s tímom veľmi spokojný. V tíme nás bolo 5:

- **Dominik Horňák** - manager
- **Igor Marga** - analytik (pravá ruka managera :D)
- **Štefan Labbancz** - prezentér
- **Ivan Vojtko** - programátor #1
- **Richard Halčín** - programátor #2

Všetci členovia tímu si plnili svoje povinnosti a úlohy zodpovedne a podieľali sa na práci rovnomerným dielom, vrámci svojich kompetencií.

Kto čo vlastne robil

Dominik Horňák - management tímu, posledné slovo pri rozhodnutiach, návrh architektúry systému, záverečná správa, dohľadanie na vykonávanie všetkého čo bolo treba urobiť, správa git repozitáru, programovanie Neurónovej siete a Odometrie

Igor Marga - analýza problému, hľadanie článkov, pomoc pri diskusiách a objasňovaní jednotlivých častí systému, zástup managera pri prezentovaní na cvičeniach, programovanie ROS časti robota

Štefan Labbancz - tvorba web stránky projektu, správa web stránky projektu, tvorba prezentácií, upravovanie prezentácií, dopĺňanie informácií na webe, prispievanie názormi do diskusií ohľadom systému

Ivan Vojtko - tvorba hardvéru (Arduino, senzozy - put it all together), programovanie Arduino časti systému, čítanie dát zo serialu v pythone, pomoc pri návrhu architektúry celého programu, linux guru

Richard Halčín - programovanie fuzzy regulátoru, pomoc pri vykonávaní experimentov, pomoc pri návrhu systému, riešenie problémov počas testovania

Záver

Cieľom tohto zadania bolo urobiť robota, ktorý mi prinesie kávu po tom čo ho dokončím. Žiaľ z rôznych príčin, no najmä kvôli chybnému prístupu k tvorbe dát sa nám to nie tak celkom podarilo dosiahnuť. Jednotlivé časti samotného systému su ale samostatne funkčné čo je vcelku pozitívne. Toto zadanie všetkým členom tímu prinieslo veľa skúseností z oblasti mobilnej robotiky a najmä kombinácie neurónových sietí s fuzzy systémami. Taktiež pre niektorých členov tímu bola vcelku novinka práca s hardvérom na tej trochu nižšej úrovni čo pre daných členov tímu bol určite pozitívny prínos do ďalšej praxe. Vo výsledku práca na tomto zadaní bola pre nás veľmi príjemným vyplnením chvíľ a určite toto zadanie patrilo medzi jedno z tých zaujímavejších v tomto štúdiu. A to aj napriek tomu, že sa nám nepodarilo dosiahnuť stanovený cieľ. Myslím si, že sme sa ale poučili do budúca a vieme akým prístupom sa možno v budúcnosti vyhnúť.

Použitá literatúra

- [1] Dr. Dayal R. Parhi. 2008. Neuro-Fuzzy Navigation Technique for Control of Mobile Robots. Dostupné online: <https://www.intechopen.com/download/pdf/5369>
- [2] Brahim S., Azouaoui O., Loudini M. 2015. Neuro-Fuzzy Navigation for a Car-Like Robot in Unknown Environments. Dostupné online: <https://pdfs.semanticscholar.org/ce40/a05ee38fffb489cafdc9d4b3eb231bd6318c.pdf>
- [3] Khaldoun K. Tahboub, Munaf S. N. Al-Din. 2009. A Neuro-Fuzzy Reasoning System for Mobile Robot Navigation. Dostupné online: <http://jjmie.hu.edu.jo/files/v3n1/jjmie-81-08%20Modified.pdf>