



UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A
INFORMATIKY



SAMOORGANIZÁCIA SENZOMOTORICKÝCH REPREZENTÁCIÍ
A ICH VYUŽITIE PRI UCHOPOVANÍ OBJEKTOV

Diplomová práca

2013
Bc. Matej Pecháč



UNIVERZITA KOMENSKÉHO
V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A
INFORMATIKY



SAMOORGANIZÁCIA SENZOMOTORICKÝCH REPREZENTÁCIÍ
A ICH VYUŽITIE PRI UCHOPOVANÍ OBJEKTOV

Diplomová práca

Študijný program: Kognitívna veda

Študijný odbor: 2503 Kognitívna veda

Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: doc. Ing. Igor Farkaš, PhD.

Konzultant: RNDr. Kristína Rebrová

Bratislava 2013

Bc. Matej Pecháč



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Matej Pecháč
Študijný program: kognitívna veda (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.11. kognitívna veda
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Samoorganizácia senzomotorických reprezentácií a ich využitie pri uchopovaní objektov

Cieľ:
1. Implementujte, otestujte a analyzujte rekurentnú samoorganizujúcu sa mapu pri vytváraní vizuálnych reprezentácií pohybujúceho sa robotického ramena, pre rôzne typy úchopov a rôzne perspektívy pohľadu. Zamerajte sa na aspekt invariančnosti neurálnych reprezentácií.
2. Naučte robotické rameno uchopovať objekt v reálnom prostredí pomocou učenia posilňovaním a vyhodnoťte správanie systému.

Kľúčové slová: senzomotorické reprezentácie, samoorganizácia, kognitívna robotika, neurónová sieť

Vedúci: doc. Ing. Igor Farkaš, PhD.
Konzultant: RNDr. Kristína Rebrová
Katedra: FMFI.KA1 - Katedra aplikovanej informatiky
Vedúci katedry: doc. PhDr. Ján Rybár, PhD.
Dátum zadania: 14.11.2012

Dátum schválenia: 14.11.2012

prof. RNDr. Pavol Zlatoš, PhD.
garant študijného programu

študent

vedúci práce,
konzultant

Čestne prehlasujem, že som túto diplomovú prácu
vypracoval samostatne s použitím citovaných zdrojov.

.....
Bc. Matej Pecháč

Chcem sa poďakovať svojmu vedúcemu doc. Ing. Igorovi Farkašovi, PhD. za cennú pomoc, rady, konzultácie a čas, ktorý mi venoval. Taktiež ďakujem svojej konzultantke RNDr. Kristíne Rebrovej za konzultácie, rady a ochotu s akou mi pomáhala. Na záver ďakujem Dipl.-Ing. Dr.techn. Michaelovi Zillichovi za jeho odbornú pomoc pri riešení projektu, ktorý sa stal súčasťou tejto práce.

Abstrakt

V našej práci ponúkame stavebné komponenty modelu zrkadliacich neurónov, ktorý je naším konečným cieľom, avšak je nad rámec tejto práce. Podrobnou analýzou existujúcich modelov sme identifikovali niekoľko problematických miest, či biologicky neplauzibilných predpokladov a snažíme sa jeden takýto podstatný problém premostiť práve naším modelom a posunúť sa viac k reálnemu modelu zrkadliacich neurónov. Inšpirovaní neurologickými výskumami sme preto zahrnuli do nášho modelu oblasť hornej spánkovej brázdy (STS), oblasť premotorickej kôry (PMC) a model dosahovania a úchopu. Všetky modely sú založené na výpočtovej paradigme neurónových sietí. Zameriavame sa na vznik vizuálnych reprezentácií úchopov a emergenciu perspektívne variantných a invariantných neurónov v modeli STS. Zároveň sme ponúkli jednoduchý model vykonávania úchopov, ktorý sme mali možnosť otestovať v reálnom prostredí a môže slúžiť ako generátor tréningových množín pre modely STS a PMC.

Kľúčové slova: zrkadliace neuróny, vrchná spánkova brázda, perspektívne variantné a invariantné reprezentácie, model uchopovania, kognitívna robotika, výpočtová inteligencia, porozumenie akciám

Abstract

In our thesis we propose building blocks of the mirror neurons model, which is our final goal but exceeds the scope of this thesis. We have carefully analyzed existing models and have identified several problematic points or biologically implausible presuppositions and our effort is to eliminate one such crucial problem in our new model and move towards a more realistic mirror neuron model. Inspired by neurological research we decided to include to the model the area of superior temporal sulcus (STS), premotor cortex area (PMC) and the model for reaching and grasping. All the models are based on the neural networks computation paradigm. We have focused on creation of visual representations of grasps and emergence of perspective variant and invariant neurons in the STS model. We have also proposed a simple robotic model of reaching and grasping, which was tested in a real environment and will be used as a dataset generator for STS and PMC models.

Key words: mirror neurons, superior temporal sulcus, perspective variant and invariant representations, model of grasping, cognitive robotics, computational intelligence, action understanding

Obsah

Úvod.....	4
1 Teoretické východiská.....	5
1.1 Vrchná spánková brázda.....	5
1.2 Systém zrkadliacich neurónov.....	9
1.3 Výpočtové modely.....	12
1.3.1 Model MNS.....	12
1.3.2 Modely uchopovania.....	14
2 Použité formálne metódy.....	15
2.1 Umelá neurónová sieť.....	15
2.2 Dopredná neurónová sieť.....	16
2.3 Samoorganizujúce sa mapy.....	17
2.3.1 SOM.....	17
2.3.2 Zlučovacia SOM (MSOM).....	19
2.3.3 RecSOM.....	20
2.4 Učenie s posilňovaním.....	22
2.4.1 Algoritmus učenia v spojitom priestore.....	23
3 Návrh modelov.....	24
3.1 Dátové množiny.....	24
3.1.1 Trénovacie dáta pre modely PMC a STS.....	24
3.1.2 Trénovacie dáta modelu CRG.....	27
3.2 Modul PMC.....	28
3.3 Modul pSTS a aSTS.....	29
3.4 Model CRG.....	31
3.5 Architektúra modelu zrkadliacich neurónov.....	33
4 Implementácia.....	35
4.1 Technologické riešenie.....	35
4.2 Dátové množiny.....	36
4.3 Samoorganizujúce sa mapy.....	37

4.4	Architektúra modelu STS.....	38
4.5	Architektúra modelu CRG.....	39
4.6	Výstupné rozhrania.....	41
4.7	Knižnica libSOM.....	41
5	Experimenty.....	42
5.1	Prieskum vlastností modelu STS.....	42
5.2	Organizácia pSTS a PMC.....	46
5.3	Vytvorenie perspektívne invariantných reprezentácií v module aSTS.....	51
5.4	Testy modelu CRG.....	53
6	Diskusia.....	56
	Záver.....	58
	Použitá literatúra.....	59
	Príloha A – CD.....	63

Úvod

Problematika systému zrkadliacich neurónov a dizajn komplexného a biologicky plauzibilného výpočtového modelu ostáva stále otvorenou otázkou už vyše desaťrošie. Ako zástancovia hypotézy o prispievaní systému zrkadliacich neurónov k porozumeniu akciám sa domnievame, že takýto systém významne prispeje k súčasnej kognitívnej robotike a bude nevyhnutnou súčasťou modulu porozumenia akciám pre umelé životné formy.

V súčasnosti vyvinulo niekoľko výskumných tímov modely [1] založené na rôznych princípoch s rôznym stupňom abstrakcie a vhodným predspracovaním dát. Tieto modely sme pozorne preštudovali, zanalyzovali a izolovali niekoľko problematických miest, ktoré súčasné modely majú. Jedným z nich je nedostatočne komplexný model vrchnej spánkovej brázdy, ktorá hrá jednu z hlavných úloh v celom zrkadliacom systéme.

Inšpirovaní neurologickými výskumami v oblasti senzomotoriky sme navrhli vlastný model zameraný práve na vznik reprezentácií vo vrchnej spánkovej brázde a jej prepojenie s premotorickou kôrou. Výsledkom celého návrhu je modulárna architektúra pozostávajúca zo samostatných blokov, ktoré reprezentujú neurálne oblasti a jej veľkou výhodou je možnosť online učenia celej štruktúry, čo veľmi dobre simuluje bežné každodenné používanie tohto okruhu. Okrem toho modularita dovoľuje dodatočné rozšírenie modelu o nové súčasti a zabezpečuje jeho budúci vývoj. V rámci snahy o čo najširší model sme pridali aj exekutívny blok, ktorý je práve zdrojom informácií pre premotorickú kôru ako aj pre vrchnú spánkovú brázdu a ten sme experimentálne testovali v reálnom prostredí.

Pomocou série dôsledne pripravených experimentálnych úloh sme skúmali správanie sa modelov a vyhodnocovali ich úspešnosť a naplnenie našich hypotéz ohľadom ich výslednej organizácie. Dosiahli sme úspešné naplnenie našich predpokladov, ale otvorili sa nám aj nové otázky, či vyvstali nové problémy, ktoré v diskusnej časti analyzujeme a prípadne predkladáme riešenia. Vývoj tohto modelu nie je ani zďaleka pri konci (a ako celok nebol ani cieľom tejto práce), a preto predkladáme dizajn, implementáciu a experimentálne výsledky jednotlivých blokov (modulov), z ktorých sa náš model zrkadliacich neurónov skladá.

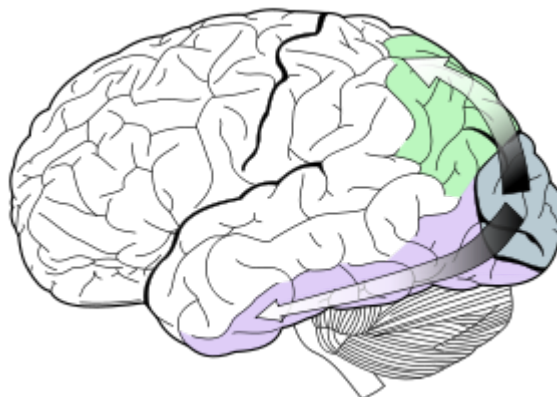
1 Teoretické východiská

V našej práci integrujem dokopy poznatky z neurologických štúdií a výpočtových modelov, pričom sa snažíme čo najviac priblížiť biologicky plauzibilnému riešeniu a vyriešiť niektoré chyby, ktoré sú v súčasných modeloch senzomotorických okruhov, zrkadliacich neurónov a modeloch porozumenia akciám. V nasledujúcich podkapitolách priblížime biologické predpoklady a oblasti mozgu, ktoré vystupujú v našom modeli. Ďalej zanalyzujeme súčasné modely, ktoré nás inšpirovali pre dizajn nášho vlastného modelu.

1.1 Vrchná spánková brázda

Jednou z najskúmanejších oblastí mozgu je v poslednom čase vrchná spánková brázda (superior temporal sulcus - STS). Táto brázda sa tiahne hornou časťou po celej dĺžke temporálnych lalokov. Jej jednotlivé oblasti sú pokladané za sídla rôznych kognitívnych funkcií ako je rozpoznávanie ľudskej tváre, audiovizuálna integrácia, spracovanie reči, sociálna percepcia a spracovanie biologických pohybov.

Podľa hypotézy o vizuálnom spracovaní v dvoch cestách (dorzálna a ventrálna) [2] práve v týchto oblastiach končí cesta ventrálna. Táto hypotéza predpokladá existenciu dvoch samostatných prúdov toku informácie z vizuálnej kôry do vrchných kôrových oblastí a spodných spánkových oblastí. Dorzálna cesta sa zvykne často nazývať aj cestou „KDE“ alebo „AKO“, keďže sa v nej predovšetkým spracúvajú priestorové vzťahy medzi objektmi a motorické oblasti poskytujú informáciu o afordanciách, čiže množinách akcií, ktoré možno s objektom vykonať. Ventrálna cesta dostala pomenovanie „ČO“, ktoré vyjadruje, že na jej konci dochádza k identifikácii a klasifikácii pozorovaného objektu alebo akcie. Práve v tom hrá významnú úlohu práve oblasť STS, ktorá takúto vysoko abstraktnú informáciu, ktorú dostáva z predchádzajúcich oblastí klasifikuje a dalo by sa obrazne povedať, že tam sa rodí znalosť o tom, čo vidíme. Dá sa ale predpokladať, že to nie sú dva separované prúdy spracovania vizuálnej informácie a počas prechodu jednotlivými cestami dochádza medzi nimi ku komunikácii.

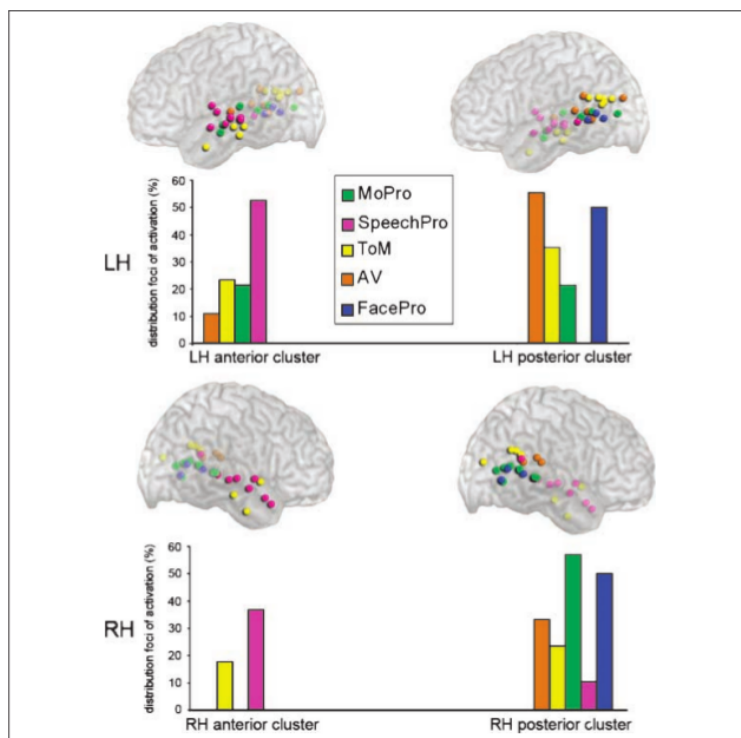


Obr. č. 1: Schéma hypotézy o dvoch prúdoch spracovania vizuálnej informácie – dorzálny prúd je zelený a ventrálly prúd fialový (Prebraté z en.wikipedia.org)

Stále ostáva nevyriešenou otázkou, nakoľko sa samotné STS podieľa na vymenovaných funkciách a nakoľko sa do tejto činnosti zapájajú okolité kôrové štruktúry. Vieme o silnom nepriamom prepojení STS s premotorickou kôrou [3] s čím súvisí aj problematika zrkadliacich neurónov, ktorej venujeme samostatnú podkapitolu. Taktiež je evidencia prepojenia so sluchovou kôrou, mozočkom a samozrejme vizuálnou kôrou. Podľa hypotézy [4] je funkčná špecifikácia STS vždy závislá na súčinnosti s inými mozgovými štruktúrami vyšších rádov, s ktorými má STS spojenie. Iné názory [4] hovoria, že STS je funkčne fragmentované a jeho oblasti samostatne prispievajú k vykonávaniu spomínaných kognitívnych funkcií a teda úloha STS nie je determinovaná súčasťou ostatných oblastí. Prvú hypotézu by podporilo zistenie, že funkčné oblasti STS sa úplne alebo čiastočne prekrývajú a tým pádom slúžia ako celky pre viac možností použitia. Naopak dôkaz o jasnej oddelenosti funkčných fragmentov by podporoval druhú hypotézu.

Podľa štúdie G. Heina a kol. [4], ktorá bola sumárnou prácou vyhodnocujúcou niekoľko desiatok výskumov oblasti STS sa autorom podarilo ukázať diverzifikáciu brázdy v pozdĺžnom smere. Aktivity v prednej (anteriornej) oblasti sa značne líšili od aktivít v zadnej (posteriovej) oblasti vzhľadom na rôzne funkcie, ktoré experimenty skúmali. Rovnako sa prejavila rôzna špecifikácia aj v rámci hemisfér. V ľavej hemisfére sa približne rovnakým podielom na spracovaní biologického pohybu podieľa anteriorná aj posteriorná oblasť STS, pričom v oboch

prípadoch je to len niečo málo cez 20% celkovej aktivity pre rôzne kognitívne funkcie. V pravej hemisfére v posteriornej oblasti naopak spracovanie pohybu dominuje (vyše 55% celkovej aktivity) a v anteriornej oblasti žiadna aktivita nameraná nebola.



Obr. č. 2: Funkčná organizácia oblastí STS v oboch hemisférach. (Prebraté z článku G. Heina a kol. [4])

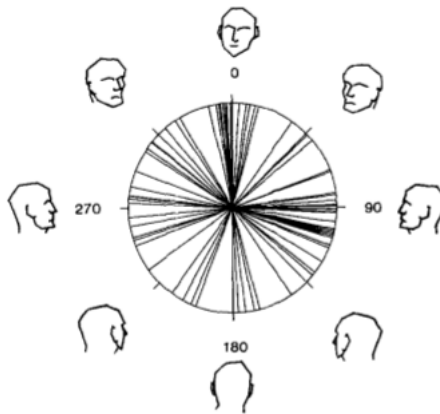
Na obr. č. 2 sú graficky znázornené výsledky prieskumu funkčnej organizácie oblasti STS, v obrázkoch naľavo môžeme vidieť distribúciu aktivít v anteriornej oblasti STS, pričom pre našu prácu je najzaujímavejší zelený stĺpec, ktorý reprezentuje spracovanie pohybu. V pravej časti vidíme distribúciu aktivít posteriornej oblasti STS. Horné dva grafy zodpovedajú ľavej hemisfére, dolné dva pravej hemisfére. Na základe týchto výsledkov autori vyslovujú domnienku, že STS nie je funkčne fragmentované, ale jeho konkrétna funkcia je determinovaná aktivitou okolitých oblastí a teda subregióny STS sú multifunkčné.

Ďalším z pozoruhodných výskumov je práca Perretta a kol. [5], v ktorej sa venujú skúmaniu aktivít oblasti STS práve z hľadiska rozpoznávania tváre u opíc rodu makak. Použitou metódou bolo meranie aktivít jednotlivých neurálnych buniek odpovedajúcich na stimuly. Výskum sa

zameriaval na zistenie, či neuróny v oblasti STS kódujú tvár z pohľadu pozorovateľa, teda, či sú perspektívne variantné, alebo či je kódovanie realizované z hľadiska objektu a neuróny sú perspektívne invariantné. Z hľadiska skúmania spracovania pohybov sa o perspektívne invariantnom kódovaní hovorí ako o kódovaní vzhľadom na cieľ akcie [5].

Inou otázkou bolo, či je aktivita perspektívne variantných neurónov distribuovaná rovnomerne pre každú perspektívu pozorovania, alebo existujú populácie kódujúce niektoré vybrané perspektívy a ostatné perspektívy sú kódované ich spoločnou aktiváciou.

Výsledky meraní ukázali, že až 96% buniek je špecializovaných na kódovanie tváre z pohľadu pozorovateľa a sú perspektívne variantné. Distribúcia kódovaných perspektív však nie je rovnomerná ale tieto neuróny sa zameriavajú na 4 hlavné smery: tvár spredu, tvár z profilu zľava a sprava a „tvár“ zozadu. Najväčší počet buniek reprezentoval tvár spredu a tvár z profilu videnú zľava (viz. obr. č. 3). Malé množstvo buniek sa aktivovalo rovnako pre každú perspektívu a podľa Perrettovej analýzy ich možno považovať za perspektívne invariantné a teda kódujúce tvár z hľadiska objektu. Rovnako veľká skupina buniek vykazovala zmiešané aktivity odpovedajúce na každú perspektívu pozorovania, ale ešte stále sa u nich prejavovala aj diskriminácia vzhľadom na perspektívy.



Obr. č. 3: Distribúcia aktivít neurónov v rámci perspektív

V neskoršej práci skúmal T. Jellema a D. Perrett [6] hlbšie organizáciu oblasti STS v spojitosti s kódovaním pohybov vzhľadom na objekt a s kódovaním vzhľadom na perspektívu. Zamerali sa na prieskum prednej oblasti STS (aSTS) kde zistili prítomnosť značnej populácie neurálnych

buniek reprezentujúcich pohyb invariantne vzhľadom na perspektívu. Na základe tohto zistenia a predošlých výskumov oblasti STS je možné tvrdiť, že sa v jeho zadnej časti (pSTS) nachádzajú prevážne neuróny diskriminujúce perspektívu pri pozorovaní úchopu a až v prednej časti sa nachádzajú abstraktnejšie reprezentácie, ktoré už sú od perspektívy nezávislé a reprezentujú pohyb na vyššej úrovni.

1.2 Systém zrkadliacich neurónov

Významným fenoménom, ktorý je skúmaný vyše dve desaťročia a vyvoláva množstvo polemiky je systém zrkadliacich neurónov. Podľa časti vedcov podieľajúcich sa na skúmaní tejto oblasti má zrkadliaci systém kritický prínos pre porozumenie akciám, imitáciu, vývoj jazyku, príčiny autizmu a iné kognitívne funkcie. Naopak kritici týchto teórií tvrdia, že zrkadliaci systém nezohráva žiadnu úlohu a je iba náhodnou súčasťou neurálnych okruhov.

G. Rizzolatti a kol. v roku 1988 urobil sériu experimentov [7][8] s macaca nemestrina (Makak sviňochvostý), ktorými sa podarilo dokázať existenciu zrkadliacich neurónov u opíc v histochemickej oblasti F5 (ventrálna premotorická kôra). Pri experimente boli opice posadené na stoličku so zafixovanou hlavou a pomocou elektród bola zaznamenávaná aktivita neurónov v oblasti F5. Pred objekt bola umiestnená krabica, do ktorej boli vložené rôzne predmety. Opica ich po zopnutí spínača mohla dosiahnuť a pod nimi sa ukrývala odmena v podobe jedla. Pomocou dvoch infračervených kamier bola zaznamenávaná poloha ruky a všetko sa nahrávalo pre neskoršiu analýzu. V druhej časti experimentu predvádzal experimentátor opiciam podobné pohyby, ako samé vykonávali v prvej časti. Jednalo sa o uchopenia jedla, manipuláciu s jedlom, trhanie, balenie a tiež pohyby, ktoré sa neviazali na žiadny objekt, ako mávanie, či zdvíhanie rúk.

Výsledkami vyššie spomínaného experimentu bola korelácia medzi aktivitou neurónov v oblasti F5 a manipulácia samotnej opice s predmetmi a taktiež keď s týmito predmetmi manipuloval experimentátor. Podobná aktivita sa vôbec neprejavila pri experimentátorových pohyboch, ktoré neboli viazané na žiaden objekt. Tiež sa tieto neuróny neaktivujú pri pozorovaní samotného objektu, alebo pri imitovaní pohybu s objektom, ktorý však chýba.

Na základe týchto zistení sa dávali zrkadliace neuróny do súvisu s rozpoznávaním motorických

akcií. V neskorších experimentoch na opiciach boli objavené ďalšie oblasti, ktoré taktiež obsahovali zrkadliace neuróny a spolu tvoria parieto-frontálny mozgový okruh. Ten pozostáva z oblasti F5, dolného temenného lalôčika (IPL) a prednej medzitemennej oblasti (AIP). Obe oblasti dostávajú tiež vizuálne informácie vyššej úrovne od oblasti v hornej spánkovej brázde (STS). Tieto oblasti kódujú biologické pohyby, avšak nereagujú na motorické podnety. Preto, striktne vzaté, sa nepovažujú za súčasť zrkadliaceho systému. Zrkadliacu vlastnosť vykazujú tiež neuróny oblasti F6, bočnej medzitemennej oblasti (LIP) a prednej medzitemennej oblasti (VIP) [9][10].

Neurálne predpoklady existencie zrkadliaceho systému o človeka boli už v polovici 20. storočia. Naznačovali to výskumy Cohena-Seata a kol. [11] a tiež Gastauta a Breta [12]. Dôkaz však bol podaný až nedávno vďaka vizualizačným technológiám zobrazujúcich mozog ako funkčná magnetická rezonancia (fMRI), elektroencefalografia (EEG) a magnetoencefalografia (MEG).

Štúdie [9] ukázali, že podobne ako u opíc, je zrkadliaci systém človeka formovaný prednou oblasťou IPL, dolnou časťou predústredného závitú a dolnou časťou dolného čelového závitú. Tiež sa občas pozorovala aktivita neurónov premotorickej mozgovej kôry a vrchného temenného lalôčika. Analýza výskumu pomocou fMRI na jednotlivých objektoch ukázala, že aktívne sú aj ďalšie oblasti kôry (napr. primárna a sekundárna somatosenzorická kôra). Dá sa predpokladať, že tieto oblasti sú spúšťané zrkadliacimi okruhmi a poskytujú dodatočné mechanizmy kognitívnym procesom (napr. použitie vnútorného modelu).

Vďaka posledným výskumom [10] pomocou fMRI bol podaný dôkaz, že parieto-frontálny zrkadliaci okruh by mohol zachytávať cieľ pozorovanej motorickej akcie. Probandi pozorovali videozáznam uchopenia predmetu ľudskou a robotickou rukou. Napriek rozdielom v tvare a spôsobe pohybu bol vždy aktivovaný zrkadliaci okruh probanda. Iný výskum sa zaoberal aktiváciou zrkadliacich okruhov pri počutí rôznych zvukových podnetov. Pri počutí charakteristických zvukov nástroja ovládaného rukou sa opäť tieto okruhy aktivovali.

Práve všetky spomínané vlastnosti takéhoto mechanizmu viedli k teórii, ktorá predpokladá, že zrkadliaci systém je priamo zodpovedný za pochopenie cieľov akcií, ktoré konajú iní ľudia. Tým, že vidíme druhých konať určité pohyby, tak sa nám aktivujú oblasti, ktoré používame pri

plánovaní pozorovaných pohybov. Pomocou toho dokážeme jednoducho predikovať výsledok bez potreby zložitejšieho odvodzovania. Túto teóriu podporili dve štúdie na opiciach. V prvom prípade opice počuli im známi zvuk, avšak nevideli žiadnu akciu. V druhom prípade bol za plachtou skrytý objekt a experimentátor sa za ním načiahol, no samotné uchopenie už opica nevidela. V oboch prípadoch sa oblasť F5 aktivovala, akoby sama opica vykonávala/pozorovala danú akciu.

Ukazuje sa teda, že by mohlo ísť u ľudí o všeobecnejší a abstraktnejší mechanizmus, ktorý napomáha porozumeniu a imitácii rôznych akcií. Tie dokonca nemusia byť vykonávané ani ľudským efektorom.

Významnou mierou z nášho hľadiska prispel k pochopeniu systému zrkadliacich neurónov výskum V. Caggiana a kol. [13], v ktorom sa zamerali na podobný experiment ako D. Perrett a kol [5] a skúmali aktivitu zrkadliacich neurónov v premotorickej kôre a ich diskrimináciu perspektívy. Výsledkom bolo, že až 52% skúmaných neurónov vykazovala zrkadliacu vlastnosť. Pri vyhodnotení aktivácií so závislosťou na perspektíve pozorovanej akcie zistili, že 74% neurónov bolo perspektívne závislých a zvyšných 26% páliło vždy, bez ohľadu na perspektívu. Perspektívne závislé populácie sa aktivitou delili na približne tretiny a páliło pre 0°, 90° a 180°.

V diskusii V. Caggiano vyslovuje domnienku, že by mohlo ísť o zámenu rozdelenie aktivít pri pozorovaní pohybu podľa vzdialenosti, nie podľa perspektív [13]. Ale hneď na to zdôrazňuje, že v ich výsledkoch boli identifikované tri rôzne populácie nie iba dve, ako by sa dalo predpokladať pri diskriminácii vzhľadom na vzdialenosť (kedy sa aktivujú neuróny pre akciu v osobnom priestore a mimoosobnom priestore).

Na základe tohto posledného výskumu vidíme, že takéto usporiadanie zrkadliacich neurónov premotorickej kôry je veľmi podobné ako usporiadanie neurónov v oblasti STS a obe oblasti tesne interagujú cez sprostredkujúce oblasti PF a AIP.

1.3 Výpočtové modely

Tvorbe výpočtových modelov zameraných na uchopovanie či zrkadliace neuróny sa v súčasnosti venuje niekoľko tímov po celom svete. V nasledujúcich dvoch podkapitolách predstavíme vybrané modely, ktoré predstavovali základ našich modelov uchopovania a modelu zrkadliacich neurónov.

Veľmi inšpirujúcimi v oblasti zrkadliacich neurónov boli pre nás modely Tessitoreho a kol. [14] a Oztopa a kol. [18], rovnako sme preštudovali aj iné jednoduchšie modely [15][16][17].

Pri ich analýze sme zistili, že mnohé modely zrkadliacich neurónov predpokladajú, že majú k dispozícii vizuálny vstup perspektívne invariantný a tak sa vyhýbajú riešeniu problému s pozorovaním akcií z určitej perspektívy. Tento problém sa pokúšal preklenúť už Tessitore a kol. vo svojom modeli [14] a rovnako je jeho riešenie aj našim cieľom.

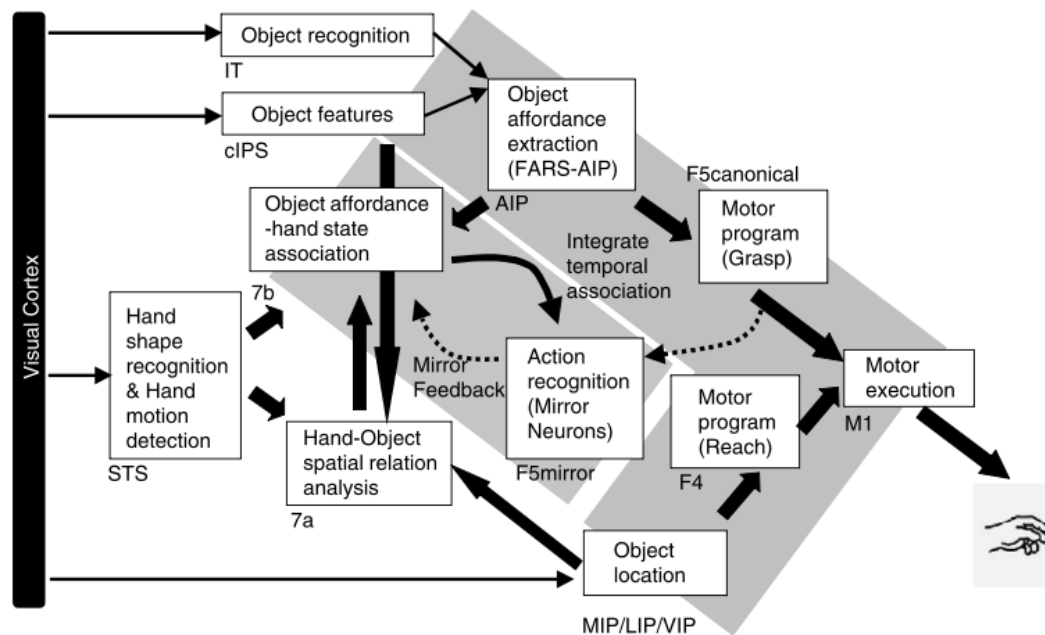
1.3.1 Model MNS

Model MNS je biologicky inšpirovaný a vychádza z vývoja zrkadliaceho systému v detstve. Podľa hypotézy autorov Oztopa a Arbiba [18] sa zrkadliace neuróny tvoria u dieťaťa v jeho rannom veku, kedy samovoľne stimuluje motorickú kôru rôznymi pohybmi.

Model MNS simuluje zrkadliaci systém opíc, pričom sa zameriava na rozvoj zrkadliacich neurónov seba pozorovaním. Jednotlivé komponenty MNS by mali zodpovedať oblastiam tvoriacim zrkadliaci systém opíc. Jadrom bol zrkadliaci okruh pozostávajúci z oblastí 7b a F5 (viď. Obr. č. 4)

Komponenty MNS rozpoznávali z vizuálneho vstupu pohyb a tvar robotickej ruky, ktorú MNS ovládal. Následne analyzoval jej priestorové vzťahy voči objektu, z čoho extrahoval stav, v akom sa nachádza (napr. relatívne súradnice, uhly apod.). Takto sa zabezpečilo, že sa systém mohol sám pozorovať a učiť sa. Vďaka invariancii získaného stavu mohol byť považovaný rovnako za stav pozorovaný z inej perspektívy.

Vstupom komponentu reprezentujúceho oblasť F5 bola teda trajektória ruky. Pre učenie zrkadliacich neurónov bol použitý výstup (motorické akcie) kánonických neurónov komponentu F5 ako učiaci signál.



Obr. č. 4: Schmatický náčrt prepojenia komponentov modelu MNS

(Prebraté z Oztop a kol. [18])

Zrkadliaci okruh (7b-F5) bol implementovaný doprednou neurónovou sieťou. Vstupnú vrstvu predstavovala oblasť 7b, výstupnú F5 a medzi nimi sa nachádzala 1 skrytá vrstva. Model zaznamenal značné úspechy ohľadom predikcie trajektórií ruky zrkadliacim systémom, pričom si dokázal hlavne poradiť aj s novými typmi uchopenia, ktoré sa nenachádzali v tréningovej množine. Viac detailov sa dá nájsť v práci Oztopa a Arbiba [18].

Určite stojí za zmienku, že tento model bol vylepšený na model MNS2 [19], pridaním rekurentnej siete. Tá rozšírila kapacitu modelu a mohol sa zamerať na simulovanie audio-vizuálnych zrkadliacich neurónov alebo situácie, kedy bol cieľový objekt aktuálne skrytý.

1.3.2 Modely uchopovania

Schopnosť interagovať s prostredím je jednou zo základných predispozícií pre úspešný živý a aj umelý organizmus. Preto sa v súčasnosti venuje veľká pozornosť riešeniu tejto problematiky a existuje mnoho prístupov a funkčných modelov dosiahnutia a uchopenia objektu [20] [21] [22].

Nepochybne veľmi komplexným modelom je model D. Krafťa a kol. [20], kde do samotného modelu dosahovania a uchopovania predmetov zahrnul aj niekoľko druhov pamäte. Prvou je ikonická pamäť, ktorá by sa dala chápať ako vizuálny vstup. Do vizuálno-priestorového skicára sa ukladá cieľový objekt a buduje sa jeho interná reprezentácia. Po vykonaní úchopu sa do epizodickej pamäte ukladá záznam o kvalite úchopu. Nakoniec sa buduje objektová a úchopová pamäť a v úchopovej pamäti vznikajú na základe skúsenosti afordancie. Učenie modelu pozostáva z troch štádií, ktoré sú blízke štádiám dieťaťa. V prvej si model vytvára priestorovú predstavu o tvare a veľkosti objektu a pokúša sa robiť náhodné úchopy. O ich úspešnosti si práve v epizodickej pamäti vedie záznam. V druhom štádiu, ktoré Kraft nazýva hraním si model skúša rôzne uchopenia a buduje už množinu afordancií, na základe ktorej vie vytvoriť odhad pre najlepšiu pozíciu ruky. Nakoniec v tretej fáze dochádza k vytváraniu plánov, ktorých cieľom je uchopiť objekt optimálne (typickým príkladom môže byť šálka, ktorú by tento model uchopil za ucho).

Takto nadizajnovaný model dosahoval veľmi dobré výsledky aj pre rôzne netriviálne úchopy objektov (ako napr. spomínaná šálka, alebo košík, ktoré sa naučil uchopovať za uško, rúčku) a proces jeho učenia sa dal prirovnať k procesu učenia malých detí [23], ktoré sa postupne učia dosiahnuť na predmet a najskôr ho uchopiť jednoduchým typom úchopu (napr. zhora, od boku) a až postupom času sa buduje množina afordancií a jemná motorika, ktorá umožňuje naučenie zložitejších úchopov.

2 Použité formálne metódy

V tejto kapitole priblížime čitateľovi použité formálne nástroje, ktoré tvoria stavebné bloky našich modelov. V súlade s konekcionistickými názormi považujeme myseľ a kognitívne schopnosti človeka (aj zvierat) za emergentný jav, ktorý je výsledkom správania sa systému jednoduchých výpočtových jednotiek – neurónov. Práve systém neurónov a ich spojení obsahuje distribuovanú informáciu, ktorá je paralelne spracovávaná. O túto tézu sa opiera subsymbolový prístup [25] k umelej inteligencii, ktorého hlavným nástrojom sú umelé neurónové siete.

2.1 Umelá neurónová sieť

Umelá neurónová sieť (ďalej budeme používať skrátený výraz neurónová sieť) je výpočtový model inšpirovaný biologickými neurónovými sieťami. Z matematického hľadiska môžeme považovať neurónovú sieť za orientovaný graf (obr. xx), kde vrcholy predstavujú neuróny a hrany sú ich spojeniami. Každý neurón má svoj vstupný vektor, pri čom každá zložka vektora je prenasobená reálnym číslom (váhou). Takto preváňované vstupy sú následne zosumované a výsledok sumy je argumentom aktivačnej funkcie neurónu, ktorá musí spĺňať podmienku diferencovateľnosti. Aktivačná funkcia reprezentuje frekvenciu akčných potenciálov excitovaného neurónu. Najjednoduchšou aktivačnou funkciou je binárna aktivačná funkcia, ktorá nadobúda dve hodnoty (0 = stav v pokoji, 1 = excitovaný stav). Omnoho častejšie používané sú logistická (sigmoidálna) funkcia a hyperbolický tangens.

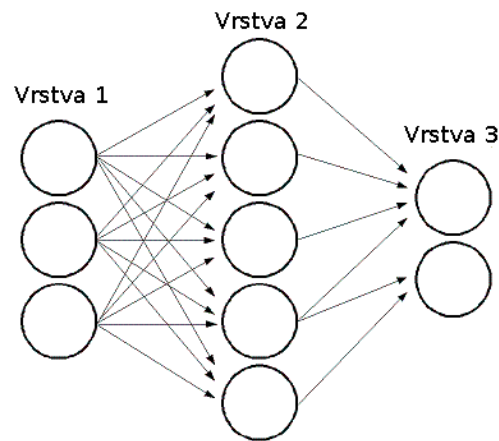
$$net = f\left(\sum_i w_i \cdot x_i\right) \quad (2.1)$$

Rovnica 2.1 vyjadruje matematický zápis veľkosti aktivácie *net* jedného neurónu, kde f_a je aktivačná funkcia, x_i je *i*-ta zložka vstupného vektora a w_i je *i*-ta váha vstupného signálu x_i .

S návrhom neurónovej siete (perceptrón) prišiel F. Rosenblatt [24], keď použil jeden neurón s binárnou aktivačnou funkciou schopný klasifikovať dátovú množinu do dvoch tried. Takto ukázal, že neurónová sieť pozostávajúca z jedného neurónu sa správa ako klasifikátor a je schopná klasifikovať lineárne separovateľné množiny. Kvôli neschopnosti klasifikovať lineárne neseperovateľné množiny bolo nutné ideu perceptrónu rozšíriť a vytvoriť viacvrstvovú sieť.

2.2 Dopredná neurónová sieť

Trieda sietí s dopredným šírením zahŕňa jedno aj viacvrstvové typy neurónových sietí, kde sú jednotlivé vrstvy pospájané iba v jednom smere (od vstupu po výstup) a tak tvoria orientovaný acyklický graf. Takáto sieť dokáže riešiť lineárne neseparovateľné problémy, dokáže sa správať ako funkčný aproximátor a asociátor [25]. Každá vrstva obsahuje niekoľko neurónov a vstupom pre vyššiu vrstvu sú výstupy neurónov predchádzajúcej vrstvy, ktoré vypočítame pomocou rovnice 2.1. Informácia v takejto sieti je distribuovaná v jednotlivých spojeniach medzi vrstvami.



Obr. č. 5: Schéma viacvrstvej neurónovej siete

Dnes už tradičným spôsobom učenia viacvrstvových neurónových sietí je algoritmus spätného šírenia chyby (BackPropagation) [26], ktorý patrí do triedy algoritmov učenia s učiteľom. Preto sa však nepokladá za príveľmi biologicky plauzibilný.

Algoritmus GeneRec je ďalším z algoritmov vhodných na tréning viacvrstvových sietí a podľa autora O'Railleyho [27] je bližší biologickému spôsobu učenia (hoci by sme ho tiež mohli zaradiť do triedy učenia s učiteľom), keďže sa v ňom nepočíta žiadna chyba, ale spojenia sú modifikované pri prechode informácie oboma smermi.

2.3 Samoorganizujúce sa mapy

Spoločnou charakteristikou tejto celej triedy neurónových sietí je ich schopnosť extrahovať príznaky z dátovej množiny pomocou učenia bez učiteľa (ako vyplýva z pomenovania - samoorganizáciou). Sú založené na pozorovaní, ktoré vyslovil už Turing [29] v roku 1952: „Všeobecný poriadok môže vzniknúť z lokálnych interakcií“. Z tohto môžeme odvodiť tri hlavné princípy, ktoré sa týkajú samoorganizácie a samoorganizujúcich sa máp. [25]

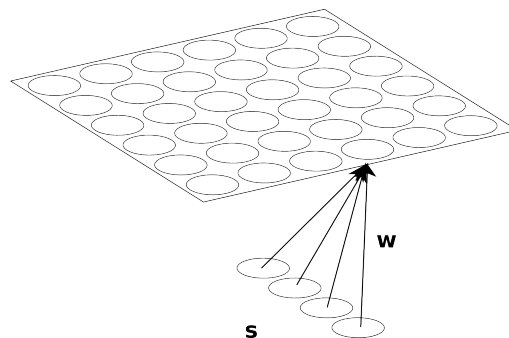
Princíp č.1: Modifikácie v synaptických spojeniach vedú k samozosilneniu.

Princíp č.2: Obmedzenosť zdrojov (vstupných vzorov) vedie k súťaživosti.

Princíp č.3: Modifikácie v synaptických spojeniach vedú ku spolupráci.

2.3.1 SOM

Samoorganizujúca sa mapa (SOM – Self-Organizing Map) je základným predstaviteľom triedy sietí, ktorým sa venujeme v tejto podkapitole. Navrhol ju T. Kohonen [28] a niekedy sa označuje ako Kohonenova mapa. Je tvorená dvojrozmernou maticou neurónov, ktoré majú okrem vstupných spojení aj laterálne interakcia zavádzajúce tri hlavné princípy samoorganizácie.



Obr. č. 6: Schéma architektúry SOM.

Vďaka usporiadaniu neurónov do matice má takáto sieť topológiu a metriku, čo umožňuje merať vzdialenosti medzi neurónmi na matici. To následne umožňuje zavedenie funkcie okolia h_{ib} , ktorá simuluje laterálne interakcie v modeli SOM. Táto funkcia určuje spolu s rýchlosťou

učenia mieru adaptácie váh neurónu, pričom zahŕňa jeho vzdialenosť v sieti od neurónu, ktorého váhy najlepšie aproximujú vstupný vzor. Častou sa využíva gaussovská funkcia okolia:

$$h_{ib} = e^{-\frac{d(i,b)^2}{\sigma^2}} \quad (2.2)$$

Kde index i značí index neurónu, pre ktorý počítame vzdialenosť, index b značí neurón, ktorý je najlepšie aproximuje vstupný vzor. Člen $d(i,b)$ je euklidovská vzdialenosť neurónov i a b na matici neurónov siete SOM a σ je šírkou gausiánu.

Kvôli stabilizácii siete a jej skonvergovaníu do konečného stavu sa rýchlosť učenia a šírka funkcie okolia postupom epôch znižujú lineárnou alebo exponenciálnou mierou, čím dochádza na konci k vytvoreniu klastrov neurónov, ktoré reagujú na podobný vstupný vzor.

Učiaci algoritmus siete SOM môžeme popísať v nasledujúcich krokoch:

Krok 1: Sieti je prezentovaný vstupný vektor s .

Krok 2: Vypočítame vzdialenosť váhového vektora w každého neurónu od vstupného vektora s :

$$d_i = \|s - w_i\|^2 \quad (2.3)$$

Krok 3: Nájďme neurón, ktorého vzdialenosť d je najmenšia a nazveme ho „vít'azom“

$$b = \arg \min_i \{d_i\} \quad (2.4)$$

Krok 4: Aktualizujeme vstupné váhy všetkých neurónov podľa nasledujúceho pravidla:

$$\Delta w_i = \gamma \cdot h_{ib}(\sigma) \cdot (s - w_i) \quad (2.5)$$

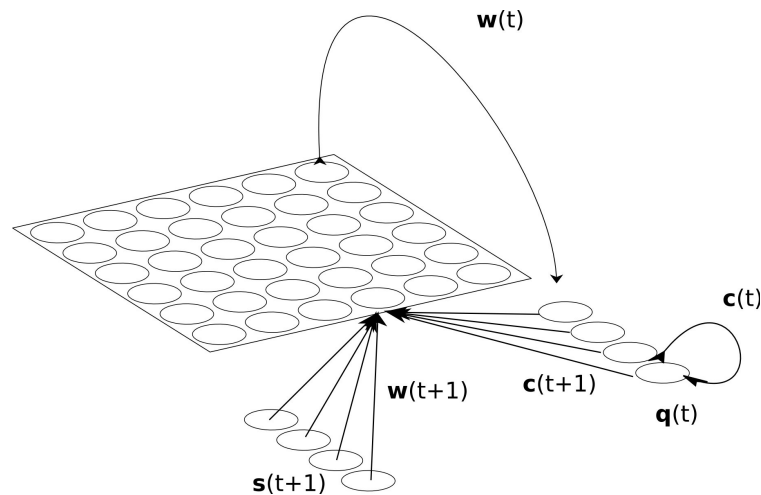
Krok 5: Znížime parametre šírky funkcie okolia σ a rýchlosti učenia γ .

Z rovníc samotného učiaceho algoritmu je možné vidieť, že sieť SOM je schopná spracovávať iba statické dáta bez časovej závislosti. Ak by sme chceli SOM rozšíriť na spracovanie sekvenčných dáta, je nutné pridať sieti pamäť realizovanú ako kontextovú vrstvu, ktorá je rekurentne prepojená s výstupnou vrstvou siete SOM.

Existuje niekoľko realizácií rekurentných SOM, z ktorých sme vybrali dve, ktoré sa pre naše potreby najviac hodili a ich detailný popis uvádzame v nasledujúcich podkapitolách.

2.3.2 Zlučovacia SOM (Merge SOM)

Je podtriedou samoorganizujúcich sa máp navrhnutou B. Hammer a M. Strikertom [30], ktoré spracúvajú sekvenčné dáta. Z tohto dôvodu je nutné do tradičnej siete SOM zaviesť pamäť. V tomto modeli je pamäť realizovaná ako kontextový vektor, s rovnakou dimenziou akú má vstupný vektor. Alternatívnym prístupom by mohlo byť kopírovanie aktivít celej mapy (viď. 2.3.3 RecSOM), no oproti tomu je MSOM pamäťovo aj časovo efektívnejším modelom. Schematický náčrt modelu môžeme vidieť na obr. č. 7.



Obr. č. 7: Schéma architektúry MSOM.

Učenie MSOM môžeme zhrnúť do nasledujúcich krokov:

Krok 1: Sieti je prezentovaný vstupný vektor $s(t)$

Krok 2: Vypočíta sa rekurzívna vzdialenosť každého neurónu od vstupného vektora podľa rovnice 2.6:

$$d_i(t) = (1 - \alpha) \cdot \|s(t) - w_i(t)\|^2 + \alpha \cdot \|q(t) - c(t)\|^2 \quad (2.6)$$

Krok 3: Spomedzi všetkých neurónov sa nájde taký, ktorého vzdialenosť je najmenšia – víťaz

$$b(t) = \arg \min_i \{d_i(t)\} \quad (2.7)$$

Krok 4: Vstupné váhy všetkých neurónov sú aktualizované nasledujúcou formou Hebbovského pravidla:

$$\Delta w_i(t) = \gamma \cdot h_{ib}(\sigma) \cdot (s(t) - w_i(t)) \quad (2.8)$$

A kontextové váhy všetkých neurónov podľa analogického pravidla:

$$\Delta c_i(t) = \gamma \cdot h_{ib}(\sigma) \cdot (q(t) - c_i(t)) \quad (2.9)$$

Krok 5: Aktualizujeme kontextový vektor ako lineárnu kombináciu vstupných a kontextových váh víťaza:

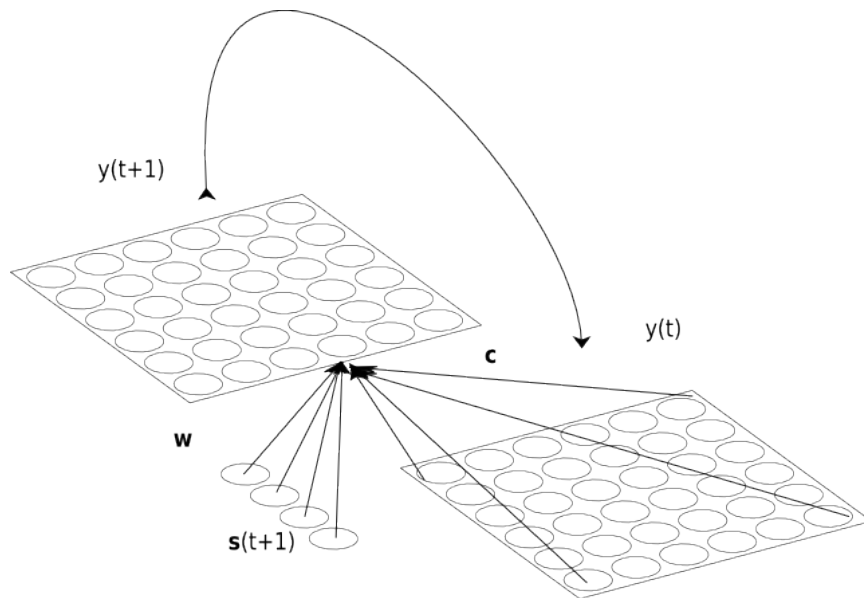
$$q(t+1) = (1 - \beta) \cdot w_i(t) + \beta \cdot c_i(t) \quad (2.10)$$

Krok 6: Znížime parametre šírky funkcie okolia σ a rýchlosti učenia γ .

Môžeme si všimnúť, že v uvedených rovniciach vystupujú tri parametre α , β , γ . Parameter α riadi výpočet rekurzívnej vzdialenosti a určuje nakoľko sa bude prihliadať na aktuálny vstup a nakoľko na predchádzajúce vstupy. Parameter β ovláda aktualizáciu kontextového vektora a určuje aká časť sa zoberie z aktuálnych váh víťaza a aká časť z kontextových váh víťaza. Parameter predstavuje rýchlosť učenia, teda ako rýchlo sa adaptujú váhy Hebbovským pravidlom. Všetky tri parametre sú na počiatku inicializované na hodnoty z intervalu $\langle 0, 1 \rangle$, pričom rýchlosť učenia môže byť potencionálne aj vyššia, avšak nie je to zaužívaný prístup. Od nastavenia parametrov α a β závisí ako sa bude sieť organizovať na tréningovej množine. Vhodné hodnoty parametrov nie je možné nijak matematicky vypočítať, a preto je nutné ich empiricky zistiť, najlepšie dávkovým prieskumom roviny nimi definovanej.

2.3.3 RecSOM

Ďalšou z rekurentných sietí SOM je RecSOM [31], ktorá má namiesto kontextového vektora kontextovú vrstvu a tá je identickou kópiou výstupnej vrstvy mapy. Takto sa v modeli RecSOM uchováva celá informácia o minulosti. Analýzou [32] bolo ukázané, že model dokáže indukovať bohatšie spektrum dynamického správania počas učenia a taktiež, že dynamika RecSOM vedie vo väčšine prípadov k markovovskému správaniu, teda neuróny majú pamäť časovo ohraničenú [33].



Obr. č. 8: Schéma modelu RecSOM.

Algoritmus učenia siete RecSOM je veľmi podobný ako pri modely MSOM a prebieha v nasledujúcich krokoch:

Krok 1: Sieti je prezentovaný vstupný vektor $s(t)$

Krok 2: Vypočíta sa rekurzívna vzdialenosť každého neurónu od vstupného vektora podľa rovnice 2.11:

$$d_i(t) = \alpha \cdot \|s(t) - w_i(t)\|^2 + \beta \cdot \|q(t) - c(t)\|^2 \quad (2.11)$$

Krok 3: Spomedzi všetkých neurónov sa nájde taký, ktorého vzdialenosť je najmenšia – víťaz

$$b(t) = \arg \min_i \{d_i(t)\} \quad (2.12)$$

Krok 4: Vstupné váhy všetkých neurónov sú aktualizované nasledujúcou formou Hebbovského pravidla:

$$\Delta w_i(t) = \gamma \cdot h_{ib}(\sigma) \cdot (s(t) - w_i(t)) \quad (2.13)$$

Krok 5: Aktualizujeme kontextovú vrstvu prekopírovaním aktivity výstupnej vrstvy, ktorú vypočítame podľa rovnice 2.14:

$$y_i(t) = e^{-d_i(t)} \quad (2.14)$$

Krok 6: Znížime parametre šírky funkcie okolia σ a rýchlosti učenia γ .

Správanie siete je riadené dvoma parametrami α , β , ktoré určujú v akom pomere sa bude vypočítavať rekurzívna vzdialenosť každého neurónu od vstupného vzoru. Oproti MSOM sú tieto parametre nezávislé a vyberáme ich z intervalu $\langle 0,1 \rangle$. Zistenie vhodnej konfigurácie ostáva na empirických testoch na konkrétnej dátovej množine.

2.4 Učenie s posilňovaním

Učenie s posilňovaním predstavuje jednu z metód navrhnutých pre Markovove rozhodovacie procesy (MRP). MRP možno definovať ako n-ticu (S, A, T, R, γ) [34], kde S je stavový priestor, A je priestor akcií, T je funkcia prechodov medzi stavmi, ktorá vyjadruje pravdepodobnosť prechodu zo stavu s do stavu s' pri vykonaní akcie a . R je funkcia, ktorá vracia odmenu pre prechod zo stavu s do stavu s' pri vykonaní akcie a . Posledným členom n-tice je γ , ktorá vyjadruje faktor poklesu hodnoty budúcich získaných odmien (tzv. faktor zľavy).

Problémom MRP môže byť nájdenie optimálnej rozhodovacej politiky reprezentovanej pravdepodobnostnou funkciou $\pi : S \times A \rightarrow [0,1]$.

Riešenie je možné nájsť pomocou Bellmanových rovníc [34] avšak toto vyžaduje pre jednoduché stavové priestory a malý počet akcií veľmi dlhý čas výpočtu. Keby sme mali mnohodomenzionálny stavový priestor a veľkú množinu akcií, či spojitý stavový priestor a spojité akcie, tak by sa riešenie pomocou Bellmanových rovníc dalo iba veľmi obtiažne nájsť (za nesmierne dlhý čas).

Preto sa učenia s posilňovaním sústreďuje na hľadanie aproximácie optimálnych riešení:

- Aproximácia modelu: algoritmus sa učí funkcie T a R za predpokladu, že S , A a γ sú známe.
- Aproximácia ohodnotenia stavov: algoritmus sa učí funkciu $V : S \rightarrow \mathbb{R}$ alebo funkciu ohodnotenia akcií $Q : S \times A \rightarrow \mathbb{R}$
- Aproximácie rozhodovacej politiky: algoritmus sa učí politiku π

2.4.1 Algoritmus učenia v spojitom priestore

Pri rozšírení úlohy na spojitý priestor sa objavuje problém s nekonečným počtom stavov a nekonečnou množinou akcií, ktoré môže agent vykonať. Algoritmus, ktorý sa dokáže s týmto vysporiadať navrhol H. von Hasselt [34] a nazval ho Continuous Actor-Critic Learning Automaton (CACL). Ide o pomerne nový a menej používaný algoritmus, preto uvádzame aj jeho zápis v pseudokóde s krátkym vysvetlením:

```

1: Inicializuj  $\theta_0$  (ďalej  $V_t(s) = V(s, \theta_t)$ ),  $\psi_0$ ,  $s_0$ .
2: for  $t \in \{0, 1, 2, \dots\}$  do
3:   Vyber akciu  $a_t \sim \pi(s_t, \psi_t)$ 
4:   Vykonaj  $a_t$ , pozoruj  $r_{t+1}$  and  $s_{t+1}$ 
5:    $\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)$ 
6:    $\theta_{t+1} = \theta_t + \alpha_t(s_t) \delta_t \nabla_{\theta} V_t(s_t)$ 
7:   if  $\delta_t > 0$  then
8:      $\psi_{t+1} = \psi_t + \beta_t(s_t) (a_t - Ac(s_t, \psi_t)) \nabla_{\psi} Ac(s_t, \psi_t)$ 
9:   if  $s_{t+1}$  je konečný then
10:    Reinitializuj  $s_{t+1}$ 

```

V algoritme CACL sa kritik snaží aproximovať funkciu ohodnotenia stavu, ktorá prichádza z prostredia a predikovať ju do budúcnosti ($V_t(s_t)$). Aktér generuje výstupný vektor, ktorý je prostredím interpretovaný ako akcia. V prvom kroku dochádza k inicializácii parametrov kritika (θ), a aktéra (ψ), čo sú v našom prípade váhy neurónových sietí, a tiež samotného stavu prostredia (s) (riadok 1). V nasledujúcom kroku aktér generuje akciu (a), ktorá sa ešte zašumuje s gaussovským rozložením pravdepodobnosti (riadok 3,4). Tento proces sa nazýva explorácia a zabezpečuje vykonávanie nových akcií, ktoré sa môžu ukázať ako vhodnejšie oproti predošlým vykonaným v rovnakom stave. Po vykonaní akcie prichádza z prostredia odmena (r) a stav prostredia (s) sa aktualizuje. V poslednom kroku sa vypočíta chyba (riadok 5) a na základe tejto chyby sa aktualizujú parametre kritika (riadok 6) a pokiaľ je chyba pozitívna, čo znamená, že akcia po explorácii sa ukázala byť vhodnejšia, tak sa aktualizujú aj parametre aktéra (riadok 8).

3 Návrh modelov

V tejto kapitole budú popísané návrhy dvoch hlavných modelov, ktoré sú výsledkom našej práce. Jedná sa o model uchopovania (CRG – Controller for Reaching and Grasping), ktorého úlohou je veľmi naivne modelovať časť motorického okruhu mozgu. Druhým modelom je komplexný model neurálneho okruhu vrchnej temporálnej brázdy (STS – Superior Temporal Sulcus) s jej subregiónmi prednou a zadnou časťou (aSTS – anterior STS a pSTS – posterior STS) prepojený s modelom premotorickej kôry (PMC – PreMotor Cortex). Tento model má ešte prepojovací modul PF, ktorý je v našej práci spomenutý iba okrajovo, avšak hrá hlavnú úlohu v biologickom systéme zrkadliacich neurónov rovnako ako v našom modeli systému zrkadliacich neurónov [35][36].

3.1 Dátové množiny

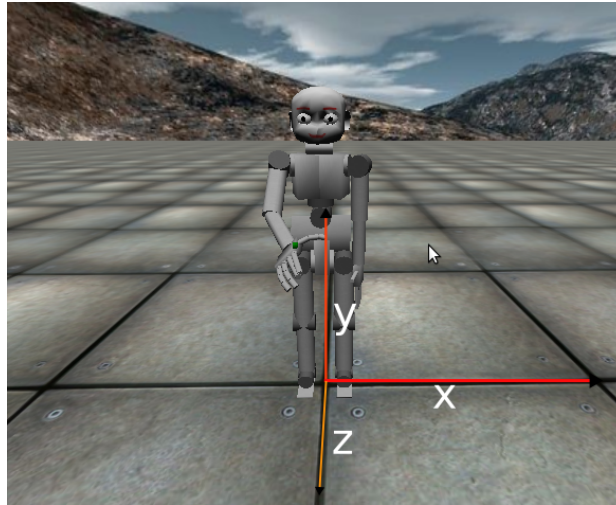
Pre trénovanie modelov sme použili pripravené dáta, ktoré sme získali z reálneho sveta alebo zo simulátora. Tieto dáta možno považovať za výstup z oblastí mozgu, ktoré nie sú predmetom nášho modelovania a v budúcnosti je možné o ne modely vhodným spôsobom rozšíriť a tak sa viac priblížiť k ich úplnosti. Dáta bolo potrebné vždy vhodne predspracovať matematickými metódami a upraviť ich tak, aby boli v prijateľnom formáte pre modely.

Z tohto dôvodu sme nadizajnovali dve rozhrania, ktoré dáta v pôvodnom stave spracovali preškoľovali do požadovaných intervalov, aby mohli slúžiť ako vstup do jednotlivých modulov. Tieto rozhrania detailne popisujeme v nasledujúcich podkapitolách.

3.1.1 Trénovacie dáta pre modely PMC a STS

Dáta pre trénovanie modelu PMC a pSTS boli vygenerované zo simulátora humanoidného robota iCub, ktorý je modelom 2.5-ročného dieťaťa s 53 stupňami voľnosti, dvoma farbenými kamerami simulujúcimi zrak a inými senzormi, ktoré z neho robia vhodnú platformu pre kognitívne experimenty. Počiatok súradnicového systému robota iCub je na podlahe medzi jeho chodidlami s osou X mieriaca v pozitívnom smere doľava, osou Y s pozitívnym smerom hore

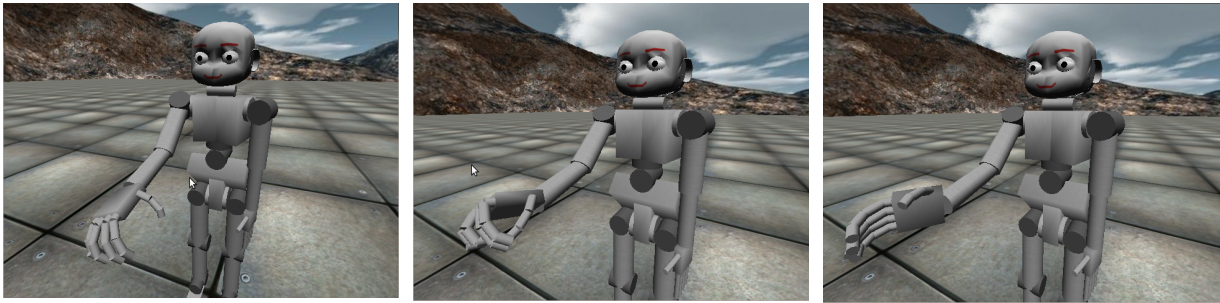
a osou Z mieriacou v pozitívnom smere od iCuba (pre ilustráciu uvádzame obr. č. 9). Dáta modelu aSTS tvorili aktivácie modelov PMC a pSTS, preto sa o nich zmienime iba okrajovo.



Obr. č. 9: zobrazenie súradnicového systému iCuba

Naše experimenty sme zamerali na vytvorenie reprezentácií pre rôzne typy úchopov videné z rôznych perspektív vďaka existujúcemu modelu uchopovania L. Zdechovana [37], ktorý sme použili ako generátor dátovej množiny.

Dáta vygenerované modelom L. Zdechovana pozostávali z troch typov úchopov, ktoré je možné charakterizovať ako úchop zhora, úchop od boku a presný úchop (ilustrované na obr. č. 10). Každý typ úchopu bol zastúpený približne 20 inštanciami, ktoré pozostávali z jednej sekvencie. Každá sekvencia začínala v rovnakej póze a pozostávala z 18 až 24 krokov, ktoré zachytávali stav ruky. Ten bol charakterizovaný ako vektor 16 hodnôt, kde každá zodpovedala hodnote jedného stupňa voľnosti (uhol natočenia kĺbu). Takto vyťažené dáta zo súboru následne prešli do rozhrania, ktoré ich predspracovalo, aby boli použiteľné pre samotný model PMC alebo pSTS.



Obr. č. 10: Príklady troch typov úchopov (zľava: úchop zhora, presný úchop a úchop od boku) zo simulátora robota iCub

Predspracovanie dát sa líšilo pre model PMC a model pSTS. Model PMC potreboval ako vstup proprioceptívne dáta, čo v našom prípade predstavuje práve 16-rozmerný vektor s hodnotami natočenia kĺbov pravej ruky. Z dôvodu inicializácie váh modelu PMC bola každá zložka vektora nezávisle preškáľovaná na požadované hodnoty $\langle -1, 1 \rangle$.

Zložitejšiu vetvu predspracovania predstavuje príprava dátovej množiny pre model pSTS (schematicky zobrazená na obr. č. 11). Východiskom boli dáta s rovnakou štruktúrou, ale stav ruky nie je charakterizovaný 16-rozmerným vektorom hodnôt natočenia kĺbov, ale 60-rozmerným vektorom obsahujúcim súradnice kĺbov a koncov prstov v trojrozmernom priestore.

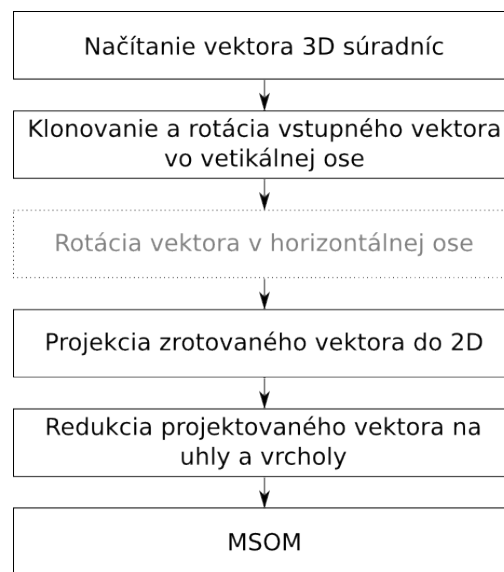
Krok 1: Každý vektor bol zrotovaný okolo osi Y pomocou pre násobenia maticou rotácie o 90° , 180° a 270° . Týmto spôsobom sme z jednej inštancie úchopu vyrobili štyri klony tej istej inštancie pozorované pod štyrmi perspektívami. Z hľadiska správnosti sme mali urobiť aj rotáciu okolo osi X, keďže pozorovateľ vidí úchopy pod určitým uhlom náklonu hlavy, ale tento krok sme zanedbali, keďže pre naše experimenty nemá žiaden dopad.

Krok 2: Urobili sme projekciu zrotovaných vektorov, aby sme tak dosiahli nasimulovanie dopadu lúčov na sietnicu oka. Pri tomto postupe sme využili podobnosť trojuholníkov s predpokladom, že sietnica sa nachádza 5 cm v pozitívnom polpriestore (rozdeleného osou X) v smere osi Z. Týmto sa dáta zredukovali o dimenziu Z a tak sa rozmer vektora zmenšil na 40.

Krok 3: Niektoré trojice bodov bolo možné zredukovať na uhol, ktorý zvierajú a tak sme

nahradili trojicu bodov ukazovák, dlaň, palec jedným uhlom a tiež trojicu lakeť, dlaň, palec. Tým sme opäť dosiahli redukciu rozmeru vektora, v ktorom sme 6 hodnôt súradníc bodov nahradili jedným uhlom. Okrem rýchlosti nemala táto operácia žiaden výrazný dopad na klasifikáciu dát, a preto sme ju neskôr vynchávali kvôli jednoznačnejšej možnosti dáta preškálovať.

Krok 4: Hodnoty vektora sú preškálované do intervalu $\langle -1, 1 \rangle$ škálovaním v dvoch rozmeroch. Každý nepárny element vektora predstavuje x-ovú súradnicu bodu a každý párný y-ovú. Pri tomto predpoklade je preškálovanie priamočiarou operáciou.



Obr. č. 11: Schéma vytvárania dátovej množiny pozorovaných úchopov.

3.1.2 Trénovacie dáta modelu CRG

Na učenie a neskôr otestovanie nášho modelu sme pripravili trénovaciu množinu použitím zariadenia *Microsoft Kinect*, ktoré dokáže vytvárať RGB-D dáta (rozšírené o hĺbku). Táto technológia umožňuje okrem iného zachytenie obrazu spolu s hĺbkou. Pri známej polohe kamery je jednoduché prepočítať tieto údaje na súradnice s počiatkom vzťažnej sústavy v centre kamery.

Vybrali sme štyri objekty s rozličným tvarom a veľkosťou (čajová krabica, krabica so šesťuholníkovou podstavou, vertikálna krabica, plechovka), pri čom dva objekty mali hlavnú os rovnobežnú so stolom a dva os kolmú na rovinu stola. Sekundárna os objektov bola mierne

pootočená, čo vyžadovalo dodatočné natočenia ruky.

Na predspracovanie dát, ktoré nahradilo vizuálnu kôru sme použili knižnicu Point Cloud Library (PCL) [38] a knižnicu Vision for Robotics (V4R), ktorá je vyvíjaná na Univerzite vo Videni.

Predspracovanie prebiehalo v týchto krokoch:

Krok 1: Načítanie súboru formátu pcl, ktorý obsahoval body s ich súradnicami zachytené kamerou *Microsoft Kinect*

Krok 2: Vysegmentovanie objektu zo stola pomocou funkcie knižnice V4R a identifikácia priestorov záujmu, kde každý priestor záujmu predstavuje jeden objekt. Empiricky sme boli schopní zistiť (zobrazením), ktorý priestor záujmu je naším objektom.

Krok 3: Transformácia počiatku súradnicovej sústavy pomocou geometrických transformácií posunu a rotácie keďže súradnice objektov boli v súradnicovej sústave kamery a bolo nutné zjednotiť ich so súradnicovým systémom ramena, ktorého počiatok bol v jeho základni.

Krok 4: Pomocou metódy knižnice PCL sme boli schopní vypočítať ťažisko objektu, ktoré sme považovali za koordináty objektu.

Krok 5: Výpočítali sme kovariančnú maticu z oblaku bodov objektu. Aplikovali sme na ňu analýzu hlavných komponentov (Principal Component Analysis). Jej výsledkom boli tri vlastné vektory a tri vlastné čísla. Pomocou veľkosti vlastných čísiel bolo možné určiť hlavnú os objektu reprezentovanú príslušným vlastným vektorom.

Takýmto spôsobom sme boli schopní z dáta nasnímané zariadením *Microsoft Kinect* transformovať do podoby zrozumiteľnej pre model CRG, ktorý potreboval poznať iba polohu a orientáciu objektu.

3.2 Modul PMC

Modul PMC je modelom premotorickej kôry (F5 v mozgu makaka) a obsahuje motorických reprezentácie úchopov. Keďže dáta majú sekvenčný charakter, modul sme implementovali ako samoorganizujúca sa mapu s rekurentnými spojeniami. Z hľadiska kvality výstupu a výpočtových nárokov bola najoptimálnejšia voľba architektúry MSOM.

3.3 Modul pSTS a aSTS

Modul pSTS sme sa rozhodli implementovať ako sieť architektúry MSOM z viacerých dôvodov. Tým prvým je charakter trénovacích dát, ktoré majú časovú štruktúru a teda bolo nutné použiť niektorú rekurentnú architektúru samoorganizujúcich sa máp. Uskutočnili sme niekoľko testov, ktorých výsledky v práci nepopisujeme, kde sme odskúšali ako implementáciu siete RecSOM a MSOM a porovnávali sme ich výsledné organizácie. Bol predpoklad, že sieť RecSOM, ktorá do kontextovej vrstvy kopíruje aktivitu celej výstupnej vrstvy a tak nedochádza k redukcii informácií môže mať lepšiu výslednú organizáciu ako sieť MSOM, ktorá využíva iba kontextový vektor, ktorý je lineárnou kombináciou súčasných a kontextových váh víťaza. Avšak architektúra MSOM sa ukázala ako úplne dostačujúca a výpočtovo menej náročná oproti architektúre RecSOM, a preto sme v kľúčových experimentoch používali modul pSTS iba s touto sieťou. Za pomoci dávkových testov sme zistili najvhodnejšie parametre siete MSOM (vid'. kap. 5.1.).

Najväčšiu výzvu v našom modeli predstavuje práve vytvorenie modulu, ktorý bude schopný zreprodukovať dáta namerané Perrettom a kol. [5], v ktorých uvádzajú existenciu neurónov invariantných vzhľadom na perspektívu pozorovania a špecializujú sa iba na typ pohybu. V práci [6] Perrett a kol. ukázali, že táto populácia neurónov sa nachádza v prednej časti oblasti STS (aSTS).

Pri dizajnovaní tohto modulu sme zvažovali viacero možných riešení. Najskôr sme uvažovali nad zavedením dodatočných laterálnych interakcií, ktoré budú viesť k požadovanému efektu tzv. pooling (,,zlievanie“ aktivít neurónov do jedného). Podľa experimentálnych výsledkov [5] však dochádza k odozve invariantných neurónov omnoho neskôr, a preto sme sa zamerali na viacvrstvovú architektúru modelu, pričom aSTS predstavuje samostatnú sieť. Za biologicky plauzibilnejšie sme pokladali riešenie, v ktorom bude reprezentácia populačne kódovaná a preto sme si zvolili ako implementáciu modulu aSTS jednu z architektúr samoorganizujúcich sa máp. Uvažovali sme sieť SOM a rovnako aj rekurentné samoorganizujúce sa mapy MSOM a RecSOM. Aby sme boli schopní použiť optimálnu sieť a dosiahnuť zároveň dobré výsledky, v jednom z experimentov (vid'. kap. 5.1.) sme sa zamerali práve na vyhodnotenie vlastností rôznych implementácií modulu aSTS a najlepšie z toho vyšla architektúra SOM.

Vzhľadom na cieľ (vznik perspektívne invariantných neurónov v aSTS) bolo treba najskôr zabezpečiť obrátenie organizácie pSTS opačne. To sme riešili rozšírením vstupu pre modul aSTS, do ktorého nebudú vstupovať iba aktivácie modulu pSTS, ale aj aktivácie modulu PMC. V module pSTS je vždy aktívna cca 1/12 pre daný úchop videný z jednej zo štyroch perspektív. V module PMC je aktívna vždy približne tretina neurónov pre daný úchop čím by mal tento vstup prevážiť nad vstupom z pSTS, ktorý je viazaný na perspektívu a tak by sme mohli dosiahnuť v module aSTS organizáciu podľa typov úchopov a v rámci každej populácie rozdelenie podľa perspektív.

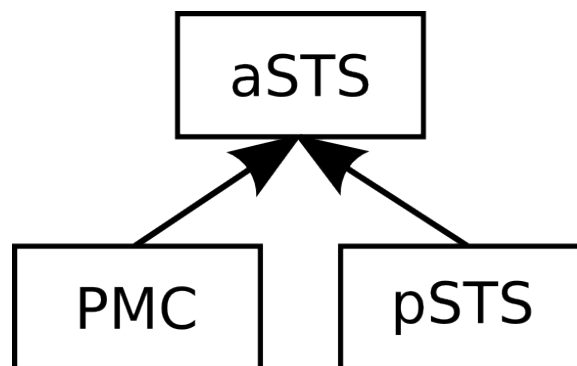
Keďže modul aSTS je implementovaný ako SOM, ktorá spracúva iba statické dáta, tak sme upravili model trénovania celého systému a pozostáva z nasledujúcich krokov:

Krok č. 1: Sieť pSTS dostáva vstup a upraví sa jej váhy

Krok č. 2: Ak sa v dátach zmenil typ úchopu, tak dostáva vstup aj sieť PMC a upraví sa jej váhy, ináč dáva rovnakú aktiváciu ako pre posledný vstup

Krok č. 3: Na konci sekvencie sa zoberie aktivácia modulov pSTS a PMC a spojí sa do jedného vektora.

Krok č. 4: Vektor aktivácií oboch modulov vstupuje do siete aSTS a následne sa adaptujú váhy



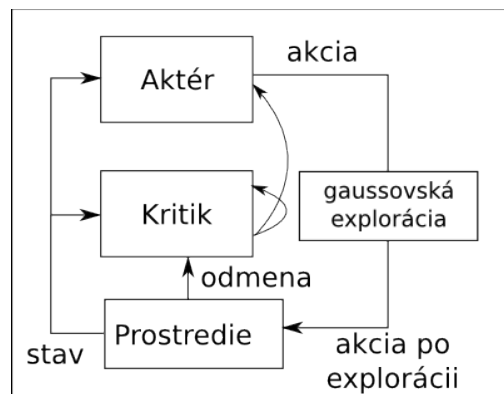
Obr. č. 12: Schéma integrácie modulu aSTS s modulmi PMC a pSTS.

3.4 Model CRG

Problém uchopovania sme rozdelili na dva podproblémy: dosiahnutie objektu a uchopenie objektu. Z toho dôvodu sme vytvorili dva ekvivalentné moduly s rovnakou architektúrou aj učiacim algoritmom avšak s iným rozhraním prostredia. Celý tento blok nazývame modelom CRG (Controller for Reaching and Grasping).

Po analýze komplexity prostredia a stanovení cieľov sme sa rozhodli vo výpočtovom modeli použiť dopredné neurónové siete a učenie s posilňovaním ako biologicky prijateľné riešenie pre našu úlohu. Keďže akcie aj stavy prostredia sú spojité vybrali sme algoritmus spojitého učenia aktéra a kritika (CACLA) [34].

Tento algoritmus beží nad tromi základnými blokmi: aktér, kritik a interaguje s prostredím (detailnú schému možno vidieť na obrázku č. 13).



Obr. č. 13: Architektúra modulu dosiahnutia/uchopenia objektu (model CRG)

Aktér aj kritik boli implementovaní ako dopredné neurónové siete a ich vstupom bol vektor charakterizujúci stav. Za stav bola považovaná proprioceptívna informácia o polohe ramena. V našom prípade išlo o pozíciu ruky v priestore, jej orientáciu vyjadrenú ako kvaternión a otvorenosť prstov ramena. Akciou bola potom zmena polohy v priestore, zmena uhlu natočenia ruky v ose dopredného pohybu ramena, čo umožňovalo pripraviť orientáciu ruky na úchop a nakoniec povel k zatvoreniu prstov. Rameno Katana 450 má iba dva stavy pre prsty a to je otvorené alebo zatvorené (hoci je možné otvorenosť vyjadriť percentuálne, čo sme využili pri určení stavu prostredia).

Podstatným blokom v našom modeli je blok prostredia, ktorý vytváral rozhranie medzi ovládacou API ramena Katana 450 a vnútornými blokmi aktéra a kritika. Blok prostredia nekomunikoval priamo s ramenom avšak využívali sme medzivrstvu s názvom Golem, ktorá poskytovala funkcie na plánovanie a vykonanie presunu ramena do novej pozície a orientácie ruky, čo značne uľahčovalo náš postup.

Ďalšími podstatnými funkciami bloku prostredia sú funkcie odmeny a funkcie vracajúce či je aktuálny stav prostredia fatálny alebo finálny. Funkcia odmeny pre modul dosahovania reflektuje jeho úlohu tj. presun do vhodnej pozície blízko objektu a správne natočenie ruky. Je lineárnou kombináciou vzdialenosti ruky od objektu a natočenia ruky voči hlavnej osi objektu. Presný matematický zápis poskytuje rovnica 3.1.

$$r(t) = \alpha \cdot \left(1 - \frac{2 \cdot |d(t)|}{max_d}\right)^2 \cdot sgn\left(1 - \frac{2 \cdot |d(t)|}{max_d}\right) + \beta \cdot \left(1 - \frac{2 \cdot \sin(|\Delta(t)|)}{max_\Delta}\right)^2 \cdot sgn\left(1 - \frac{2 \cdot \sin(|\Delta(t)|)}{max_\Delta}\right) \quad (3.1)$$

Môžeme si všimnúť, že pomocou parametrov α a β je možné dávať v odmeňovacej funkcii dôraz na skrátenie vzdialenosti alebo správne natočenie ruky. V našom modeli sme odmenu počítali s oboma parametrami rovnými hodnote 0.5. Funkcia sgn je štandardnou matematickou funkciou pre určenie znamienka, ktorá vracia +1 pre argument väčší ako 0 a -1 pre argument menší ako 0. Premenná $d(t)$ predstavuje vzdialenosť ruky od objektu v čase t a $\Delta(t)$ uhol medzi osou Z (os v smere dopredného pohybu) ruky a hlavnou osou dosahovaného predmetu v čase t . Dávame do pozornosti, že používame hodnotu $\sin(\Delta(t))$ z potreby ekvivalencie uhlov 0° a 180° , ktoré sú pre natočenie ruky rovnaké. Konštantná hodnota max_d je maximálna možná vzdialenosť od objektu a max_Δ je rovné $\pi/2$.

Za finálny stav prostredia sa považuje stav, kedy je vzdialenosť ruky od objektu menšia ako polovička dĺžky jej prstov (0.15m) a rozdiel medzi uhlami hlavnej osi a osi Z ruky je menší ako 20° .

Na druhej strane považujeme za fatálny stav, kedy sa ruka príliš priblíži ku stolu (0.05m). Táto podmienka síce so samotnou úlohou dosiahnutia nemá nič spoločné, ale ide samozrejme o ochranu zariadenia. Ďalšou podmienkou je, že nesmú po sebe nasledovať dve akcie, ktoré by nevyvolali zmenu stavu, čo indikuje, že cieľová pozícia je pre rameno nedosiahnuteľná.

Poslednou možnosťou je, že vzdialenosť ruky od objektu je väčšia ako škálovací faktor max_d , ktorý je počítaný ako vzdialenosť od základne ramena k objektu násobená parametrom (1.5).

Podobne ako sme definovali funkciu odmeny pre modul dosiahnutia zdefinuujeme funkciu odmeny pre uchopovanie (rovnica 3.2).

$$r(t) = \alpha \cdot (1 - g(t))^2 + \beta \cdot \left(1 - \frac{2 \cdot \sin(|\Delta(t)|)}{max_{\Delta}}\right)^2 \cdot \operatorname{sgn}\left(1 - \frac{2 \cdot \sin(|\Delta(t)|)}{max_{\Delta}}\right) \quad (3.2)$$

Môžeme si všimnúť, že druhý člen sa vôbec nelíši od rovnice 3.1, že došlo k nahradeniu prvého člena. Ten vyjadruje otvorenosť prstov ruky (premenná $g(t)$) v čase t . Premenná $g(t)$ obsahuje hodnoty z intervalu $\langle 0, 1 \rangle$, kde 0 znamená, že ruka je zavretá, 1 naopak otvorená. Preto nebolo nutné nijako tento člen škálovať, ako v predchádzajúcich prípadoch. Druhý člen lineárnej kombinácie opäť vyjadruje uhol medzi hlavnou osou a osou Z ruky. V prípade uchopovania sme fatálny stav definovali ako stav, kedy sa prsty ruky dokázali úplne uzavrieť, čo znamenalo, že nebol uchopený žiaden objekt. V opačnom prípade kedy prsty niečo uchopili bolo zovretie zastavené vlastnou API ramena, keď „pociťovalo“ odpor v zatváraní prstov. Nevyhnutné pre finálny stav okrem neúplného uzavretia prstov bol rozdiel uhlu osi Z ruky a hlavnej osi menší ako 5° .

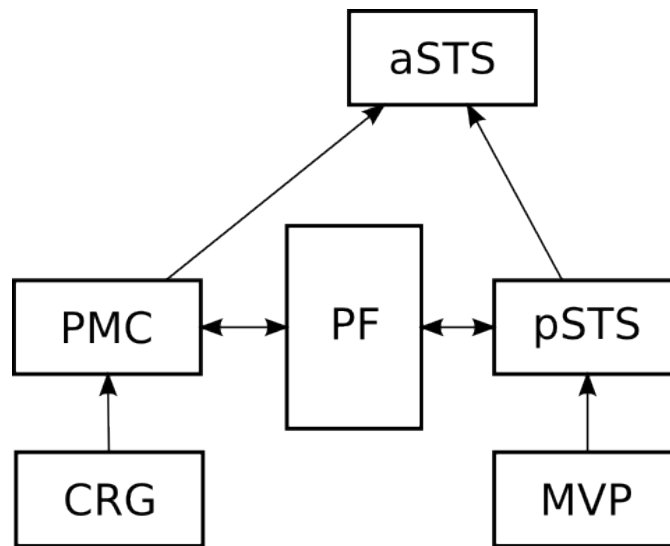
3.5 Architektúra modelu zrkadliacich neurónov

Náš model systému zrkadliacich neurónov pozostáva zo spomenutých modulov integrovaných do jedného celku. Cieľom je simulácia interakcie vizuálnych a motorických informácií v robotickom agentovi [35].

Základom celého modelu je okruh pSTS - PF – PMC(F5), v ktorom prúdia informácie oboma smermi. Modul PF je implementovaný ako skryté vrstva neurónovej siete, kde ďalšie dve vrstvy tvoria moduly pSTS a PMC. Celý tento komplex je učení pomocou algoritmu obojsmerného asociatívneho učenia BAL [39], ktorý by sme mohli zaradiť do triedy algoritmov učenia s učiteľom a bol inšpirovaný algoritmom GeneRec [27].

Pre úplnosť ešte doplníme architektúry jednotlivých okruhov a spôsoby učenia, ktoré používame na ich tréningovanie:

- Vizuálny okruh MVP – pSTS – tréňovaný samoorganizáciou
- Motorický okruh CRG – PMC – tréňovaný samoorganizáciou
- Zrkadliaci okruh pSTS – PF – PMC – tréňovaný pomocou algoritmu BAL
- Okruh abstrahovania invariantnosti pohybu MVP – pSTS + PMC – aSTS – tréningovanie samoorganizáciou



Obr. č. 15: Architektúra modelu zrkadliacich neurónov

4 Implementácia

V nasledujúcich podkapitolách sa budeme venovať detailnému popisu riešenia našich modelov z hľadiska implementácie. Popíšeme štruktúru tried a ich celkové prepojenie, architektúru aplikácie a použité technologické riešenia.

4.1 Technologické riešenie

Oba modely sme implementovali v jazyku C++ kvôli dobrej dostupnosti knižníc tretích strán a existujúcich rozhraní pre programovanie aplikácií (API) pre použité simulačné prostredia a rozhranie s ovládačom ramena Katana.

V projekte CRG sme využívali knižnicu Point Cloud Library (PCL) verzie 1.7 [38]. Knižnica je zameraná na manipuláciu s oblakmi bodov, ktoré sú výstupom zariadenia *Microsoft Kinect*, a podporuje tradičné geometrické transformácie. Spolu s knižnicou PCL 1.7 sme museli používať knižnicu Eigen, ktorá, ako názov predpovedá, je zameraná na vektorovú algebru a knižnica PCL je na nej závislá. Metódy knižnice Eigen sme využívali aj vo vstupnom rozhraní pri výpočtoch parametrov cieľového objektu. Rovnako sme vo vstupnom rozhraní použili aj knižnicu Vision For Robotics (V4R), ktorá je nadstavbou knižnice PCL a poskytuje sofistikované metódy na spracovanie oblakov bodov, segmentáciu oblastí záujmu z hlavnej roviny (obvykle podlaha alebo stolová doska) a ich diskrimináciu na izolované celky. Na komunikáciu s vrstvou s priamym prístupom k ovládaču ramena Katana 450 sme použili API Golem vyvinutú na Univerzite v Birminghame [40], ktorá poskytuje metódy ovládania vyššieho rádu ako je príprava plánu dosiahnutia cieľovej pozície a vykonávanie vypočítaného plánu. Tým sme však stratili prístup k niektorým informáciám o póze ramena Katana a museli sme používať transformovaný stav poskytovaný API Golem.

Celé testovanie modelu CRG prebiehalo v prostredí simulátora Robotic Operation System (ROS) [41], ktorý simuluje fyzikálne zákony, a do ktorého sme vložili zásuvný modul obsahujúce model ramena Katana 450. Prostredie ROS je veľmi flexibilný a široko používaný nástroj na simuláciu prostredia pre vývoj akýchkoľvek riadiacich systémov pre roboty.

V projekte STS sme pri výstupnom zobrazovaní používali aplikáciu Octave, v ktorej sme si

pripravili skripty na zobrazenie kontúrových grafov, a nami implementovanú aplikáciu `Plotter`, ktorá používala knižnicu na tvorbu grafických rozhraní `Qt4`.

4.2 Dátové množiny

Pre model STS bola vytvorená samostatná trieda `CDataSet` spravujúca dáta a zároveň slúžiaca ako vstupné rozhrania. Dátová množina je interne reprezentovaná zoznamom definovaných štruktúr `sSequence`. Táto štruktúra zastupuje jednu sekvenciu úchopu. Obsahuje zoznam vektorov desatinných čísiel, typ triedy a typ perspektívy.

```
struct sSequence {
    list< vector<float> > seq;
    int                    seq_class;
    int                    seq_view;
};
```

K načítaniu súboru, jeho rozparsovaniu, následnému spracovaniu a konečnému škálovaniu dochádza v konštruktoze samotnej triedy `CDataSet` za použitia privátnych metód. Trieda môže byť skonštruovaná v dvoch módoch, kedy sa odlišuje načítanie dátového súboru, a to v móde pre tvorbu trénovacích dát modulu `PMC` a trénovacích dát modulu `pSTS`

Trieda `CDataSet` disponuje:

- geometrickými operátormi rotácie, posunu a projekcie, ktoré sa všetky využívajú v predspracovaní dátovej množiny pre modul `pSTS`
- podpornými funkciami pre operácie s dátovým typom `vector` ako je porovnávanie, odčítanie, delenie, výpočet veľkosti a skalárny súčin vektorov
- funkciami zabezpečujúcimi nájdenie maximálneho vektora a predelenie celej dátovej množiny týmto vektorom, ktoré sú jadrom škálovacej funkcie

Výstupom bol zoznam štruktúr `sSequence` s dátami preškálovanými do intervalu $\langle -1, 1 \rangle$.

V modely `CRG` bolo vstupné rozhranie a funkcie predspracovania dát integrované do triedy prostredia `CEnvironment`. Na tento účel trieda `CEnvironment` obsahovala:

- funkciu načítania oblaku bodov využívajúcu metódy knižníc `PCL` a `V4R`

- pomocné funkcie definujúce rovinu stola, ktoré slúžili na transformáciu súradníc dát a robotického ramena
- funkcie pre výpočet ťažiska predmetu a výpočet hlavných osí predmetu

Výstupom bola pozícia a orientácia cieľového objektu, ktoré reprezentovali ideálny finálny stav.

4.3 Samoorganizujúce sa mapy

Patria sem triedy SOM, MSOM, RecSOM. Všetky vymenované triedy zdieľajú spoločnú programovú štruktúru. Základom je jednorozmerné pole veľkosti výstupnej vrstvy siete tvorené triedami `Neuron`.

Trieda `Neuron` obsahuje vstupný vektor, vektor váh a vektor kontextových váh (pre sieť SOM má nulový rozmer). Táto trieda je spoločná pre všetky architektúry sietí a obsahuje:

- preťažené funkcie implementujúce rovnice pre výpočet vzdialenosti
- preťažené funkcie pre aktualizáciu váh
- pomocné funkcie na výpočet euklidovskej a gaussovskej vzdialenosti, implementovaná je aj funkcia na výpočet vzdialenosti typu mexický klobúk, avšak tá sa nikde nevyužíva

Každá sieť obsahuje smerník na triedu `CDataSet` a taktiež sadu funkcií:

- inicializačné funkcie naplňajúce smerník na trénovacie dáta a parametre učenia
- funkcie zabezpečujúce prechod trénovacej množiny, funkciu nájdania víťazného neurónu a funkciu trénovania na jednom vstupnom vektore, ktorá sa volá pre každý vektor trénovacej množiny, a funkciu aktualizácie kontextovej vrstvy
- testovacie funkcie aktivácie siete na vstupný vektor, funkcie zápisu výstupu do súboru
- pomocné trénovacie funkcie znižujúce parametre rýchlosti učenia a okolia

4.4 Architektúra modelu STS

Model STS pozostával zo štyroch modulov:

- Modul PMC implementovaný triedou MSOM
- Modul Vizuálneho Predspracovania (MVP), ktorý predstavuje trieda CDataSet
- Modul pSTS implementovaný triedou MSOM
- Modul aSTS implementovaný triedou SOM

Prepojenie všetkých štyroch modulov zabezpečuje samostatná trieda STS, ktorá vo svojom konštruktore vytvára všetky samoorganizujúce sa mapy a vkladá smerníky na dve inštancie triedy CDataSet, jednu v móde načítania dát pre modul PMC a druhú v móde načítania dát pre modul pSTS. Okrem toho táto trieda obsahuje metódu tréningu, v ktorej sa súčasne prechádzajú obe dátové množiny, pričom množina pre PMC je štvornásobne menšia a tak sa vždy zvyšuje iterátor po štyroch zvýšeníach iterátora množiny pre pSTS. Po prechode jednou sekvenciou sa vždy zoberie aktivácia oboch modulov PMC a pSTS a pomocou funkcie na spájanie polí sa vytvorí vstupný vektor pre modul aSTS. Pre ilustráciu prikkladáme aj časť zdrojového kódu.

```

while(seq_it_v != vis_trn_data->end()) {

    sSequence pmc_sample = *seq_it_m;
    sSequence sts_sample = *seq_it_v;

    sam_it_v = sts_sample.seq.begin();
    sam_it_m = pmc_sample.seq.begin();

    while(sam_it_v != sts_sample.seq.end()) {
        // train PMC
        if (persp == 0) {
            qerr_pmc += PMC->train_one_sample(*sam_it_m, GAUSSIAN, EXP);
            sam_it_m++;
        }
        // train STSp
        qerr_stsp += STSp->train_one_sample(*sam_it_v, GAUSSIAN, EXP);
        sam_it_v++;
    }
    // train STSa
    PMC_output = PMC->get_activation();
    STSp_output = STSp->get_activation();
}

```

```

        concat_array(PMC_output, pmc_size[0], STSp_output, stsp_size[0],
&input);
    qerr_stsa += STSa->train_one_sample(input, GAUSSIAN, false);

    STSp->reset_context();
    seq_it_v++;
    persp++;

    if (persp == 4) {
        PMC->reset_context();
        seq_it_m++;
        persp = 0;
    }
}

```

Testovanie modelu STS implementované v samostatnej funkcii prebieha v obdobnej programovej štruktúre, kde akurát pochopiteľne nedochádza k adaptácii váh, iba k ukladaniu aktivácií všetkých modulov do dátových štruktúr reprezentujúcich histogramy distribúcie aktivity pre každý neurón. Tie sú následne uložené do súborov na pevnom disku.

4.5 Architektúra modelu CRG

Jadro modelu CRG tvorí trieda `CEnvironment`, ktorá slúži ako vstupné rozhranie a zároveň ako rozhranie medzi nižšími kontrolnými vrstvami Golem a ovládača ramena Katana 450. Táto trieda plní niekoľko kľúčových úloh:

- udržiava neustále aktuálny stav prostredia, ktorý získava z nižšej vrstvy Golem
- posiela signály na vykonanie akcií, ktoré dostáva od vyšších riadiacich modulov pre dosahovanie uchopovanie
- kontroluje či nebol dosiahnutý finálny alebo fatálny stav
- obsahuje funkcie na výpočet zmeny stavu a výpočet hodnoty odmeny

Trieda `CEnvironment` je inicializovaná v dvoch módoch, jeden určený pre modul dosahovania, druhý pre modul úchopu.

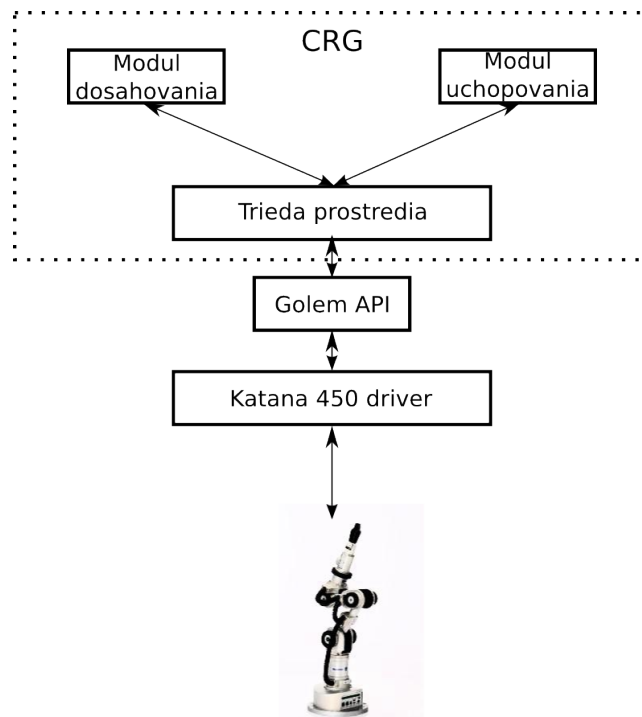
Oba kontrolné moduly sú implementované ako samostatné triedy `CGraspingModule` a `CReachingModule`. Z hľadiska programovej štruktúry sú takmer identické a ich hlavnou

úlohou je integrácia nižších tried `CActorCritic`, `CCACLA` a pomocnej triedy na zápis výstupu počas učenia `CLogger`.

Trieda `CActorCritic` implementuje dve neurónové siete (triedy `CNetwork`) aktéra a kritika a zabezpečuje distribúciu vstupu pre obe siete, ich aktiváciu a spätné šírenie odmeny. Triedy `CNeuron` a `CNetwork` boli zabezpečené našou knižnicou `libNeuralNetwork`, ktorá je tiež súčasťou projektu. Trieda `CNetwork` implementuje napevno trojvrstvovú doprednú sieť, kde jednotlivé vrstvy pozostávajú z inštancií triedy `CNeuron`.

Trieda `CCACLA` je implementáciou trénovacieho algoritmu `CACLA`, pričom hlavnou metódou je metóda `Train`. Obsahuje taktiež pomocné metódy na výpočet chyby, výpočet gaussovskej explorácie a výpočet náhodnej pozície distribuovanej gaussovsky metódou G. Marsagliu [42].

Tok riadenia prechádza z modelu `CRG` ešte dvoma nižšími vrstvami (viď. Obr. č. 14), ktoré zabezpečujú komunikáciu so samotným ramenom v reálnom svete alebo simulátore `ROS`.



Obr. č. 14: Architektúra toku riadenia ramena Katana 450 modelom `CRG`

4.6 Výstupné rozhrania

Pomocou aplikácie Plotter sme vizualizovali výstupy všetkých samoorganizujúcich sa máp v našom modeli STS, ktoré možno vidieť v kapitole Experimenty. Pri vývoji aplikácie sme použili knižnicu Qt4 a voľne dostupný prvok QtGraphWidget.

4.7 Knižnica libSOM

Jedným z bočných produktov celej našej práce bola séria rôznych architektúr samoorganizujúcich sa máp, ktoré boli pomerne transparentne naimplementované a dali sa jednoducho spájať od väčších štruktúr. Rovnako boli schopné generovať výstupné dáta a obsahovali jednotné vstupné rozhranie, ktoré je jednoduché aplikovať na akúkoľvek dátovú množinu a tak trénovať mapy na akýchkoľvek dátach.

Rozhodli sme sa preto, že naše zdrojové kódy mierne zovšeobecníme a vytvoríme z nich dynamickú knižnicu s názvom *libSOM*, ktorú poskytneme ako opensource projekt potencionálnym záujemcom a prípadne ďalším študentom. Knižnica nie je nijak licencovaná a tým pádom môže byť kýmkoľvek rozvíjaná a dopĺňaná. Dúfame, že tak v budúcnosti uľahčíme a urýchlíme proces vytvárania zložitejších modelov, ktoré budú využívať architektúry samoorganizujúcich sa máp.

Zdrojové kódy knižnicu je možné voľne stiahnuť napísaním príkazu do konzoly:

```
svn checkout http://libsom.googlecode.com/svn/trunk/ libsom-read-only
```

Knižnicu neskôr plánujeme presunúť do repozitára cogsci, o tom však už každý nájde informáciu na stránke projektu.

5 Experimenty

V tejto kapitole zhrnieme dizajn a výsledky všetkých podstatných experimentov a testov, ktoré sme na našich modeloch robili. V prípade modelu pSTS a PMC sme sa v prvom kroku sústredili na dávkové testy, pomocou ktorých sme hľadali vhodné množiny parametrov pre sieť MSOM a taktiež sme sa snažili podľa možnosti skrátiť výpočtový čas celého modelu STS. To sme zabezpečili vhodným výberom architektúry pre oblasť aSTS, čo prispelo k skráteniu času tréningu celkovo k lepšej použiteľnosti modelu.

V druhom kroku experimentov s modelom STS sme prešli k testovaniu primárnej a sekundárnej organizácie modulu pSTS a PMC, pričom sme predpokladali výslednú organizáciu a očakávali sme splnenie našej hypotézy.

V treťom kroku sme pristúpili k otestovaniu učenia modulu aSTS, v ktorom sme sa snažili o vytvorenie perspektívne invariantných neurónov. Taktiež sme pri tom odskúšali niekoľko druhov filtrácie vstupných signálov braných z modulov pSTS a PMC.

Pri testovaní modelu CRG sme sa sústredili na správne vykonanie primárnej úlohy a to dosiahnutie oblasti cieľového objektu a jeho uchopenie. Skúmali sme taktiež limity a vlastnosti použitej architektúry vzhľadom na zašumenie polohy cieľového objektu a na schopnosť naučiť model uchopiť viac objektov.

5.1 Prieskum vlastností modelu STS

Ako bolo spomínané v návrhu (Kap. 3) model STS je najkomplexnejší z modelov obsiahnutých v tejto práci. Tvoria ho dve siete architektúry MSOM, ktoré svojím výstupom aktivujú tretiu sieť architektúry SOM. V popise formálnych modelov (Kap. 2) sme uvádzali, že sieť MSOM má dva parametre α a β , ktoré určujú akým spôsobom sa bude vypočítavať rekurzívna vzdialenosť každého neurónu a ako sa bude aktualizovať kontextový vektor. Pre správne určenie týchto parametrov v závislosti od tréningových dát neexistuje žiaden postup, a preto je nutné tieto parametre zistiť sériou testov. Preto sme sa rozhodli vykonať niekoľko dávkových testov, kde sme systematicky prechádzali rôzne kombinácie parametrov a vyhodnocovali sme tri miery určujúce úspešnosť:

- kvantizačná chyba

V našom prípade vyjadruje rekurzívnu vzdialenosť víťaza ($d(t)_{b(t)}$) od vstupného vektora. Počíta sa vždy za jednu epochu, čiže po prejdení celej tréningovej množiny X .

$$q_{err} = \sum_{x \in X} d(t)_{b(t)} \quad (5.1)$$

- odlišnosť víťazov

Vyjadruje koľko neurónov v sieti sa aspoň raz stalo víťazmi. Väčšinou sa vyjadruje ako relatívne číslo, ale my sme sa rozhodli preškálovať túto hodnotu na percentá, čo nám prišlo ako zrozumiteľnejšie podanie. Rovnica 5.2 teda zachytáva, že odlišnosť víťazov je podielom mohutnosti množiny všetkých víťazov za epochu a celkového počtu neurónov n .

$$wd = \frac{|[b(t)]|}{n} \quad (5.2)$$

- entrópia

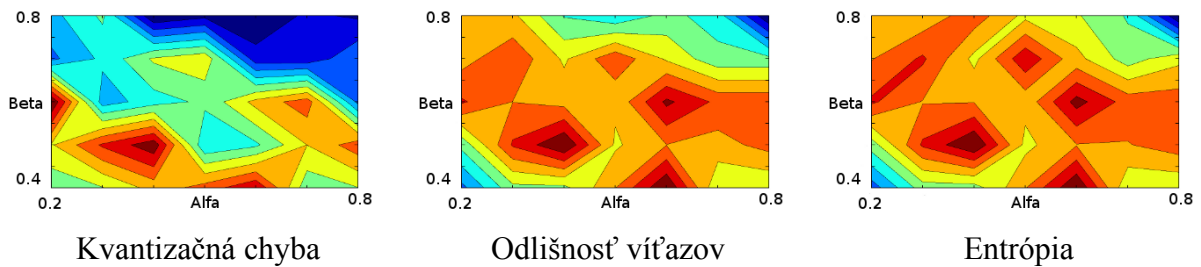
Vyčísluje mieru neusporiadanosti v mape, ktorá rastie s počtom variability víťazov. Výpočet je formálne opísaný rovnicou 5.3, kde $w(i)$ je funkcia, ktorá priradí neurónu i počet víťazstiev na tréningovej množine X a n je počet všetkých neurónov siete.

$$e = \sum_{i=1}^n \frac{w(i)}{|X|} \cdot \log \left(\frac{w(i)}{|X|} \right) \quad (5.3)$$

Výstupom takýchto dávkových testov bola tabuľka (Tab. č. 1) hodnôt spomínaných troch mier pre jednotlivé kombinácie hodnôt parametrov α a β . Hodnoty kvant. chyby, entrópie a odlišnosti víťazov Tú sme zobrazili ako tri kontúrové grafy, ktoré zachytávali hodnoty miery pre všetky možné hodnoty parametrov.

α	β	Kvantizačná chyba	Entrópia	Odlíšnosť víťazov
0.2	0.4	31.91	0.72	0.806
0.2	0.5	33.63	0.77	0.861
0.2	0.6	37.28	0.77	0.861
0.2	0.7	39.99	0.81	0.903
0.2	0.8	43.72	0.86	0.958
0.3	0.4	32.10	0.78	0.868
0.3	0.5	33.69	0.78	0.868
0.3	0.6	36.05	0.77	0.854
0.3	0.7	41.76	0.83	0.924
0.3	0.8	45.54	0.86	0.958
0.4	0.4	29.03	0.79	0.882
0.4	0.5	30.82	0.81	0.903
0.4	0.6	36.34	0.81	0.896
0.4	0.7	40.37	0.82	0.910
0.4	0.8	46.37	0.83	0.924
0.5	0.4	27.41	0.81	0.903
0.5	0.5	30.12	0.81	0.896
0.5	0.6	31.84	0.80	0.889
0.5	0.7	37.19	0.85	0.944
0.5	0.8	40.23	0.82	0.917
0.6	0.4	25.52	0.82	0.917
0.6	0.5	25.65	0.81	0.896
0.6	0.6	29.33	0.83	0.917
0.6	0.7	34.20	0.79	0.882
0.6	0.8	35.74	0.84	0.917
0.7	0.4	22.17	0.80	0.882
0.7	0.5	21.97	0.77	0.854
0.7	0.6	25.85	0.79	0.854
0.7	0.7	27.94	0.80	0.889
0.7	0.8	31.76	0.80	0.889
0.8	0.4	18.71	0.82	0.847
0.8	0.5	20.42	0.79	0.833
0.8	0.6	19.29	0.78	0.854
0.8	0.7	21.66	0.76	0.840
0.8	0.8	20.85	0.68	0.757

Tab. č. 1: Príklad dát dávkového testu, z ktorých boli skonštruované kontúrové grafy.



Obr. č. 16: Kontúrové grafy pre testovanie modelu pSTS

Za optimálne sme považovali miesto, kde mal modul na konci tréningovania najnižšiu kvantizačnú chybu (modré oblasti) a najvyššiu entropiu a odlišnosť víťazov (červené oblasti). Môžeme si všimnúť, že kontúrový graf pre entropiu a odlišnosť víťazov sa veľmi podobajú, keďže tieto dve miery spolu úzko súvisia. Dávkové testy sme zbiehali zvlášť pre modul pSTS a zvlášť pre modul PMC. Optimálne hodnoty sú zobrazené prehľadne v tabuľke č. 2.

Modul \ Parametre	α	β
pSTS	0.3	0.7
PMC	0.3	0.5

Tab. č. 2: Optimálne hodnoty parametrov architektúry MSOM pre moduly pSTS a PMC.

Pri testovaní celého modelu STS zahŕňajúceho aj modul aSTS sme robili niekoľko experimentov s rôznou architektúrou modulu aSTS, pričom sme sa snažili nájsť kompromis medzi biologickou plauzibilitou a optimálnym výpočtovým časom. Hodnotili sme taktiež organizáciu modulu aSTS po natrénovaní, pričom hlavným cieľom v tejto fáze experimentu bolo dosiahnuť opačnú organizáciu ako pri module pSTS. Ako najmenej vhodná sa ukázala architektúra MSOM, ktorá z dôvodu veľkej dimenzie vstupu bežala najdlhšie a výsledok sa nezhodoval s našimi očakávaniami, pričom nemožno vylúčiť, že pri vhodnej konfigurácii parametrov, by sa modul aSTS nezachoval správne, avšak na tak dlhé dávkové testy nebol dostatok času a prišlo nám to ako podružná úloha. S lepšimi výsledkami skončila architektúra RecSOM, ktorá sa správne naorganizovala, avšak čas učenia mala len o málo nižší ako architektúra MSOM. Najlepšie sa osvedčilo použitie architektúry SOM, ktorá nespracovávala sekvenčne aktivácie modulov pSTS a PMC, ale iba aktivácie na konci sekvencií, čo výrazne

napomohlo urýchleniu celého tréningového procesu. Rovnako splnila aj naše očakávania o výslednej organizácii, a preto sme sa rozhodli v ďalších experimentoch používať výlučne architektúru SOM ako implementáciu modulu aSTS. Čas učenia samozrejme závisia od mnohých iných faktorov, ako je dimenzionalita vstupu a veľkosť samotnej siete. Pri našich testoch boli tieto hodnoty zafixované.

Architektúra aSTS \ Kritériá	Čas učenia modelu STS	Hodnotenie
MSOM	5-6 hod	Model nespĺňal kritériá
RecSOM	4-5 hod	Model spĺňal kritériá
SOM	2 hod	Model spĺňal kritériá

Tab. č. 3: Výsledky testovania rôznych architektúr ako implementácie modulu aSTS.

5.2 Organizácia pSTS a PMC

Cieľom tohto experimentu bolo reprodukovať výsledné dáta výskumu Perretta a kol. [5], ktoré poukazovali na vnútornú organizáciu oblasti STS. Pri výskume aktivácií populácií neurónov reagujúcich na jeden druh biologického pohybu sa ukázalo, že existuje druhá organizácia, v ktorej sa jednotlivé populácie aktivujú pre jednotlivé perspektívy, pod ktorými pozorovateľ akciu vidí. V neskoršej práci Peretta a kol. [6] bol upresnený výskyt týchto populácií v oblasti pSTS.

Preto sme pre pSTS pripravili dáta pozostávajúce z inštancií troch typov úchopov, pričom každá inštancia bola zastúpená ešte v štyroch perspektívach. Pomocou predošlých dávkových testov na tréningových dátach sme poznali optimálne parametre modulu pSTS implementovaného sieťou MSOM.

Náš predpoklad bol, že v sieti sa po správnom natrénovaní vytvorí hlavná organizácia zodpovedajúca štyrom perspektívam, pod ktorými sú úchopy pozorované a v rámci každej perspektívy vznikne sekundárna organizácia, ktorá zachytáva reprezentáciu troch typov úchopov.

V nasledujúcej tabuľke vidno všetky parametre simulácií, ktoré sme v tomto experimente spúšťali:

α	β	γ	ϵ	x
0.3	0.7	0.1	300	12x12
0.3	0.7	0.1	300	14x14
0.3	0.7	0.1	300	16x16
0.3	0.7	0.1	300	18x18
0.3	0.7	0.1	300	20x20

Tab. č. 4: Prehľad parametrov simulácií experimentu pSTS

Parametre α a β sú parametrami siete MSOM, parameter γ vyjadruje rýchlosť učenia, parameter ϵ je počet epôch učenia a parameter x vyjadruje veľkosť siete, teda počet neurónov vo výstupnej vrstve.

Aktivitu siete sme vyhodnocovali dvoma spôsobmi:

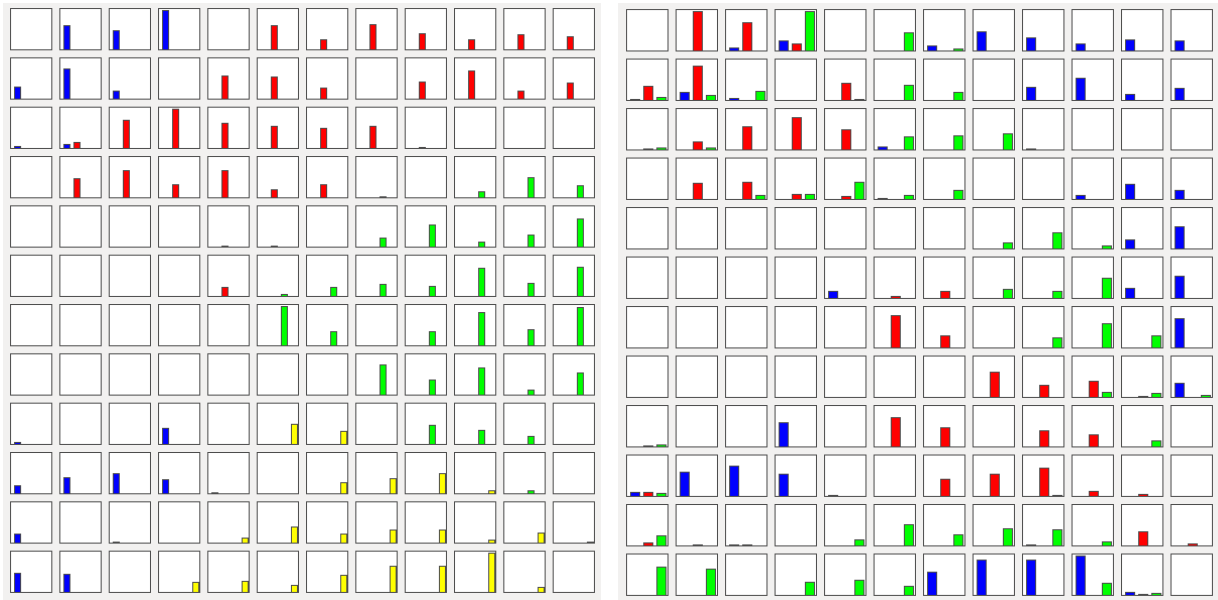
1. Ako rekurzívnu vzdialenosť každého neurónu od víťaza, ktorá vstúpila ako argument do exponenciálnej funkcie. Takto vypočítanú hodnotu sme potom pripočítali k jednej z n -tice hodnôt, kde n vyjadrovalo typ triedy (tj. typ úchopu alebo perspektívu) a k hodnote celkovej vzdialenosti, ktorá fungovala ako normalizačný faktor, ktorým boli hodnoty pre každú triedu n predelené, aby dávali v súčte hodnotu 1.

$$a_i^n = \sum_{x \in X_n} e^{-d_i(x)} \quad \text{norm}_i = \sum_{x \in X} e^{-d_i(x)} \quad (5.4)$$

2. Ako počet víťazstiev neurónu v trénovacej dátovej množine pre jednotlivé triedy (typ úchopu alebo perspektívu). Ak sa teda neurón stal víťazom, tak bola o jedna zvýšená hodnota, ktorá udržiaval počet víťazstiev pre ten neurón a danú triedu.

$$a_i^n = \sum_{x \in X_n} (1 | \arg \min_i \{d_i(x)\}) \quad (5.5)$$

Po úspešnom natrénovaní sa v sieti vytvorila nami očakávaná organizácia podľa perspektív a následná sekundárna organizácia podľa typov úchopov pre každú perspektívu. Pri vyššom počte neurónov začínali byť reprezentácie redšie, pričom sme pozorovali vznik bivariantných a trivariantných, rovnako ako neurónov reagujúcich na každú perspektívu, ktoré sa nikdy nestávali víťazmi a ich aktivita bola veľmi slabá.



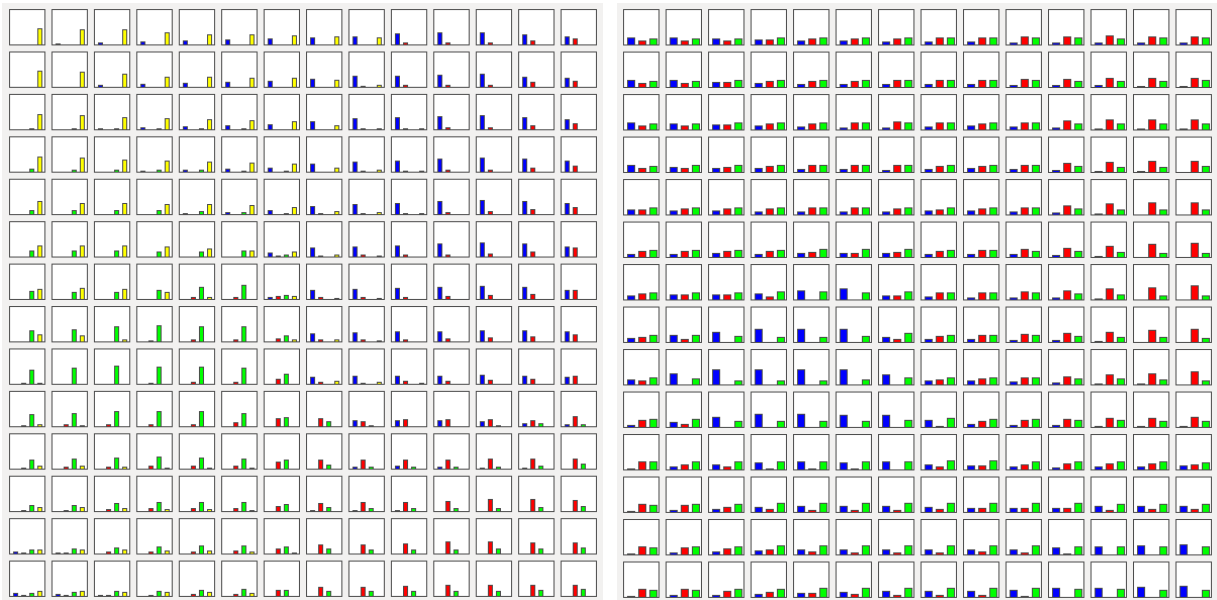
Obr. č. 16: Zobrazenie natrénovaných máp zobrazovacou technikou č. 2.

Vľavo: Zobrazenie z hľadiska perspektív (modrá 0°, červená 90°, zelená 180°, žltá 270°)

Vpravo: Zobrazenie z hľadiska typov úchopov (modrá úchop zhora, červená úchop od boku, zelená presný úchop)

V nami použitom zobrazení reprezentuje každý štvorec jeden neurón mapy a farebné stĺpce v ňom zodpovedajú triedam vzhľadom na ktoré bola mapa vyhodnocovaná. Pod obrázkami sa vždy nachádza legenda. Výška stĺpcov vyjadruje veľkosť hodnoty aktivity alebo počtu víťazstiev, ktoré neurón prejavil pre dáta patriace do príslušnej triedy.

Na výsledných zobrazeniach môžeme vždy vľavo vidieť štyri oblasti, ktoré sú farebne oddelené a zodpovedajú štyrom perspektívam a napravo je tá istá mapa, iba vyhodnotená vzhľadom na typy úchopov. Neuróny, ktoré sú aktívne pre jednu perspektívu vľavo, sú vždy napravo rozdelené na tri menšie skupiny reagujúce na rôzny typ úchopu. Takto natrénovaná sieť obsahuje až 12 populácií neurónov, ktoré sa aktivujú pre rôzne vstupné kombinácie úchop-perspektíva.



Obr. č. 17: Zobrazenie natrénovaných máp zobrazovacou technikou č. 1.

Vľavo: Zobrazenie z hľadiska perspektív (modrá 0°, červená 90°, zelená 180°, žltá 270°)

Vpravo: Zobrazenie z hľadiska typov úchopov (modrá úchop zhora, červená úchop od boku, zelená presný úchop)

Výsledné zobrazenie aktivít netrénovaných máp je na obr. č. 17. Je nutné ešte podotknúť, že z výsledných hodnôt pre aktivitu bolo ubratých prvých 15% kvôli zvýrazneniu rozdielu. Vľavo si môžeme všimnúť už spomínané bivariantné, trivariantné aj veľmi slabo reagujúce invariantné neuróny (približne v strede). Napravo môžeme vidieť, že väčšina neurónov reaguje na všetky typy úchopov, čo je možné vysvetliť značnou podobnosťou úchopov, ktoré sú v princípe rozdielne až v záverečných krokoch sekvencie, kedy sa odlišujú pózou a natočením dlane. Rovnako si môžeme všimnúť, že pre niektoré perspektívy (zelená) je zjavne odlišnosť typov úchopu väčšia ako pre iné (žltá).

Druhou úlohou v tomto experimente bolo vytvorenie reprezentácií v module PMC, ktorý zodpovedá premotorickej kôre v ľudskom mozgu a oblasti F5 v mozgu primátov. Práve v tejto oblasti sa nachádzajú reprezentácie pohybov, ktoré sme schopní vykonávať, o ktorých máme znalosť, a medzi neurónmi kódujúcimi tieto reprezentácie má časť aj zrkadliacu vlastnosť a dochádza k ich aktivácii pri pozorovaní rovnakej akcie.

Na tréning sme použili dátovú množinu pozostávajúcu z inštancií troch typov úchopov,

pričom každý bol reprezentovaný proprioceptívnou informáciou o polohe ruky, teda úhlami natočenia jednotlivých kĺbov. Z každej sekvencie sme ešte vystrihli 4 úvodné kroky, aby sa čo najmenej vytvárali kváziinvariantné neuróny, ktoré by v skutočnosti reagovali na iniciálnu polohu ruky, ktorá je pre každý typ úchopu spoločná a to, že je vždy zaradená do inej triedy je len vecou experimentálneho dizajnu, nie objektívnou skutočnosťou.

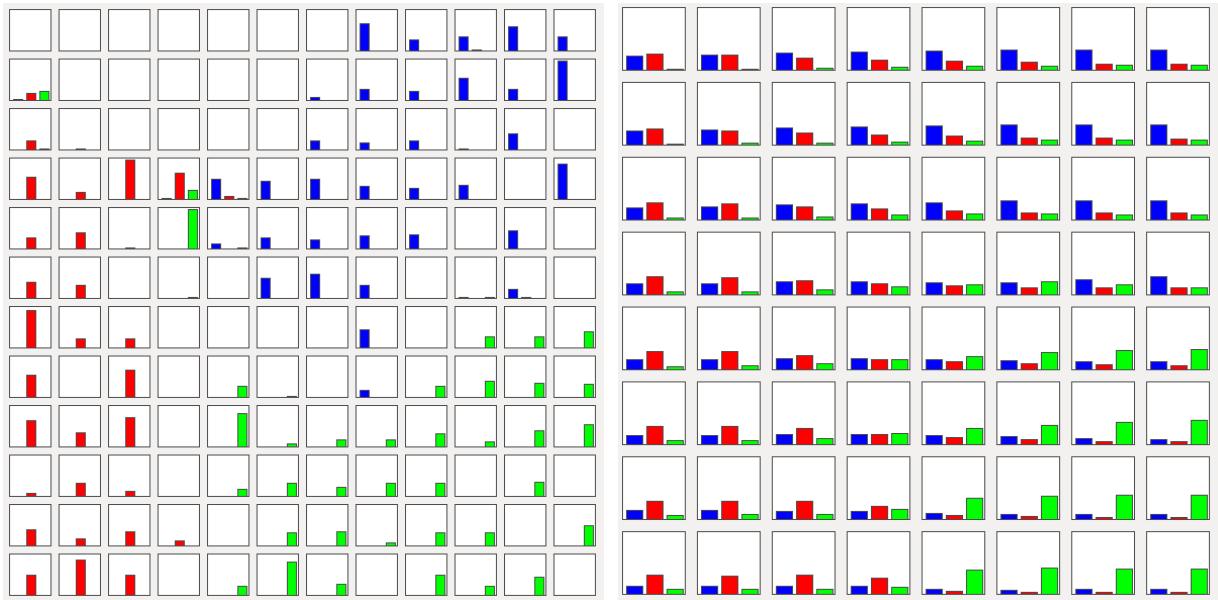
V nasledujúcej tabuľke vidno všetky parametre simulácií, ktoré sme v tomto experimente spúšťali:

α	β	γ	ϵ	x
0.3	0.5	0.02	300	8 x 8
0.3	0.5	0.02	300	10 x 10
0.3	0.5	0.02	300	12 x 12
0.3	0.5	0.02	300	14 x 14

Tab. č. 5: Prehľad parametrov simulácií experimentu PMC

Parametre α a β sú parametrami siete MSOM, parameter γ je rýchlosť učenia, parameter ϵ je počet epôch učenia a parameter x vyjadruje veľkosť siete, teda aj počet neurónov vo výstupnej vrstve.

Predpokladali sme, že organizácia siete bude podľa typov úchopu, pričom časť neurónov sa nebude špecializovať a budú v rámci úchopov invariantné, prípadne bivariantné. Natrénované mapy sme vyhodnocovali rovnakými metódami ako pri experimente s tréňovaním pSTS (rovnica 5.4 a 5.5).



Obr. č. 18: Zobrazenie natrénovaných máp modulu PMC
 Vľavo: Zobrazenie víťazov podľa typov úchopov
 Vpravo: Zobrazenie aktivity neurónov podľa typov úchopov

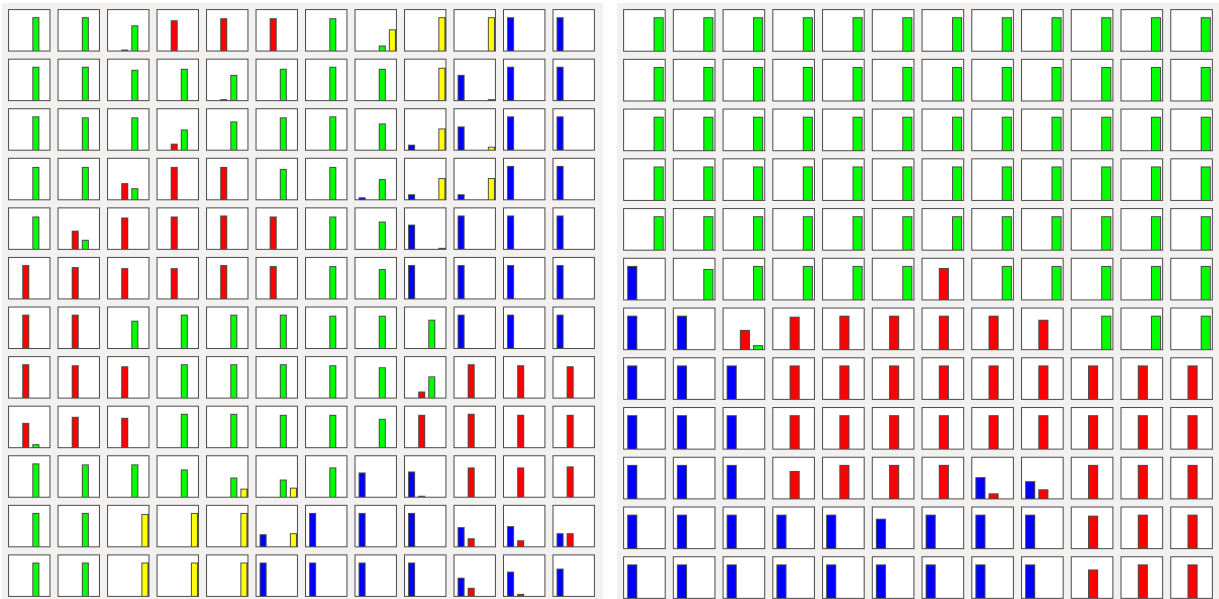
Napravo aj naľavo si môžeme všimnúť invariantné neuróny, ktoré ešte stále reagujú na takmer rovnaké úvodné kroky každej sekvencie. Zreteľne ich vidno na pravom zobrazení v strede, kde majú najnižšiu aktivitu a sú aktívne takmer rovnako pre každý typ úchopu. Smerom k rohom sa aktivita stále viac diferencuje a stáva zameranou na jeden druh úchopu s výnimkou ľavého horného rohu, kde môžeme pozorovať signifikantnú aktivitu bivariantných neurónov citlivých na úchop zhora a úchop od boku, čo sú tvarom dlane veľmi podobné úchopy akurát vzájomne pootočené o 90° .

5.3 Vytvorenie perspektívne invariantných reprezentácií v module aSTS

Cieľom našich experimentov s modulom aSTS bolo dosiahnutie presne opačnej organizácie, akú sme generovali v module pSTS, čiže hlavné populácie neurónov by mali odpovedať na typ úchopu a v rámci nich by sa mali vytvoriť druhotné populácie diskriminujúce perspektívu pod akou bol úchop pozorovaný. Za týmto účelom sme k výstupu z pSTS pridali výstup z modulu PMC a takýto kombinovaný vektor slúžil ako vstup do modulu aSTS.

V prvom experimente sme brali aktívacie oboch modulov a spájali ich do vstupu pre aSTS bez

akejkoľvek úpravy a dosiahnutý výsledok bol uspokojivý ako možno vidieť na obrázku č. 19.



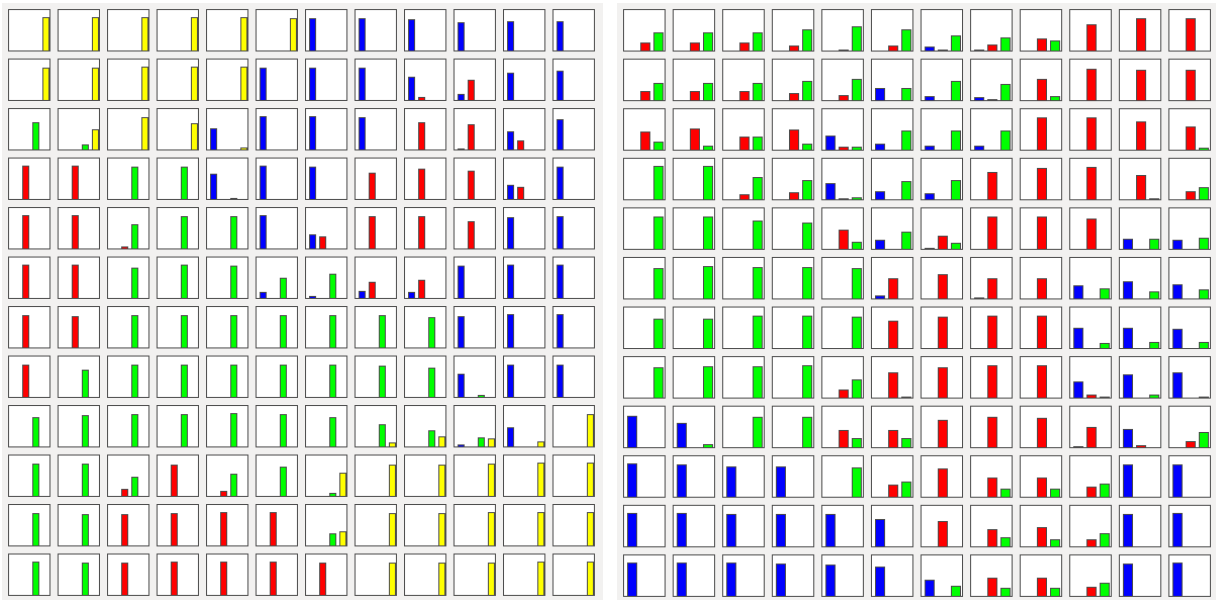
Obr. č. 19: Výsledky organizácie mapy po zložení výstupov z PMC a pSTS bez úprav.

Vľavo: Zobrazenie z hľadiska perspektív (modrá 0°, červená 90°, zelená 180°, žltá 270°)

Vpravo: Zobrazenie z hľadiska typov úchopov (modrá úchop zhora, červená úchop od boku, zelená presný úchop)

Na zobrazeniach možno vidieť hlavnú organizáciu podľa úchopov, pričom zelený typ úchopu zaberá celú hornú polovicu mapy. To vysvetľujeme nerovnakom distribúciou dĺžky sekvencií úchopov v tréningových dátach sietí PMC a pSTS. V zobrazení perspektív vidno, že žltá perspektíva je potlačená na úkor ostatných, čo sa nám zatiaľ zdôvodniť nepodarilo. Snáď by v tom mohla hrať úlohu počiatočná inicializácia siete, v ktorej bola žltá perspektíva najmenej pokrytá. Niekoľko neurónov vykazuje miernu bivariantnú aktivitu vzhľadom na perspektívy, čo chceme využiť a dodatočnými mechanizmami v budúcnosti dosiahnuť aj vytvorenie perspektívne invariantných neurónov, kvôli ktorým sme primárne tento modul dizajnovali.

V druhom experimente sme ponechali učiaci proces identický s prvým experimentom, čiže aSTS dostáva nepozmenený vstup od modulov PMC a pSTS. Pri testovaní modulu sme vstup od PMC úplne vypli a sledovali sme ako sa bude sieť aktivovať, ak bude dostávať vstup iba od modulu pSTS.



Obr. č. 20: Výsledky organizácie mapy po vypnutí vstupu od modulu PMC pri testovaní.
 Vľavo: Zobrazenie z hľadiska perspektív (modrá 0° , červená 90° , zelená 180° , žltá 270°)
 Vpravo: Zobrazenie z hľadiska typov úchopov (modrá úchop zhora, červená úchop od boku, zelená presný úchop)

Vo výsledných zobrazeniach druhého experimentu vidíme, že pôvodne tri kompaktné populácie neurónov, ktoré diskriminovali typy úchopov, sa rozpadli na viacero populácií, pričom niektoré ostali zamerané čisto na jeden typ úchopu a mohli by sme ich nazvať jadrami reprezentácie úchopov. Iné sa stali bivariantnými vždy s preferenciou niektorého typu úchopu. Iba malá množina vykazuje slabé aktivity pre všetky úchopy opäť s dominanciou pre jeden typ. Z hľadiska perspektív sa organizácia mapy nijak výrazne nezmenila.

5.4 Testy modelu CRG

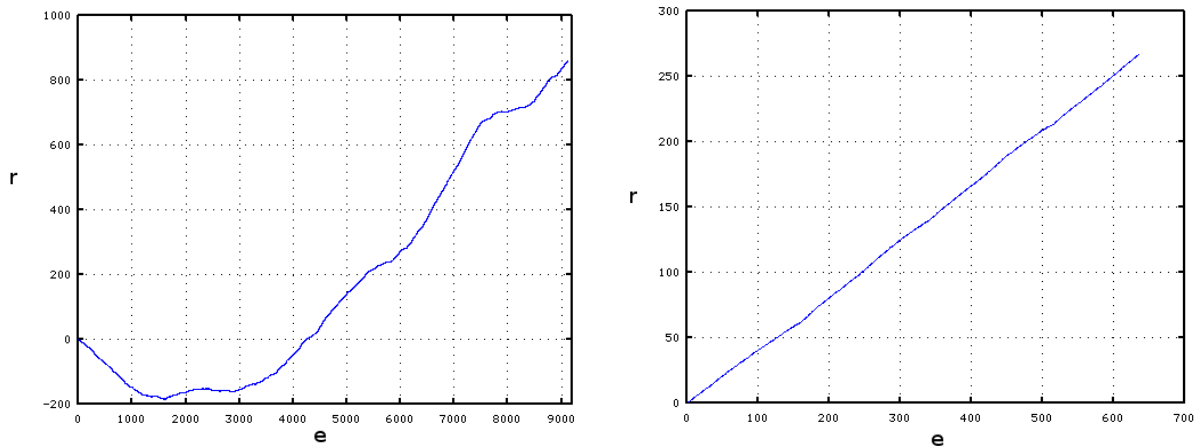
Naše experimenty sme rozdelili do dvoch častí. V prvej časti sa mal model naučiť uchopiť objekt z tréningových dát. V tomto prípade sme merali dve miery úspešnosti – kumulatívnu odmenu r a počet akcií (krokov) e , potrebných n -krát úspešne splniť cieľ, kde n je voľný parameter, ktorý sme nastavili na 5. Celá prvá časť experimentu prebiehala v simulovanom prostredí ROS (Robotic Operating System) z pochopiteľných dôvodov časovej náročnosti tohto procesu. V druhej časti sme pristúpili k experimentom v laboratóriu s ramenom Katana 450. K pozícii objektov z tréningovej množiny sme pridali šum a testovali sme na nich natrénovaný

model. Tu sme merali úspešnosť modelu a počet akcií, nutných k pretrénovaniu modelu na uchopenie zašumeného objektu. Je dôležité pripomenúť, že sme zároveň so šumom pridali aj mieru exploračie, ktorá bola vždy rovná veľkosti zašumenia, aby bol model schopný explorať akcie a tak sa preučiť na uchopenie nového objektu.

Na obrázku č. 21 vidíme typickú krivku učenia pre jeden z objektov (vertikálna krabica). Na začiatku robí model v podstate náhodné akcie a mapuje nimi svoj stavový priestor, preto je kumulatívna chyba negatívna a klesá. Po približne 4300 krokoch začne dominovať pozitívna odmena a model nájde akcie, ktoré sú posilňované a vedú k cieľovému stavu. Tabuľka č. 6 ukazuje výsledky pre všetky štyri objekty v tomto poradí: 1. objekt – vertikálna krabica, 2. objekt – horizontálna krabica (od čaju), 3. objekt – plechovka (vertikálna hlavná os) a 4. objekt – ležiaca krabica so šesťuholníkovou podstavou. Vyhodnotili sme taktiež pomer kumulovanej odmeny a počtu krokov, čo sa dá považovať za priemerný prírastok odmeny po jednej akcii. V prípade modulu dosahovania vidíme, že prírastok nepresiahol hodnotu 0.01, ale pri uchopovaní je pomerne dosť vysoký: 0.38. Dá sa to vysvetliť oveľa väčším stavovým priestorom pri dosahovaní, kedy model omnoho ťažšie hľadá akcie, ktoré sú posilňované a dostáva sa do finálnej polohy. Preto model potrebuje omnoho viac krokov k akumulovaniu odmeny, taktiež v prípade uchopovania úplne chýba perióda negatívnej odmeny, čo je dané dizajnom funkcie odmeny pre modul dosahovania (rovnicu č. 2). Keďže je rozdiel osí Z ruky a hlavnej osí objektu veľmi malý (modul uchopovania začína s uhlom osí najviac 20° , čo je finálny stav dosahovania) a je škálovaný hodnotou $\pi/2$, model dostáva od začiatku pozitívnu odmenu.

	Dosiahnutie obj.			Uchopenie obj.		
	r	e	r/e	r	e	r/e
Obj. 1	858.18	9110	0.094	247.62	637	0.389
Obj. 2	1284.16	12936	0.099	196.54	584	0.337
Obj. 3	823.53	8652	0.095	291.47	622	0.467
Obj. 4	1198,85	12571	0.095	268,93	645	0.417

Tab. č. 6: Výsledky učenia uchopovania štyroch objektov



Obr. č. 21: Vľavo vývoj akumulovanej odmeny počas fázy dosahovania, vpravo vývoj akumulovanej odmeny počas fázy úchopu.

V tabuľke č. 7 môžeme vidieť výsledky testu, kde bol pridaný šum k pozícii objektu. V tomto štádiu experimentu bolo učenie vypnuté nastavením učiacej rýchlosti na nulu. Miera exploračie sa vždy rovnala pridávanému šumu. Pre každú hodnotu šumu sme spúšťali 20 testov a merali úspešnosť modelu. Za úspešný sme považovali pokus ak rameno skončilo vo finálnom stave a objekt uchopilo. Ak pokus skončil fatálnym stavom, či už pri dosahovaní alebo uchopovaní, pokus sme považovali za neúspešný.

Následne sme nastavili rýchlosť učenia na pôvodné hodnoty a nechali model nech sa naučí uchopiť objekt na novej pozícii. V tomto prípade sme merali počet akcií (krokov) potrebných k preučeniu modelu.

Šum	Úspešnosť	Počet krokov
5 %	95 %	29
10 %	80 %	65
15 %	60 %	161
20 %	45 %	339

Tab. č. 7: Výsledky testov s pridaním šumu

6 Diskusia

Doterajšie výsledky, ktoré sme predstavili v tejto práci nasvedčujú, že náš prístup sa zakladá na dobrých predpokladoch a nedopustili sme sa žiadnej kritickej chyby. Stále však ostáva otvorených niekoľko otázok, ktoré bude treba pre dokončenie modelu zrkadliacich neurónov vyriešiť. Hlavným problémom sa javí navrhnutie mechanizmu vzniku invariantných neurónov, ktoré sa nám zatiaľ nepodarilo nijak vytvoriť. Dosiahli sme iba prvý predstupeň, kedy sa nám podarilo dosiahnuť organizáciu v module aSTS presne opačnú ako je organizácia pSTS, avšak neuróny sú stále perspektívne variantné a to aj pri pokusoch s veľmi malými sieťami, kedy sme predpokladali, že by mohla odlišnosť zlúčených dát z PMC a pSTS podľa typu úchopu prevážiť nad odlišnosťou podľa perspektívy, avšak vstup od PMC sa javí ako nedostatočný na túto úlohu.

Zatiaľ nevyriešeným problémom tiež ostáva ako zabezpečiť selektívnu aktivitu neurónov modulu PMC pri vstupe signálov z oblasti STS cez modul PF, ktorý bol len spomenutý v návrhu celého modelu zrkadliacich neurónov. Nesmieme zabúdať, že iba niektoré z premotorických neurónov vykazujú zrkadliace vlastnosti a toto v našom modeli zatiaľ neriešime. Ďalšou úlohou v budúcnosti bude v zrkadliacich neurónoch vytvoriť asociácie perspektívne variantných neurónov s rovnako zameranými neurónmi v module pSTS.

Ako hlavný problém modelu CRG sa ukázala katastrofická interferencia, keď sme chceli model naučiť uchopovať dva a viac predmetov. Model je schopný sa naučiť uchopovať objekt v novej pozícii a orientácii, avšak stráca ihneď schopnosť uchopiť predchádzajúci objekt. Keďže sa naučené neurónové siete ukladali do súborov, nami navrhované riešenie na predídanie tomuto problému zahŕňa implementáciu klasifikátora, ktorý by na základe vizuálneho vstupu bol schopný určiť o aký objekt sa jedná a následne by nahral súbor s vhodnými neurónovými sieťami. Klasifikátor by mohol pozostávať zo samoorganizujúcej sa mapy (SOM) a doprednej neurónovej siete. SOM by sa organizovala pomocou vstupných obrázkov objektu a následne pomocou učenia s učiteľom by sa dopredná neurónová sieť naučila asociovať výstup zo SOM s korektným modelom uloženým v pamäti. Toto riešenie sme však už neimplementovali a rozhodli sme sa pre naivnejší prístup, v ktorom sme rozdelili stôl, na ktorom boli predmety

položené na štyri segmenty a pre každý segment bol natrénovaný jeden model na uchopenie objektu, ktorý sa v ňom nachádza. Objekty sa samozrejme vždy nachádzali v tom istom segmente pri testoch s pridaným šumom.

Na druhej strane sa pozitívne prejavila schopnosť nášho modelu generalizovať, čo vidieť z výsledkov druhého experimentu, kde aj napriek zašumeným dátam bola stále úspešnosť nad 50% po hodnotu šumu 15%.

Záver

V tejto práci sme sa zamerali na nadizajnovanie, implementáciu a experimentálne testy modulov tvoriacich nami predstavený model zrkadliacich neurónov. Ako je vidno v priehľadovej časti, súčasné výpočtové modely zamerané na systém zrkadliacich neurónov sa viac či menej odkláňajú od experimentálnych východísk vybudovaných na základe neurologických výskumov sensorických a motorických okruhov. Preto sme sa naším návrhom pokúsili aspoň jeden z identifikovaných problémov preklenúť a ponúknuť riešenie v podobe implementácie oblasti vrchnej spánkovej brázdy obsahujúcej reprezentácie biologických pohybov so zahrnutím perspektívy pozorovania. Stanovili sme si tiež za cieľ dosiahnuť vytvorenie perspektívne invariantných reprezentácií a simulovať tak neurálne oblasti s abstraktnejšou reprezentáciou, k čomu sme zatiaľ vykonali prvé potrebné kroky. Pomocou série experimentov sme skúmali vlastnosti jednotlivých modulov pSTS, aSTS, PMC a modelu CRG a dospeli sme k záveru, že náš dizajn a prvé výsledky ukazujú na dobrú cestu za vytýčeným cieľom, ktorým je funkčný model zrkadliacich neurónov pre roboty, ktorý im umožní vytvoriť reprezentácie o svojich akciách a porozumieť akciám druhých.

Použitá literatúra

- [1] I. Farkaš, M. Malý, and K. Rebrová. Mirror neurons – theoretical and computational issues. Technical Report TR-2011-28. Comenius University, Bratislava, Slovakia, 2011.
- [2] M. A. Goodale, A. D. Milner. Separate visual pathways for perception and action. *Trends Neurosci.* 15 (1): 20–5, 1992.
- [3] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3(2):131–141, 1996.
- [4] G. Hein, R.T. Knight. Superior temporal sulcus—It’s my area: or is it? *J. Cogn. Neurosci.* 20:1–12, 2008.
- [5] D. Perrett, M. Oram, M. Harries, R. Bevan, J. Hietanen, P. Benson, and S. Thomas. Viewer-centred and object-centred coding of heads in the macaque temporal cortex. *Experimental Brain Research*, 86(1):159–173, 1991.
- [6] T. Jellema and D. Perrett Neural representations of perceived bodily actions using a categorical frame of reference in *Neuropsychologia*, vol. 44, pp. 1535–1546, 2006.
- [7] G. Rizzolatti et al. Functional organization of inferior area 6 in the macaque monkey: II. Area F5 and the control of distal movements. *Exp. Brain Res.* 71, 491–507 (1988).
- [8] G. Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental brain research*, 91(1):176–180, 1992.
- [9] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Rev. Neuroscience*, 2:661–670, 2001.
- [10] G. Rizzolatti and C. Sinigaglia. The functional role of the parieto-frontal mirror circuit: interpretations and misinterpretations. *Nature Reviews Neuroscience*, 11(4):264–74, 2010.
- [11] G. Cohen-Seat, H. Gastaut, J. Faure and G. Heuyer. Etudes expérimentales de l’activité nerveuse pendant la projection cinématographique. *Rev. Int. Filmol.* 5, 7–64 , 1954.
- [12] H. J. Gastaut and J. Bert. EEG changes during cinematographic presentation. *Electroencephalogr. Clin. Neurophysiol.* 6, 433–444, 1954.

-
- [13] V. Caggiano, L. Fogassi, G. Rizzolatti, P. Thier, and A. Casile. Mirror neurons differentially encode the peripersonal and extrapersonal space of monkeys, *Science*, vol. 324, no. 5925, pp. 403–406, 2009.
- [14] G. Tessitore, R. Prevede, E. Catanzariti, and G. Tamburrini. From motor to sensory processing in mirror neuron computational modelling. *Biological cybernetics*, 103(6):471–485, 2010.
- [15] M. Ito and J. Tani. On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, 12(2), 93–115, 2004.
- [16] E. Borenstein and E. Ruppin. The evolution of imitation and mirror neurons in adaptive agents. *Cognitive Systems Research*, 6(3), 2005.
- [17] E. Oztop, M. Kawato, and M. Arbib. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006.
- [18] E. Oztop and M. Arbib. Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, 87:116–140, 2002.
- [19] J. Bonaiuto, E. Rosta and M. A. Arbib. Recognizing invisible actions, workshop on modeling natural action selection. Workshop on modeling natural action selection, Edinburgh., 2005.
- [20] D. Kraft, R. Detry, N. Pugeault, E. Baseski, F. Guerin, J. Piater, and N. Kruger. Development of object and grasping knowledge by robot exploration. *IEEE Transactions on Autonomous Mental Development*, 2010.
- [21] E. Oztop, M. Kawato. *Sensorimotor Control of Grasping: Physiology and Pathophysiology*. Cambridge University Press, Ch. Models for the control of grasping, 2009.
- [22] R. Prevede, G. Tessitore, M. Santoro, and E. Catanzariti. A connectionist architecture for view-independent grip-aperture computation, *Brain Research*, vol. 1225, pp. 133-145, 2008.
- [23] E. Oztop, N. Bradley, and M. Arlib, Infant grasp learning: A computational model, *Exp. Brain Res.*, vol. 158, pp. 480–503, 2004.

-
- [24] F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386–408, 1958.
- [25] S. Haykin. Neural Networks: A comprehensive foundation. Upper Saddle River, NJ, USA: Prentice Hall, pp. 59-65, 1999.
- [26] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning Internal Representations by Error Propagation. In: J.L. McClelland, D.E. Rumelhart, and PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations, chapter 8, MIT Press, Cambridge, 1986.
- [27] R.C. O'Reilly. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural computation*, 8(5):895–938, 1996.
- [28] T. Kohonen. Self-organizing Maps. Springer, 1997.
- [29] A. M. Turing. The Chemical Basis of Morphogenesis in *Philosophical Transactions of the Royal Society*, Series B, **237**, 5-72, 1952.
- [30] M. Strickert and B. Hammer. Merge SOM for temporal data. *Neurocomputing*, 64:39–71, 2005.
- [31] T. Voegtlin. Recursive self-organizing maps. *Neural Networks* 15 979-992, 2002.
- [32] B. Hammer, A. Micheli, A. Sperduti, M. Strickert. Recursive self-organizing network models. *Neural Networks* 17 (2004a) 1061-1085.
- [33] P. Vančo and I. Farkaš. Experimental comparison of recursive self-organizing maps for processing tree-structured data. *Neurocomputing*, 73(7-9):1362–1375, 2010.
- [34] H. Van Hasselt, H. Reinforcement learning in continuous state and action spaces in *Reinforcement Learning: State of the Art*, eds M. Wiering and M. van Otterlo (Berlin: Springer), 207–251, 2012.
- [35] K. Rebrová, I. Farkaš: Robotický model systému zrkadliacich neurónov. In Kelemen J., Nahodil P. (eds.), *Kognice a umělý život XII*, Slezská univerzita. 231-238. 2012.

-
- [36] K. Rebrová, M. Pecháč, and I. Farkaš. Towards a robotic model of the mirror neuron system. Submitted to ICDL-EpiRob2013, 2013.
- [37] L. Zdechovan. Modeling the object grasping using the neural networks and icub robotic simulator (in Slovak), Master's thesis, Comenius University in Bratislava, 2012.
- [38] PCL. Point Cloud Library <http://pointclouds.org/> Online:accessed 07-Apr-2013.
- [39] K. Rebrová and I. Farkaš. Bidirectional activation-based neural network learning algorithm, submitted to ICANN 2013 .
- [40] Golem API. <http://www.robot-golem.org/> Online:accessed 31-Mar-2013.
- [41] ROS: Robot operating system. <http://www.ros.org/> Online:accessed 20-May-2013.
- [42] G. Marsaglia and T. A. Bray. A convenient method for generating normal variables. SIAM Rev. 6, 260–264, 1964.

Príloha A – CD