

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

BAKALÁŘSKÁ PRÁCE



Šimon Fojtů

**Aplikace evolučních algoritmů v navigační úloze
mobilního robotu**

Katedra kybernetiky

Vedoucí bakalářské práce: Ing. Jan Faigl

Praha, 2009

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, SW, projekty atd.) uvedené v příloženém seznamu.

V Praze dne 30. 6. 2009

..... Fojtík

podpis

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Šimon Fojtů

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný

Obor: Kybernetika a měření

Název tématu: Aplikace evolučních principů v navigační úloze mobilního robotu

Pokyny pro vypracování:

1. Seznamte se s úlohou „homing navigation“ a přístupy řešení.
2. Seznamte se s robotickou platformou MORBOT příslušejícím základním softwarovým vybavením.
3. Seznamte se s robotickým prostředím Player/Stage.
4. Navrhněte a implementujte algoritmus řešící úlohu navigace založený na evolučních principech.
5. Algoritmus experimentálně ověřte v simulátoru a na reálném robotu MORBOT.

Seznam odborné literatury:


- [1] Floreano, D.; Mondada, F.: Evolution of Homing Navigation in a Real Mobile Robot. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26:396-407, 1996.
- [2] <http://playerstage.sf.net>.
- [3] Grimmer, V.: Bakalářská práce: Platforma pro výuku mobilní robotiky. ČVUT v Praze, FEL, 2008.
- [4] Szücsová, H.: Bakalářská práce: Podpurný software pro výuku mobilní robotiky. ČVUT v Praze, FEL, 2008.

Vedoucí bakalářské práce: Ing. Jan Faigl

Platnost zadání: do konce zimního semestru 2009/2010


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 10.12.2008

Abstrakt

Práce se zabývá evolučním vývojem diskretní rekurentní umělé neuronové sítě pro řízení autonomního mobilního robota. Standardním genetickým algoritmem je hledáno řešení problému obstacle avoidance. Vývoj probíhá v několika specifických prostředích v simulátoru Player/Stage. Nalezené řídicí systémy jsou zkoumány z hlediska úspěšnosti ve všech prostředích. Evolucí nového řídicího systému s rychlejší frekvencí vyčítání dat z infračervených senzorů je analyzována schopnost adaptace na konkrétní konfiguraci robota. Robustnost je testována přenesením nejlepšího nalezeného řídicího systému na reálný mobilní robot Morbot v robotické aréně SyRoTek. Úspěšnost navigace v reálném prostředí je minimálně 60 %. Dosažené výsledky ukazují, že vyvinutý řídicí systém je adaptován na konkrétní možnosti robota a prostředí tak, aby maximalizoval svou hodnotu fitness funkce, a zároveň je schopen správně řídit reálného robota. Z toho lze usuzovat, že je možné vyvinout neuronovou síť v simulátoru a pro řízení reálného robota použít až nalezené řešení.

Abstract

The thesis deals with the evolution of a discrete-time recurrent artificial neural network to control an autonomous mobile robot. Standard genetic algorithm is employed to solve the obstacle avoidance problem. Evolution takes place in several specific simulated environments in Player/Stage. The obtained solutions are analyzed in all environments and compared with each other. The ability of adapting to particular robot configuration is explored by evolving a new control system with higher frequency of sensors readings. Robustness is tested by placing the best control system into the real mobile robot Morbot in the robotic arena SyRoTek. Reliability of its navigation is at least 60 %. Achieved results show that evolved control system is well adapted to specific possibilities of the robot and the environment, so that it maximizes its fitness function. It is also able to control the real robot properly. From that a conclusion might be drawn, that it is possible to evolve an artificial neural network in a simulated environment and to use the found solution for controlling of the real robot.

Především děkuji vedoucímu své bakalářské práce Ing. Janu Faiglovi za čas a cenné rady, které podstatně přispěly k výsledné podobě a obsahu této práce. Dále děkuji členům skupiny Inteligentní a Mobilní Robotiky za umožnění experimentů s reálným robotem. A velmi děkuji své přítelkyni Štěpánce za trpělivost a psychickou podporu.

Obsah

Úvod	1
1 Definice úlohy	6
1.1 Robot	7
1.2 Nastavení genetických algoritmů	8
1.3 Zaměření experimentů	9
2 Experimenty	10
2.1 Úvod	10
2.2 Aplikace řídicí evoluci	13
2.3 Fitness funkce	13
2.4 Vývoj řídicích systémů	14
2.5 Porovnání nalezených chování	21
2.6 Perioda vzorkování	23
2.7 Experiment s reálným robotem	24
2.8 Shrnutí experimentů	27
3 Závěr	29
A Obsah CD	32

Seznam tabulek

2.1	Parametry genetických algoritmů experimentu Obstacle avoidance.	10
2.2	Průměry hodnot fitness funkce nejlepších robotů v jednotlivých prostředích.	21
2.3	Porovnání průměrných rychlostí robotů s různými periodami vzorkování.	23
A.1	Adresářová struktura na CD.	32

Seznam obrázků

1.1	Umístění a dosah infračervených senzorů robota Morbot.	6
1.2	Schéma umělé neuronové sítě.	7
2.1	Prostředí použita v experimentech.	11
2.2	Robot v prostředí <i>Tunel</i>	15
2.3	Vývoj hodnoty fitness v prostředí <i>Tunel</i>	15
2.4	Robot v prostředí <i>Osmička</i>	16
2.5	Vývoj hodnoty fitness v prostředí <i>Osmička</i>	16
2.6	Robot v prostředí <i>Okruh</i>	17
2.7	Vývoj hodnoty fitness v prostředí <i>Okruh</i>	18
2.8	Robot v prostředí <i>Kostky</i>	19
2.9	Vývoj hodnoty fitness v prostředí <i>Kostky</i>	19
2.10	Robot v prostředí <i>SyRoTek</i>	20
2.11	Vývoj hodnoty fitness v prostředí <i>SyRoTek</i>	20
2.12	Trajektorie robota vyvinutého v prostředí <i>Tunel</i> v ostatních prostředích. .	22
2.13	Převodní charakteristika IR dálkoměru.	24
2.14	Linearizovaná převodní charakteristika.	25
2.15	Reálný robot v testovacím prostředí.	26

Úvod

Tato bakalářská práce je inspirována experimenty D. Floreana a F. Mondady [4], které prováděli na mobilním robotovi¹ řízeném umělou neuronovou sítí. Autoři ukázali, že je možné evolučními metodami vyvinout složité chování, které není předem nijak specifikováno, ale spíše plyne z interakce mobilního robota s prostředím. Toto chování navíc využívá vhodných vlastností prostředí a robota ke splnění zadaného úkolu. Kromě klasického vyhýbání se překážkám vyvinuli i takové chování, kdy robot dokázal nalézt v prostředí zdroj energie a naučil se k němu periodicky vracet. Tohoto složitějšího chování dosáhli zjednodušením hodnotící funkce a přidáním nových vlastností do prostředí a robota [4].

Cílem této práce je experimentální ověření prvního z výše zmíněných experimentů a následné detailní zkoumání a porovnávání nalezených chování. Snahou je nalézt řídicí systém robota, který by byl schopen vyhýbat se překážkám, což je základním předpokladem pro bezpečnou navigaci mobilního robota. V literatuře bývá tato vlastnost označována jako *obstacle avoidance*. Jednou z otázek, na které budou hledány odpovědi, je úspěšnost výsledného chování v jiných podmínkách, než při kterých bylo vyvinuto. Tedy zkoumání, zda se bude robot v podobném (či úplně jiném) prostředí chovat úspěšně nebo bude jeho úspěšnost horší. Tím lze zjistit míru obecnosti jeho chování. V prvním případě lze usuzovat, že vyvinuté chování je univerzálněji použitelné, ve druhém případě je možné usoudit, že došlo spíše k adaptaci na konkrétní prostředí.

Autonomní mobilní roboti mají s biologickými agenty (jednobuněčnými organismy, živočichy a lidmi) mnoho společných vlastností. Svými senzory vnímají prostředí, ve kterém se nacházejí, a aktuátory na něj reagují a ovlivňují jej. Jsou schopni se přizpůsobit svému prostředí a vhodně je využít ke splnění svých cílů. Z této podobnosti lze usuzovat, že je k vytvoření řídicího systému robota výhodné využít metody, které mají svůj základ v přírodě. Řídicí systém živočichů (nervová soustava, centrální nervový systém) se vyvíjí spolu s jejich fyzickou stavbou tak, aby se dokázali přizpůsobit podmínkám okolního světa a také jej vhodně využít ve svůj prospěch.

Adaptace jedince v přírodě probíhá ve dvou fázích. Skládá se z evoluce a učení, tedy *fylogenetické* a *ontogenetické* adaptace, z nichž každá působí v jiném časovém horizontu. Evoluce je proces výběru, reprodukce a změn populace jedinců, kteří se od sebe liší svou genetickou informací. V průběhu svého života se jedinec nemění, a proto dokáže evo-

¹ I když je možné používat slovo robot v neživotné podobě, je v této práci dána přednost tvaru životnému. Hlavním důvodem je inspirace živými tvory a biologickými principy, které jsou využívány při vývoji jeho řídicího systému.

luce reagovat pouze na pomalé změny, které ovlivňují více generací. V přírodě se takto přizpůsobují kromě fyzické stavby jedinců i jejich pudy a instinkty. Učení naopak umožňuje jedincům reagovat na změny, které jsou rychlé a nezachytitelné z generačního hlediska, protože ovlivňuje jedince v průběhu jeho života [8].

Dále je vhodné zdůraznit, že robot, který dokáže pracovat, aniž by byl připojen k externímu zdroji energie či vzdáleně řízen počítačem, není nutně autonomní. Ačkoliv je taková nezávislost často velmi výhodná vlastnost, autonomnost je spíše chápána jako schopnost robota přizpůsobit se prostředí nalezením optimálních řešení, vytvořením vhodného řídicího systému, definováním vlastních cílů a popřípadě i monitorováním svého stavu. Tyto schopnosti navíc nemají být předem definované, ale měly by být výsledkem interakce mezi robotem a jeho prostředím [3], což odpovídá principu vývoje sensorických a motorických systémů biologických agentů. Tyto systémy se spoluvyvíjely s jejich chováním v závislosti na prostředí, kterému se potřebovali přizpůsobit [2]. Z tohoto hlediska tedy může být výhodné ponechat takovou možnost i mobilnímu robotovi, aby si sám zvolil senzory a informace, které využije, a používal vhodné aktuátory k provedení svých cílů.

Genetické algoritmy

Softwarovou implementací evolučních principů jsou genetické algoritmy (GA). Jsou to stochastické optimalizační metody založené na darwinovském principu evoluce [6]. Pracují s populací jedinců, z nichž každý reprezentuje jedno řešení, které je vhodně zakódováno do jeho genotypu. Volba správné reprezentace genomu a operací s ním je velmi důležitá pro správnou funkci GA. K ohodnocení konkrétního jedince slouží tzv. *fitness funkce*, což je libovolná, obvykle kladná, funkce, která danému jedinci přiřadí nějakou hodnotu. Průběh hledání řešení pak spočívá v inicializaci počáteční (nulté) generace jedinců—buď náhodně nebo podle nějaké apriorní znalosti řešení—ze které jsou určitým způsobem vybráni jedinci, kteří pak podstoupí křížení a mutaci, čímž vznikne generace nová. Tento postup se opakuje tak dlouho, dokud není splněna ukončovací podmínka, která může zastavit evoluci při dosažení daného počtu generací, při dosažení uspokojivého řešení nebo pokud už nedochází k vylepšování nalezeného řešení. Při hledání řešení genetickými algoritmy hraje podstatnou roli náhodná složka, z čehož plyne, že nalezené řešení dané úlohy může být pokaždé jiné. Obecný tvar GA je zapsán v pseudokódu v Algoritmu 1 [7].

Fitness funkce

Fitness funkce hraje při vývoji řídicího systému robota klíčovou roli, neboť je jediným měřítkem mezi jednotlivými roboty při evoluci. Existují dva přístupy, jak tuto funkci vytvořit [4].

První možnost je pečlivě ji navrhnout tak, aby přímo určovala očekávané chování a potlačila ty vlastnosti chování, které nejsou žádoucí. Tedy přesně na míru konkrétnímu robotu a prostředí experimentálně nalézt vhodnou funkci. Cílem není splnění nějakého obecného cíle, ale spíše co nejpřesnější specifikace požadovaného chování. Nevýhodou to-

Algoritmus 1: Zápis genetického algoritmu v pseudokódu.

```
1 t = 0;
2 inicializace G(t);
3 vyhodnocení G(t);
4 while not (ukončovací podmínka) do
5   | t++;
6   | selekce G(t) z G(t-1);
7   | změna G(t);
8   | vyhodnocení G(t);
9 end
```

hoto přístupu je, že i pro jednoduchý úkol je potřeba provést empirické pokusy pro konkrétní prostředí a robota, kterými je získána požadovaná fitness funkce. Také lze těžko označit tohoto robota jako autonomního, jelikož jeho chování je striktně předepsáno fitness funkcí.

Druhý přístup používá jednoduchou fitness funkci a bere ji pouze jako kritérium přežití jedince. Specifikuje cíl, kterého má robot dosáhnout, ale už ne způsob. Tím se důležitost experimentátora snižuje a robot může využít možnosti, které mu nabízí prostředí.

Z tohoto rozdělení vyplývá, že první způsob je vhodný pro jednoduché úkoly či pro případ, kdy je přesně známo požadované chování. Druhý způsob nechává více prostoru pro adaptaci robota a může vést ke složitým chováním i s použitím jednoduché fitness funkce.

Operátory genetických algoritmů

Důležitou roli v GA hraje selekce neboli výběr jedinců z předchozí generace do nové. V přírodě tuto funkci představuje přirozený výběr [1]. To je proces, při kterém se dědičné znaky umožňující organismu přežít a rozmnožit se, vyskytují v následujících generacích častěji. Jedinci, kteří mají (jakkoli malou) výhodu oproti ostatním, mají také větší šanci, že přežijí a budou mít více potomků. Naopak jedinci s vlastnostmi, které je vzhledem k ostatním poškozují, s větší pravděpodobností vymřou. Porovnávání jedinců v přírodě probíhá jejich interakcí mezi sebou navzájem a prostředím, ve kterém žijí. V GA je pro tento účel vytvořena fitness funkce. Z principu selekce je patrné, že pravděpodobnost výběru jedince do nové generace je přímo úměrná jeho hodnotě fitness. Avšak i jedinci s průměrnou či podprůměrnou hodnotou fitness mohou být vybráni, čímž se udržuje diverzita populace.

Diverzita neboli různorodost populace je velmi důležitá, protože brání uváznutí vývoje v lokálním optimu. Její důležitost se projeví také při změně prostředí. Jedinci, kteří jsou úspěšní v jednom prostředí, mohou být v jiném prostředí naprosto neúspěšní a naopak původně podprůměrní zde mohou ukázat své schopnosti. Díky tomu dojde k zachování dané populace, jelikož homogenní populace by velmi pravděpodobně zanikla, neboť její readaptace na nové prostředí by probíhala velice pomalu. K udržení diverzity slouží i genetické operátory křížení a mutace.

Operátor křížení pracuje obvykle se dvěma jedinci, jejichž chromozomy určitým způso-

bem zkombinuje a tím vytvoří dva potomky. Tak může vzniknout jedinec, který má lepší vlastnosti než oba jeho rodiče. V homogenní populaci tento operátor nemá význam, protože křížení stejných nebo podobných jedinců nepřinese žádný nový prvek. Druhým operátorem je operátor mutace, který pracuje s jednotlivými chromozomy a s určitou pravděpodobností mutuje — mění jejich prvky. Smyslem tohoto operátoru je přinášet do populace nové vlastnosti a zabránovat předčasné konvergenci. Pravděpodobnost křížení bývá obvykle 60–100 %, pravděpodobnost mutace je výrazně menší 5–15 % [7].

Vlastnosti genetických algoritmů

Výhodou použití genetických algoritmů při vývoji řídicího systému mobilního robota je kromě automatického hledání optimálního řešení také skutečnost, že takto nalezené řešení může být lepší než analyticky navržené. Náhodná řešení z první generace procházejí evolucí, na jejímž konci se pak může objevit řešení, které plně využívá konkrétní konfiguraci a možnosti robota a prostředí. Navíc je možné, že bude nalezeno takové řešení, které je jednodušší nebo využívá jiné možnosti a vlastnosti, než bylo očekáváno.

Nevýhodou tohoto přístupu je velká časová náročnost experimentu, jelikož vyhodnocení fitness funkce každého jedince probíhá během jeho jízdy v prostředí. V závislosti na délce doby jízdy robota, počtu jedinců v generaci a počtu generací může vývoj trvat několik dnů až týdnů. Tato nevýhoda může být částečně potlačena prováděním experimentu v simulátoru, kde lze například nastavit rychlejší běh času a tím zkrátit dobu evoluce².

V počátečních generacích nejezdí roboti tak, jak by se očekávalo, narážejí do zdí a vykazují další neoptimální chování. To je způsobeno tím, že nultá generace je vytvořena náhodně a teprve další generace se postupně blíží nejlépe vyhovujícímu chování. Dobrou experimentu a častými kolizemi jsou kladeny vysoké požadavky na hardwarovou robustnost robota, což může být dalším důvodem k použití simulátoru. Další vlastností je, že fitness funkcí sice lze výrazně ovlivnit chování robota, ale nelze určit přesný výsledek evoluce a řešení může být jiné, než bylo původně zamýšleno.

Neuronové sítě

Umělé neuronové sítě jsou jako řídicí systém autonomních mobilních robotů vhodné z několika důvodů [3], v následujícím seznamu jsou uvedeny hlavní výhody jejich použití.

- Mají schopnost generalizovat — na podobné vstupy odpovídají podobnými výstupy.
- Jsou odolné proti šumu, což je výhodné při použití u reálných robotů, kde jsou signály ze sensorů zatíženy šumem.
- Dokáží se přizpůsobit vlastnostem robota — buď mohou změnit svou strukturu tak, aby co nejlépe využily schopnosti robota, nebo mohou aktivně vybrat ty aktuátory a signály ze sensorů, které jsou nejvhodnější pro daný úkol.

²Rychlejší průběh simulace ovšem může být dosažen za cenu nižší kvality.

- Vhodně reagují i na případnou asymetrii robota — různě velká kola, rozestavení senzorů apod.
- Jsou robustní — chybějící spoje či částečně nefunkční hardwarové komponenty příliš neovlivní výsledné chování.
- Pokud je použita struktura neuronové sítě s rekurentními a laterálními vazbami a s nelineárními přenosovými funkcemi, je teoreticky možné získat velice silný řídicí systém, který se dokáže vyrovnat se složitými situacemi reálného světa.
- Parametry neuronových sítí, tedy váhy vazeb mezi neurony a prahy jednotlivých neuronů, lze jednoduše zakódovat do genomu jedince.

Následující kapitola popisuje použitý model robota, princip jeho řízení a nastavení genetických algoritmů pro vývoj neuronové sítě. Návrhu fitness funkce, vývoji řídicího systému a jeho zkoumání je věnována kapitola 2.

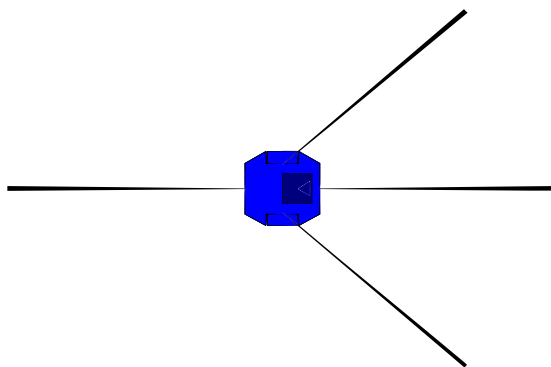
Kapitola 1

Definice úlohy

Práce se soustředí především na experimenty a zkoumání jejich výsledků. Snahou je získat řídicí systém — neuronovou síť, která je schopna kontrolovat pohyb mobilního robota v prostředí tak, aby se vyhýbal překážkám. Základem je návrh fitness funkce, která povede k nalezení optimálního řídicího systému. Evoluce probíhá současně v několika prostředích a z každého je získáno jedno řešení.

Chování řídicích systémů je nejprve detailně zkoumáno v původních a poté v prostředích ostatních. Porovnáním úspěšnosti jednotlivých chování mezi sebou je určen řídicí systém, který v dané množině prostředí kontroluje robota nejlépe. Schopnost adaptace na konkrétní konfiguraci robota je zkoumána při vývoji řídicích systémů dvou robotů, kteří se liší pouze frekvencí vyčítání dat ze senzorů. Univerzálnost řešení, které má v porovnání s ostatními nejvyšší hodnotu fitness funkce, je testována přenesením vyvinutého systému na reálného robota.

V následujících oddílech je popsán použitý model robota, vysvětlen princip jeho řízení umělou neuronovou sítí a stručně také objasněn princip konkrétních genetických operátorů.

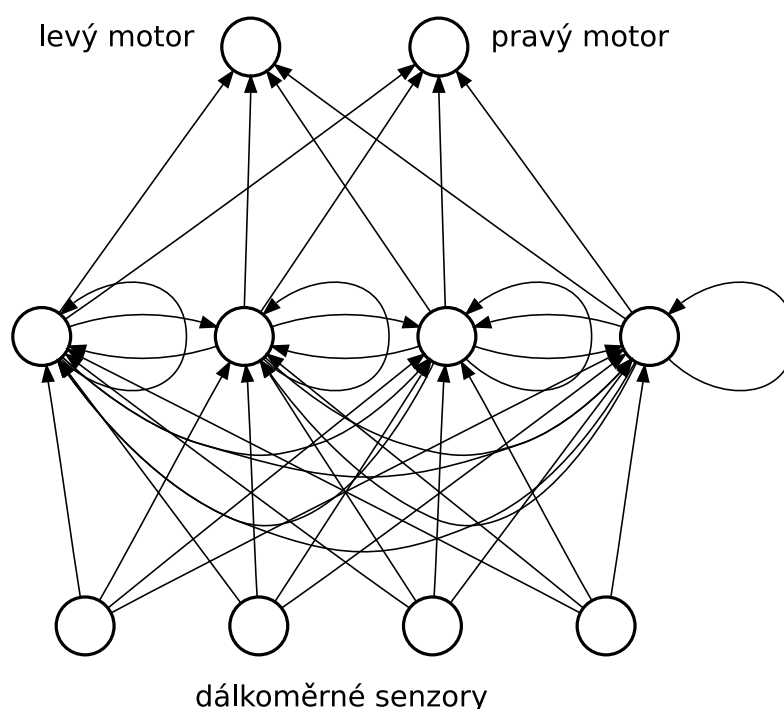


Obrázek 1.1: Umístění a dosah infračervených senzorů robota Morbot.

1.1 Robot

V experimentech je použit model robota (resp. reálný robot) Morbot [5, 12] s diferenciálním podvozkem, který má poměrně vysokou manévrovací schopnost a lze jej jednoduše řídit. Dokáže se otáčet na jednom místě, a proto má možnost snadno odjet od konvexních i konkávních překážek. V použité konfiguraci disponuje robot čtyřmi infračervenými dálkoměry, které mohou být natočeny do libovolných směrů. Nákres rozložení sensorů je na obrázku 1.1. Dálkoměrné senzory umožňují robotovi měřit jeho vzdálenost od překážek a zdi.

Prostředí je představováno dvourozměrnou uzavřenou oblastí se statickými překážkami. Vývoj řídicích systémů probíhá v simulátoru Player/Stage, experiment s reálným robotem v robotické aréně SyRoTek.



Obrázek 1.2: Schéma umělé neuronové sítě.

Řídicí systém je realizován umělou neuronovou sítí. Její struktura se během experimentů nemění, zobrazena je na obrázku 1.2. Jedná se o třívrstvou síť s diskretní rekurentní vazbou ve skryté vrstvě. To znamená, že výstup všech neuronů skryté vrstvy je v následujícím časovém kroku přiveden jako vstup zpět do skryté vrstvy. Počet neuronů v síti je konstantní. Na vstupní vrstvu jsou přivedeny signály z dálkoměrných sensorů — každý neuron je napojen na jeden sensor — obsahuje tedy čtyři neurony. Skrytá vrstva je tvořena čtyřmi neurony a výstupní dvěma. Každý výstupní neuron ovládá rychlost jednoho motoru.

Jako transformační funkce neuronů je použita upravená sigmoidální funkce (1.1), která má obor hodnot $\langle -1; 1 \rangle$.

$$P(t) = \frac{2}{1 + e^{-t}} - 1 \quad (1.1)$$

Signál z infračervených sensorů není nijak upravován, rozsah jeho hodnot je v intervalu $\langle 0, 15; 0, 8 \rangle$ ¹, kde vyšší hodnoty znamenají vzdálenější překážky.

Oba signály z výstupní vrstvy neuronové sítě jsou přímo napojeny na vstupy motorů tak, že hodnota -1 odpovídá maximální rychlosti otáčení motoru jedním směrem, 0 zastavenému motoru a hodnota 1 maximální rychlosti otáčení směrem druhým.

Pro věrnější simulaci reálného prostředí je k vstupním signálům ze sensorů superponován bílý šum o amplitudě 0,2. Pro infračervené senzory to znamená rušení o hodnotě přibližně 30 %. Jeho vliv na chování robotů je probrán v závěru experimentů.

1.2 Nastavení genetických algoritmů

Parametry evoluce uvedené v článku Floreana a Mondady [4] nepřinesly očekávané výsledky, a proto jsou v experimentech použity parametry poněkud odlišné. V souladu s článkem je použit standardní genetický algoritmus (SGA) s konstantním počtem jedinců v generaci, ruletová selekce, jednobodové křížení, biased mutace a lineární škálování. Navíc je využita možnost přenést několik nejlepších jedinců přímo do další generace tzv. *elitismus*.

Ruletová selekce vybírá jedince s pravděpodobností přímo úměrnou jeho fitness. Necht $\Phi(i)$ je hodnota fitness jedince i , pak pravděpodobnost, že bude vybrán, je rovna

$$p(i) = \frac{\Phi(i)}{N \sum_{j=1} \Phi(j)},$$

kde N je počet jedinců v generaci.

Jednobodové křížení pracuje tak, že je náhodně vybrán křížící bod, ve kterém se oba rodičovské genomy rozdělí. První potomek pak vznikne spojením levé části genomu prvního rodiče a pravé části rodiče druhého. U druhého potomka je tomu naopak.

Biased mutace spočívá v přičtení náhodné hodnoty z intervalu $\langle -\Delta m; \Delta m \rangle$ ke každému prvku genomu s pravděpodobností danou pravděpodobností mutace.

Pokud jsou změny způsobené operátory křížení a mutace příliš velké a příliš časté, dojde k degeneraci genetického algoritmu na pouhé náhodné prohledávání prostoru možností. Aby k tomu nedošlo, je třeba pro konkrétní problém experimentálně nalézt vhodné pravděpodobnosti křížení a mutace a také odpovídající implementace těchto operátorů.

V této práci jsou použiti jedinci pouze s jedním chromozomem, takže je možné zjednodušeně označit genom jako chromozom. Genom každého jedince, představovaný konkrétním nastavením umělé neuronové sítě, je zakódován jako pole reálných čísel. Každé z těchto

¹Tyto hodnoty jsou dány nastavením modelu robota Morbot.

čísel odpovídá váze jedné vazby nebo velikosti prahu neuronu. Délka tohoto pole závisí na počtu neuronů neuronové sítě, takže se během evoluce nemění.

Při inicializaci neuronové sítě v nulté generaci se všechny hodnoty vah a prahů nastaví na náhodné číslo z intervalu $\langle -0,5; 0,5 \rangle$. Maximum těchto hodnot je v rozsahu $\langle -100; 100 \rangle$, avšak je nutno podotknout, že této hranice nemůže žádná vazba v průběhu experimentů dosáhnout. Z nastavení experimentů totiž plyne, že maximální hodnota, kterou může vazba či práh nabývat, je $\pm 50,5$ (pro 100 generací, maximální velikost mutace $\Delta m = 0,5$ a maximální počáteční hodnotu $0,5$), což znamená, že lze uvažovat hodnoty vazeb a prahů jako neomezené. Počáteční nastavení vah a prahů umělé neuronové sítě blízko nule je výhodné především pro experimenty na reálných robotech. Neuronová síť totiž pro libovolné vstupní hodnoty negeneruje na svém výstupu vysoké hodnoty a robot se tudíž pohybuje pomalu. To je vzhledem k pravděpodobnosti kolizí jedinců na počátku evoluce velmi užitečná vlastnost, jelikož se tak šetří hardware robota. V dalších generacích se vlivem mutace tyto hodnoty zvětšují a roboti se pohybují rychleji, avšak pravděpodobnost kolize je již nižší.

Ohodnocení každého jedince probíhá v několika krocích. Začíná dekódováním jeho genomu do neuronové sítě, která je připojena k sensorům a motorům robota. Data ze sensorů jsou čtena s periodou 300 ms a ve stejném intervalu jsou aktualizovány i povely pro motory. Robot je určitou dobu² ponechán v prostředí, aby se volně pohyboval, zatímco je měřena jeho hodnota fitness (detaily jsou uvedeny u jednotlivých experimentů). Mezi jednotlivými jízdami jsou oba motory na 0,5 s roztočeny náhodnými rychlostmi. Tím je v průměru dosaženo podobných počátečních podmínek, protože se tímto způsobem může robot dostat dál od překážek, a také je omezen vliv mezi následujícími jedinci.

Jelikož fitness funkce nehodnotí přímo nastavení neuronové sítě, ale až její projev v chování robota, musí být měřena z jeho pohybů a prováděných akcí. Tím je způsobena nepřesnost získané hodnoty fitness funkce, protože z časových důvodů nelze vyzkoušet chování robota ve všech situacích a za všech počátečních podmínek.

1.3 Zaměření experimentů

Následující kapitola se zabývá popisem provedených experimentů a dosaženými výsledky. Experimentálně je pro různá prostředí hledáno takové nastavení umělé neuronové sítě, se kterým by byl robot schopen splnit úlohu vyhýbání překážkám v uzavřeném statickém prostředí. Nalezené řešení neboli chování robota pro konkrétní prostředí je následně zkoumáno v ostatních prostředích. Nejlepší nalezený řídicí systém je také zkoumán z hlediska robustnosti při přenesení na reálného robota.

²Určitý počet časových kroků.

Kapitola 2

Experimenty

2.1 Úvod

Cílem experimentů je nalezení takového chování, při kterém by robot v prostředí jezdil co nejvíce rovně a vyhýbal se překážkám tzv. obstacle avoidance. Existuje mnoho analytických metod, kterými lze navrhnout řídicí systém robota, jenž bude toto chování také vykazovat. Ale i přesto se může evolučně vyvinutý řídicí systém vypořádat se symetrickými nebo konkávními překážkami lépe než jiné řídicí systémy [4] (například tím, že vhodně využije rekurentní vazby v neuronové síti). Také se dokáže lépe přizpůsobit prostředí a konfiguraci robota.

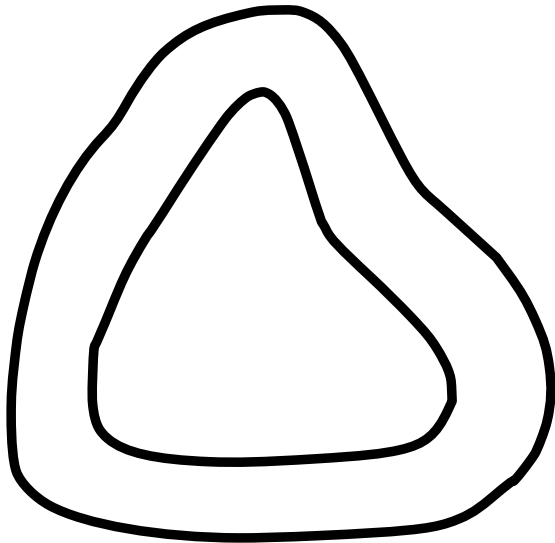
K návrhu fitness funkce pro vývoj řídicího systému byl použit první z přístupů nastíněných v úvodu. Funkce je tedy vytvořena tak, aby co nejkonkrétněji určovala výsledné chování robota. Nicméně i přes toto omezení vykazuje nalezený řídicí systém řadu dalších zajímavých emergentních vlastností, které robotovi zjednodušují splnění určeného úkolu, aniž by byly v této funkci specifikovány. Tyto vlastnosti jsou podrobněji popsány v závěru experimentů.

Konkrétní číselné hodnoty jednotlivých parametrů genetických algoritmů jsou uvedeny v tabulce 2.1.

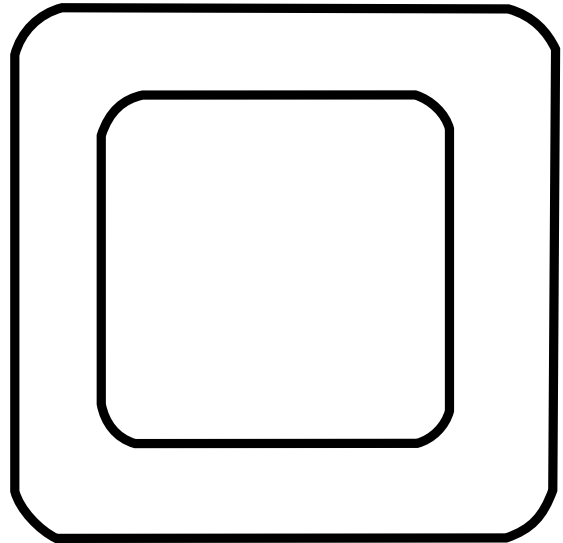
velikost populace	100
počet generací	100
délka chromozomu	42
délka života (maximální počet kroků)	50
velikost mutace Δm	0,5
pravděpodobnost křížení	10 %
pravděpodobnost mutace	20 %

Tabulka 2.1: Parametry genetických algoritmů experimentu Obstacle avoidance.

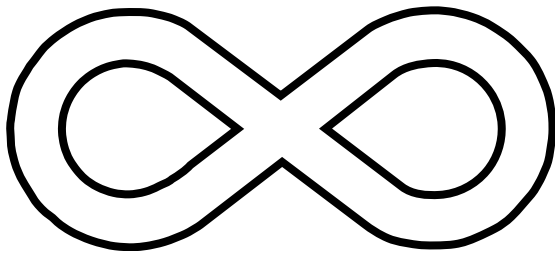
Pro evoluci je připraveno pět různých prostředí. Tři z nich, obrázky 2.1(a), 2.1(c) a 2.1(b), jsou okruhy různých tvarů a nestejně širokých cest a dvě, obrázky 2.1(d) a 2.1(e),



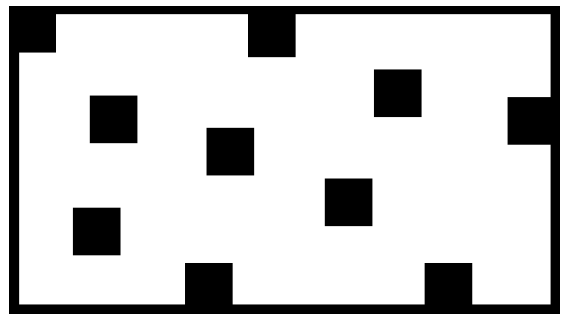
(a) Prostředí *Tunel*.



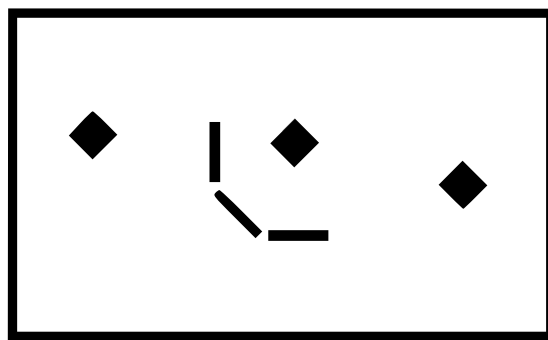
(b) Prostředí *Okruh*.



(c) Prostředí *Osmička*.



(d) Prostředí *Kostky*.



(e) Prostředí *SyRoTek*, převzato z projektu SyRo-Tek [11].

Obrázek 2.1: Prostředí použitá v experimentech.

představují místnosti s náhodně rozmístěnými překážkami rozmanitých tvarů. Evoluční běh je pro každé prostředí spuštěn dvakrát a výsledky jsou pak počítány z obou běhů. Nalezená řešení neboli chování robotů jsou následně zkoumána nejprve v původních prostředích (těch, ve kterých probíhala evoluce) a poté v prostředích jiných. Tímto způsobem mohou být lépe zjištěny vlastnosti chování a stupeň robustnosti. Porovnáním jednotlivých chování mezi sebou lze získat představu o tom, které chování nejlépe splňuje zadanou úlohu. Toto porovnání v sobě zahrnuje informaci o tom, které prostředí je pro vývoj robotů nejlepší. Vlivy vlastností a charakteristiky prostředí na výsledné chování jsou diskutovány v závěru práce. Dále je zkoumána schopnost řídicího systému adaptovat se v průběhu evoluce na konkrétní konfiguraci robota, konkrétně experimentem se změnou frekvence vyčítání.

Robot je vybaven čtyřmi infračervenými senzory rozmístěnými symetricky kolem podélné osy robota, z nichž tři jsou nasměrovány na jednu stranu a jeden na stranu druhou. Senzory umístěné na jedné straně spolu svírají úhel 30° (obrázek 1.1).

Fitness funkce byla nejprve použita stejná jako v článku Floreana a Mondady [4], tedy

$$\Phi_1 = V \left(1 - \sqrt{\Delta v} \right) i, \quad (2.1)$$

kde V je průměrná rotační rychlost obou kol, Δv absolutní hodnota rozdílu rychlostí kol (kladná v jednom směru, záporná ve druhém) přičemž $0 < \Delta v < 1$ a i je hodnota signálu infračerveného senzoru, která určuje vzdálenost k nejbližší překážce.

Funkce Φ_1 se skládá ze tří složek. První je maximalizována absolutní velikostí rychlosti, druhá přímým pohybem a třetí udržováním co největší vzdálenosti od překážek. Vzhledem k tomu, že se kola robota mohou otáčet oběma směry, je tato funkce symetrická a má dvě ekvivalentní maxima, z nichž každé odpovídá jednomu směru jízdy. Pokud by byly také senzory rozestavěné symetricky, nebylo by možné určit, kterým směrem je výhodnější se pohybovat, a bylo by správné očekávat, že v průměru by polovina populace preferovala jeden směr a druhá polovina směr opačný. Vliv asymetrie senzorů na výsledné chování robotů je více probrán v závěru tohoto experimentu.

Vývoj řídicího systému začíná zakódováním genomu do umělé neuronové sítě, ke které jsou připojeny senzory a motory robota. Ten je umístěn do prostředí, v němž má probíhat evoluce, a je v něm ponechán jezdit. Délka jízdy je omezena maximálním počtem kroků, po jejímž uplynutí je neuronová síť opět odpojena. Hodnota fitness funkce se sčítá při každém kroku jízdy robota a na konci se vydělí maximálním počtem kroků. Tato úprava zajistí, že hodnota fitness bude v intervalu $\langle 0; 1 \rangle$. Také budou touto úpravou zvýhodněni roboti, kteří dokáží jezdit prostředím celou dobu a umí se vyhnout překážkám. Jedinci, kteří narazí do překážky a nejsou schopni od ní vymanévrovat, budou mít hodnotu fitness nižší.

2.2 Aplikace řídicí evoluci

Před samotnými experimenty bylo potřeba implementovat aplikaci, která bude řídit celou evoluci. Tato aplikace není hlavním předmětem zkoumání této práce, ale pouze nezbytným nástrojem, proto zde nebude podrobně popisována a budou zde uvedeny jen její nejdůležitější vlastnosti. Kompletní zdrojové kódy jsou k dispozici na přiloženém CD.

Aplikace zajišťuje běh celé evoluce a umožňuje její přerušování a opětovné pokračování. Jejím výstupem jsou soubory, ve kterých jsou uloženy veškeré informace o jednotlivých jedincích, tedy genom (nastavení vah a prahů umělé neuronové sítě) a hodnota fitness každého jedince. Z těchto dat pak lze zobrazit chování libovolného robota a jednotlivě je zkoumat. Odděleně jsou uložena data jedince, který během evoluce dosáhl nejvyšší hodnoty fitness. Navíc umožňuje ukládat výstupní hodnoty všech neuronů, což lze využít pro ještě detailnější zkoumání vyvinutého chování.

Pro práci s genetickými algoritmy využívá aplikace knihovnu GALib [14], která poskytuje základní genetické operátory, umožňuje implementaci vlastních a také zajišťuje sběr statistických dat z evolučních běhů.

2.3 Fitness funkce

Převzatá fitness funkce Φ_1 (2.1) nevedla k očekávaným výsledkům. Nejlepší jedinci se pohybují velmi rychle po malé kružnici v místě, které je vzdáleno nejdál ode všech překážek. Toto chování sice představuje v daném prostředí¹ maximum fitness funkce Φ_1 , avšak od původně očekávaného se diametrálně liší. Problém spočívá ve třetí složce fitness funkce, která minimalizuje aktivaci infračervených senzorů. Tím je sice zabráněno robotovi v narážení do překážek, ale zároveň dochází k penalizaci využívání těchto senzorů. Roboti (a zřejmě i živé organismy), kteří se pohybují ve volném prostoru, kde nedostávají ze senzorů žádnou informaci o překážkách, nemají žádnou zpětnou vazbu a nemohou se naučit koordinovat své pohyby. Složka zohledňující vzdálenost robota od překážek je z fitness funkce odebrána a jsou v ní ponechány pouze první dvě složky.

$$\Phi_2 = V \left(1 - \sqrt{\Delta v} \right) \quad (2.2)$$

Pro odstranění penalizace za nízkou vzdálenost od překážek hovoří i fakt, že v simulátoru Stage je prakticky nekonečné tření mezi robotem a překážkami. Pokud robot narazí do překážky, zastaví se a nemůže pokračovat stejným směrem. Jedinci, kteří nejsou schopni svými senzory překážky rozpoznat, zůstanou u překážek nehybně stát a tím pádem nezískají dostatečně vysokou hodnotu fitness a budou z populace eliminováni.

Po provedení několika pokusů s novou fitness funkcí (2.2) bylo zjištěno, že evoluce našla další neočekávanou možnost pohybu robotů. V použité verzi simulátoru Stage 2.1.1 může robot neomezeně rychle zrychlovat nebo zpomalovat, neboť není omezena velikost akcelerace, a jedinci se velmi rychle naučili tuto vlastnost využívat. Nejprve se pohybují

¹Iterativní úpravy fitness funkce byly prováděny na základě výsledků evolučních běhů z prostředí *Tunel*.

maximální rychlostí jedním směrem, dokud nenarazí do překážky, tam zastaví a jedou plnou rychlostí na opačnou stranu a takto oscilují, dokud jim nevyprší čas (maximální počet kroků). Toto chování sice dokáže výborně splnit fitness funkci, protože robot se naprostou většinu času pohybuje přímým směrem nejvyšší možnou rychlostí a tím získává téměř maximální dosažitelnou hodnotu fitness, ale stále nevyhovuje očekávanému chování.

Cílem následující úpravy je zabránění výše popsanému chování. Hodnota fitness funkce je po skončení jízdy robota upravena tak, aby byly minimalizovány změny směru jízdy robota (ve smyslu vpřed a vzad). Během jízdy se počítá, kolikrát robot změnil směr pohybu. Na konci jízdy se ověří, je-li tento počet vyšší než určitý práh² a pokud ano, je přidán do fitness funkce. Celková fitness funkce se tedy skládá z dvou částí a vypadá následovně

$$\Phi_{2,celk} = \frac{e^{-c-t}}{N} \cdot \sum_{i=1}^N V_i \left(1 - \sqrt{\Delta v_i}\right), \quad (2.3)$$

kde c je počet změn směru, t práh a N maximální počet kroků. Hodnoty fitness funkce z každého časového kroku jízdy robota, které jsou vypočítány podle (2.2), jsou sčítány nakonec vynásobeny exponenciálou a normalizovány maximálním počtem kroků robota. S touto výslednou fitness funkcí $\Phi_{2,celk}$ (2.3) se již podařilo vyvinout požadované chování robota.

2.4 Vývoj řídicích systémů

Vývoj probíhal v pěti různých prostředích v simulátoru Player/Stage [13] s modelem robota Morbot [5, 12]. V následující části jsou použita prostředí popsána podrobněji a také jsou analyzovány schopnosti a vlastnosti nalezených řídicích systémů.

Tunel

Blížším prozkoumáním jízdy nejlepšího robota (jeho trajektorie je zobrazena na obrázku 2.2) bylo zjištěno, že ve volném prostředí, kde je velmi nízká aktivace senzorů, jezdí robot pomaleji a jeho trajektorie je více ovlivněna šumem. Přiblíží-li se robot ke zdi, motory na proximální straně (té straně, která je blíže k překážce) se točí rychleji a díky tomu se robotovi daří vyhýbat se překážkám. Tato vlastnost má ovšem zajímavý důsledek. V případě, že je překážka jen na jedné straně robota, robot jednoduše zatačí pryč od překážky. Pokud je ovšem překážka po obou stranách (například v úzkém koridoru), točí se rychleji oba motory a robot tak jede větší rychlostí dopředu. Z toho plyne, že ve volném prostoru se robot pohybuje pomalu, kdežto v úzkých uličkách nebo v prostředí s hustě umístěnými překážkami se pohybuje rychleji.

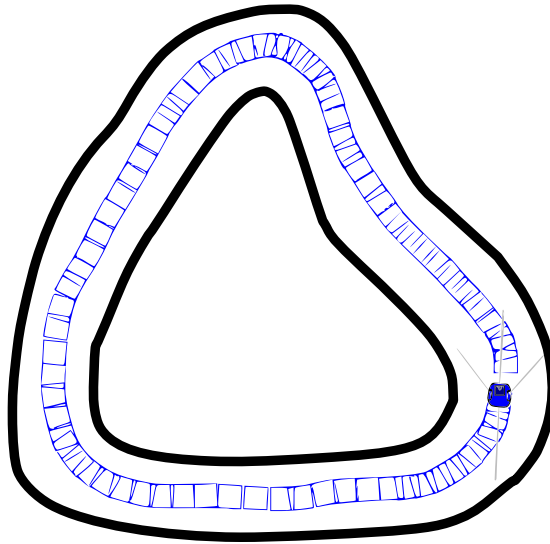
Vývoj hodnoty fitness³ celé populace během evoluce v tomto prostředí je na obrázku 2.3. Hodnota fitness rovnoměrně narůstá a během prvních 100 generací nekonverguje k žádné

²Tento práh je experimentálně nastaven na 5 změn směru jízdy.

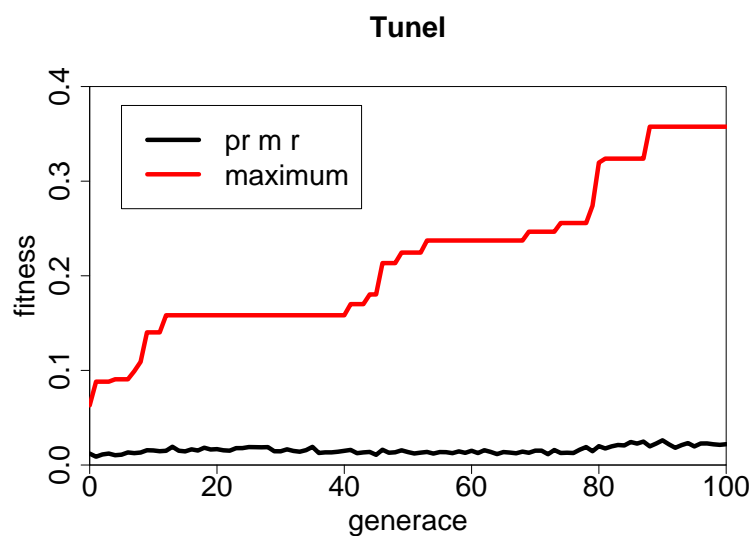
³Vykreslené hodnoty vývoje hodnoty fitness jsou ve všech grafech průměrem ze dvou evolucí. Pro získání grafů byl použit nástroj pro statistické zpracování dat R [9].

hodnotě. To by mohlo značit, že ještě nebyl nalezen optimální řídicí systém. Proto byla evoluce ponechána běžet dál, aby bylo možno říci, existuje-li lepší řešení.

Po dalších 100 generacích však nebylo dosaženo podstatného zlepšení a proto je jako nalezené řešení bráno to, které bylo nalezeno během prvních 100 generací. Hodnota fitness tohoto řešení je 0,36 a bylo dosaženo v 88. generaci.



Obrázek 2.2: Robot v prostředí *Tunel*.

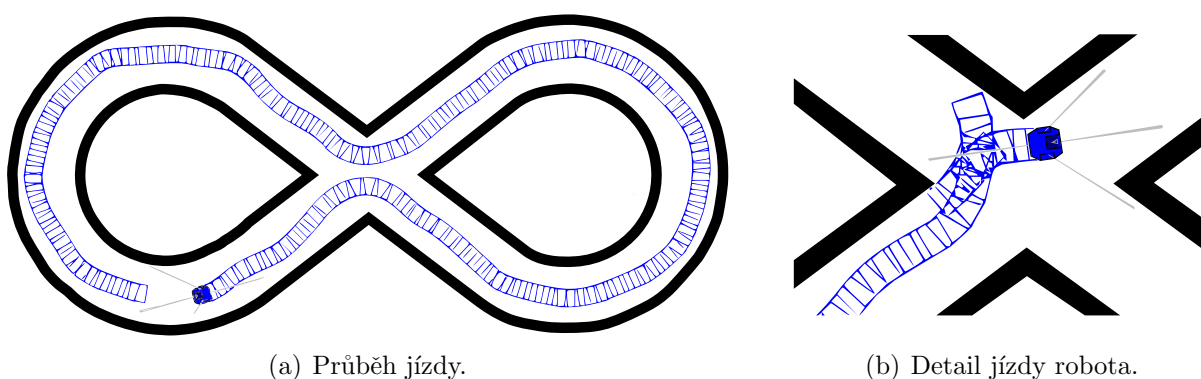


Obrázek 2.3: Vývoj hodnoty fitness v prostředí *Tunel*.

Osmička

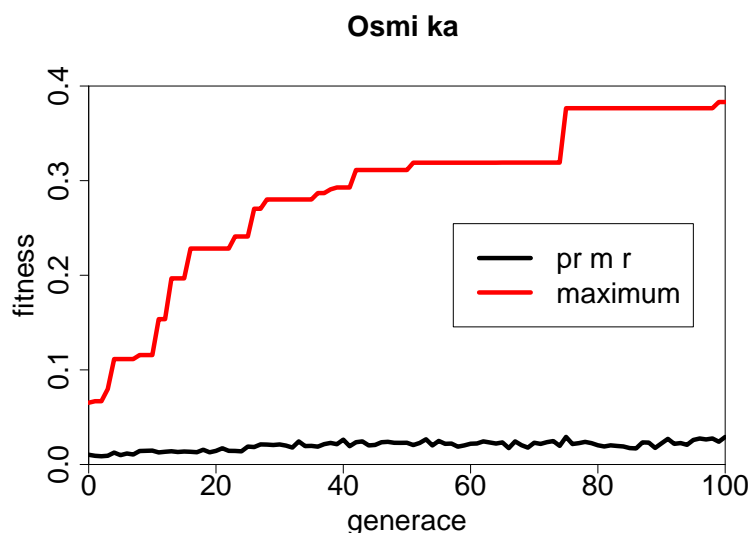
Toto prostředí je navrženo tak, aby robot, který jím bude projíždět, byl nucen zvládat pravotočivé i levotočivé zatáčky. Během evoluce v předchozím prostředí se totiž vyskytovali roboti, kteří uměli zatáčet pouze na jednu stranu. K úspěšné jízdě prostředím nebyla tato schopnost nutná a proto se u některých jedinců nevyvinula.

Chování jedinců vyvinutých v tomto prostředí je možné přirovnat k tzv. *wall following* neboli sledování zdi. Na obrázku 2.4(a) je vidět, že robot se snaží udržovat v průměru stejnou vzdálenost od levé zdi a na křižovatce preferuje zatáčení vlevo. Obrázek 2.4(b) ukazuje detail jízdy robota v ostřejší levotočivé zatáčce. Je vidět, že robot sice nedokáže vhodně zareagovat na prudší změnu směru, ale vyvinula se u něj schopnost detekovat kolizi s překážkou, odjet od této překážky a pokračovat v jízdě.



Obrázek 2.4: Robot v prostředí *Osmička*.

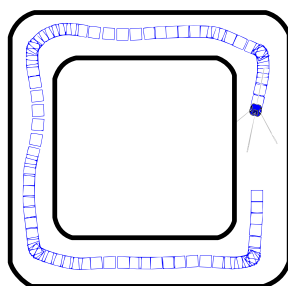
Vývoj hodnoty fitness funkce v tomto prostředí je zobrazen na obrázku 2.5. V 99. generaci je dosaženo optimálního chování s hodnotou fitness funkce 0,38.



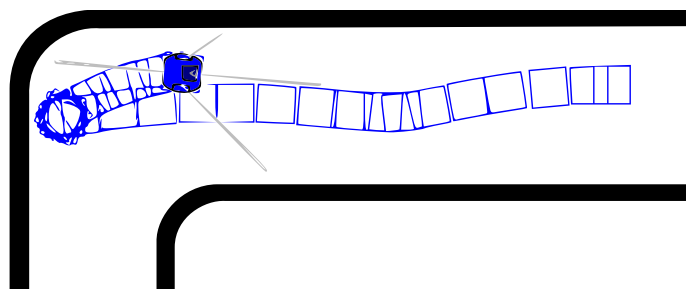
Obrázek 2.5: Vývoj hodnoty fitness v prostředí *Osmička*.

Okruh

Od tohoto prostředí se očekává, že se v něm robot naučí nejen dobře jezdit do zatáček, ale také udržovat přímý směr jízdy. Cesta je širší než v předchozích dvou prostředích, robot má více prostoru. Řídicí systém, který se zde vyvinul, zvládá navigovat robota prostředím aniž by došlo ke kolizi se zdmi prostředí (obrázek 2.6(a)), avšak robot je schopen zatáčet pouze jedním směrem (vpravo). Tato vlastnost se nejvýrazněji projeví, pokud je robot na počátku jízdy umístěn tak, že musí zatáčet vlevo. Taková situace je zobrazena na obrázku 2.6(b). Robot jede správně až k první zatáčce, kde se místo vlevo otočí vpravo a jede zpět.



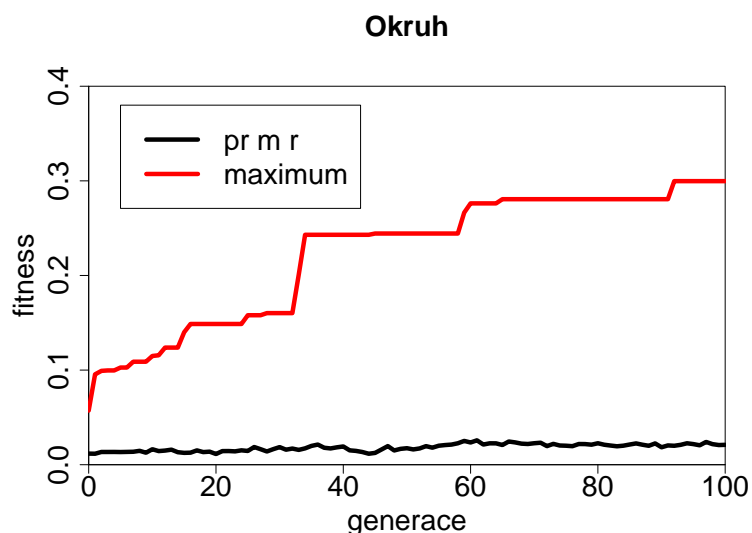
(a) Průběh jízdy.



(b) Detail jízdy robota proti směru hodinových ručiček.

Obrázek 2.6: Robot v prostředí *Okruh*.

Průběh vývoje hodnoty fitness je zobrazen na obrázku 2.7. Maximum je v tomto prostředí nižší než v předchozích, jeho hodnota je 0,30 a bylo dosaženo v 92. generaci.



Obrázek 2.7: Vývoj hodnoty fitness v prostředí *Okruh*.

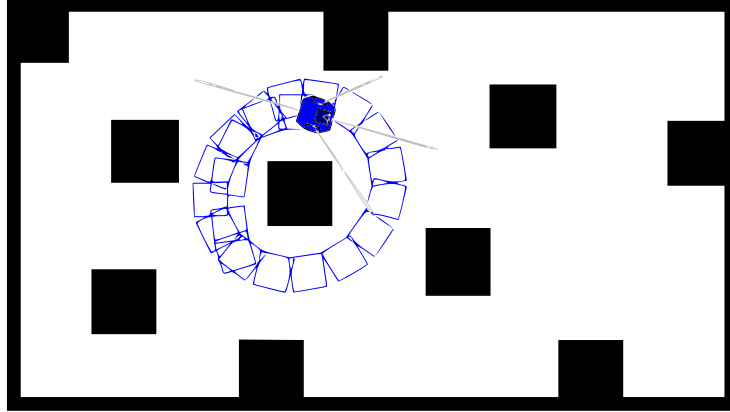
Kostky

Následující dvě prostředí se od předchozích velmi liší. Zatímco první tři jsou prakticky jen různě tvarované koridory, ve kterých je předem zřejmé, kudy musí robot jet⁴, v následujících prostředích převládá volný prostor. Robot tak má větší možnost volby své trajektorie.

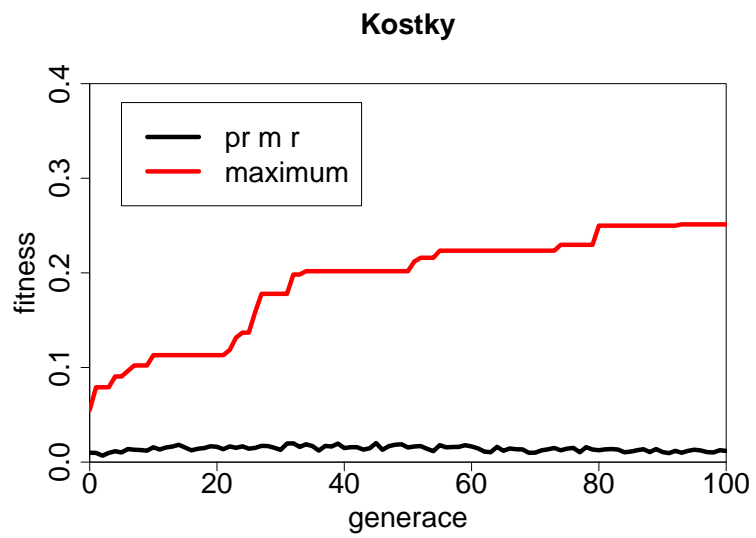
V prostředí *Kostky* se u nejlepšího jedince vyvinulo takové chování, kdy robot jezdí v kruzích. Senzory sice vnímá prostředí a podle překážek trochu upravuje svou trajektorii, ale ve srovnání s nejlepšími jedinci z minulých prostředí nevykazuje žádnou schopnost úspěšně se překážkám vyhnout. Na obrázku 2.8 je vidět, jak robot jezdí okolo překážky, ale pokud by byl ponechán jezdit déle, s velkou pravděpodobností by do nějaké narazil. Navíc jeho zdánlivě úspěšné chování je závislé na počátečních podmínkách. Pokud by nezačínal na místě, do kterého se může vrátit po přibližně kruhové trajektorii, projevila by se jeho neschopnost vyhýbat se překážkám nejvýrazněji.

Maximální hodnota fitness dosažená během prvních 100 generací je 2,6. Toto číslo je i přes nízkou kvalitu nalezeného chování poměrně vysoké v porovnání s hodnotami z předchozích prostředí. Průběh vývoje hodnoty fitness je na obrázku 2.9.

⁴Kromě prostředí *Osmička*, ve kterém se cesty kříží a robot tedy má více možností, kudy jet. Ovšem z hlediska fitness funkce je optimální přímá trajektorie.



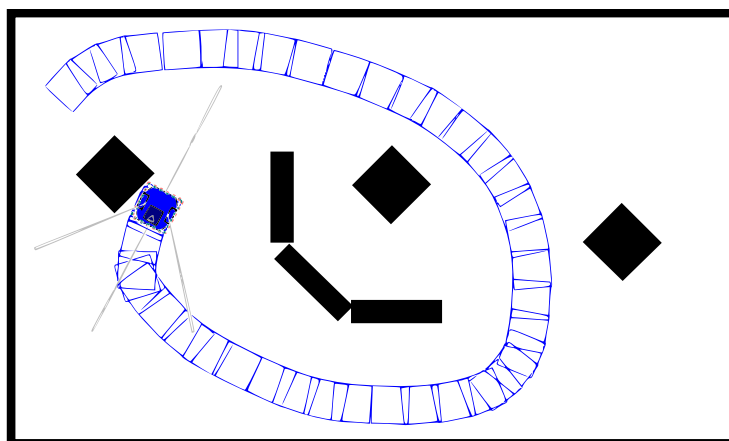
Obrázek 2.8: Robot v prostředí *Kostky*.



Obrázek 2.9: Vývoj hodnoty fitness v prostředí *Kostky*.

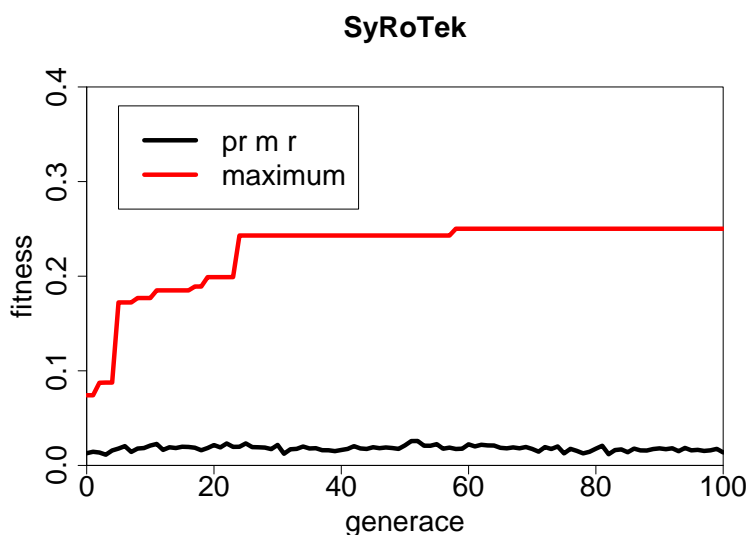
SyRoTek

V tomto prostředí se jako v jediném vyvinul preferovaný směr jízdy na tu stranu robota, kde je méně senzorů. Nejlepší jedinec má sklon udržovat mírně zakřivenou trajektorii a pokud předním senzorem zachytí překážku, zatačí ještě více, aby se překážce vyhnul. Na obrázku 2.10 je zobrazena trajektorie jízdy tohoto robota. Je vidět, že úspěšné vyhnutí se překážce záleží na její poloze vzhledem k robotovi a její velikosti. Zatímco zdi se robot dokáže vyhnout vcelku dobře, u menších překážek již nemusí být tak úspěšný. Kvůli tomu, že se robot pohybuje na tu stranu, kde získává méně informací o překážkách, je jeho schopnost správně se jim vyhnout nízká.



Obrázek 2.10: Robot v prostředí *SyRoTek*.

Maximální dosažená hodnota fitness v prostředí *SyRoTek* je 2,6. Průběh hodnoty fitness v závislosti na generaci během evoluce je zobrazen na obrázku 2.11.



Obrázek 2.11: Vývoj hodnoty fitness v prostředí *SyRoTek*.

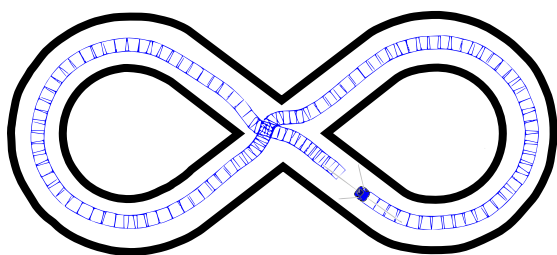
2.5 Porovnání nalezených chování

K objektivnímu posouzení kvality nejlepších nalezených řešení byl proveden následující experiment. Do všech pěti výše zmíněných prostředí byli postupně umisťováni roboti a byla měřena jejich hodnota fitness funkce. V každém prostředí bylo zvoleno pět počátečních pozic, ze kterých roboti startovali, a doba jejich života byla prodloužena na 200 kroků, aby se změřená hodnota fitness co nejméně přiblížila hodnotě skutečné. Startovní pozice byly zvoleny tak, aby vyžadovaly od robotů různé schopnosti — jízdu koridorem vlevo a vpravo, vyhýbání se malým a velkým překážkám, úspěšnou navigaci z těsné blízkosti překážek a podobně. Z naměřených hodnot byl poté vypočítán aritmetický průměr pro každého robota ve všech prostředích. Výsledky jsou rozepsány v tabulce 2.2. Pro každého robota byl navíc spočítán aritmetický průměr jeho fitness ve všech prostředích, čímž bylo získáno objektivní srovnání robotů mezi sebou.

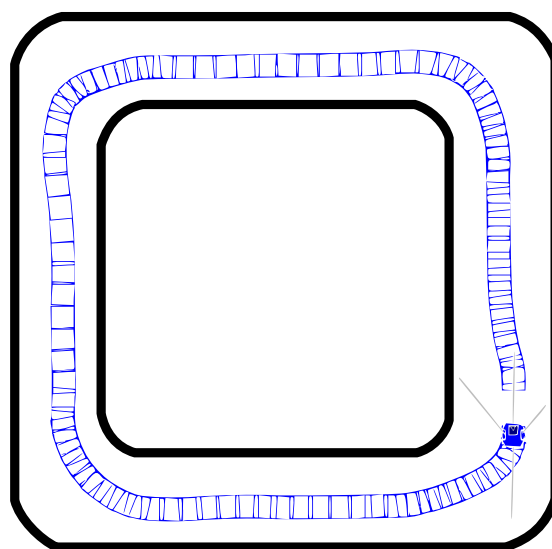
prostředí	roboti vyvinutí v prostředích				
	<i>Tunel</i>	<i>Osmička</i>	<i>Okruh</i>	<i>Kostky</i>	<i>SyRoTek</i>
<i>Tunel</i>	0,238	0,209	0,103	0,022	0,000
<i>Osmička</i>	0,258	0,275	0,243	0,044	0,000
<i>Okruh</i>	0,276	0,270	0,138	0,033	0,000
<i>Kostky</i>	0,213	0,228	0,077	0,107	0,087
<i>SyRoTek</i>	0,172	0,157	0,172	0,122	0,150
průměr	0,231	0,228	0,147	0,066	0,047

Tabulka 2.2: Průměry hodnot fitness funkce nejlepších robotů v jednotlivých prostředích.

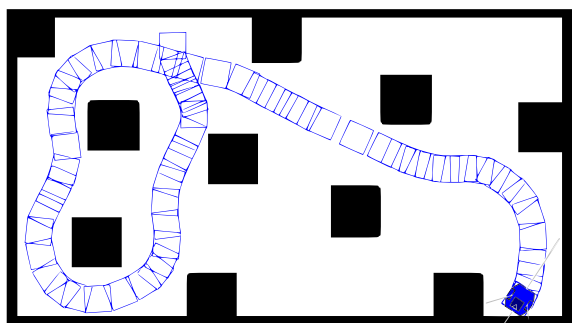
Jako nejlepší řídicí systém lze označit ten, jenž se vyvinul v prostředí *Tunel*, jelikož má v porovnání s ostatními nejvyšší průměrnou hodnotu fitness. Srovnatelnou kvalitu má také řídicí systém z prostředí *Osmička*, který je v průměru pouze nepatrně horší.



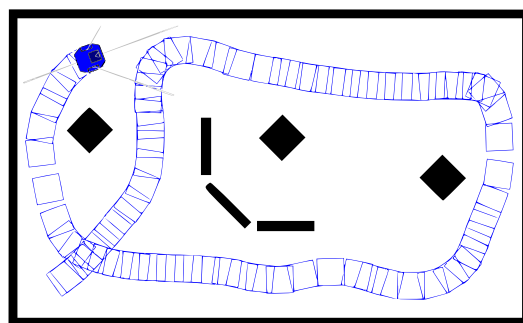
(a) V prostředí *Osmička*.



(b) V prostředí *Okruh*.



(c) V prostředí *Kostky*.



(d) V prostředí *SyRoTek*.

Obrázek 2.12: Trajektorie robota vyvinutého v prostředí *Tunel* v ostatních prostředích.

2.6 Perioda vzorkování

Bližší pozornost je věnována vlivu vzorkovací frekvence na výsledné chování robota. Jak již bylo uvedeno dříve, je předpokládáno, že robot se během evoluce přizpůsobí prostředí a své vlastní konfiguraci a vlastnostem. Charakteristika robota, která by měla ovlivnit jeho výsledné chování, není dána jen jeho fyzickou stavbou, tedy např. natočením senzorů, ale také vlastnostmi interakce s prostředím, tedy způsobem získávání informací ze senzorů. Ve všech předchozích experimentech, jsou data z infračervených senzorů vyčítána s periodou 300 ms. V závislosti na velikosti a tvaru překážek se tak může snadno stát, že robot překážku jednoduše „neuvídí“, pokud ji senzorem nezachytí v okamžiku vzorkování. Proto je ve většině robotických aplikací používáno rychlejší vyčítání senzorů, aby k podobným situacím nedocházelo a robot včas reagoval na změny prostředí.

Cílem této části práce je ověřit předpoklad, že robot se během evoluce přizpůsobí periodě vzorkování tak, aby byl schopen správně reagovat na překážky. Tedy prozkoumat, jak se při zvýšení vzorkovací frekvence změní průměrná rychlost nově vyvinutých řídicích systémů. Jelikož fitness funkce maximalizuje dopřednou rychlost robota, lze očekávat, že při zvýšení vzorkovací frekvence, která dává robotovi lepší informaci o okolí, se zvětší průměrná rychlost.

Jako hodnoty rychlosti robota s delší periodou vzorkování (300 ms) byly použity hodnoty rychlosti z jízdy nejlepšího robota v prostředí *Tunel*. Kratší perioda vzorkování byla zvolena 100 ms a evoluce probíhala v témže prostředí. Během jízdy robota byla měřena jeho rychlost v okamžicích vzorkování a z těchto hodnot vypočtena rychlost průměrná. Porovnání průměrných rychlostí robotů s oběma periodami vzorkování uvádí tabulka 2.3. Ta ukazuje, že trojnásobné zrychlení vzorkovací frekvence zvýšilo průměrnou rychlost robotů 1,8krát.

perioda vzorkování [ms]	průměrná rychlost [-]
300	0,35
100	0,63

Tabulka 2.3: Porovnání průměrných rychlostí robotů s různými periodami vzorkování.

Z provedeného experimentu vyplývá, že robot se skutečně dokáže přizpůsobit periodě vzorkování svých senzorů a vhodně jí využít k maximalizaci fitness funkce. Lze předpokládat, že se podobně adaptuje i na další své vlastnosti a vhodně je využívá.

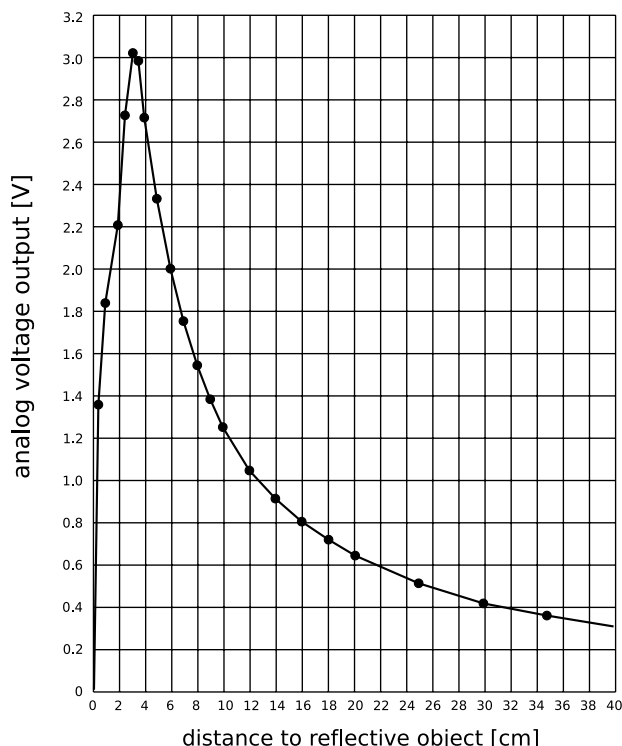
2.7 Experiment s reálným robotem

Cílem tohoto experimentu je zkoumání, je-li možné úspěšně přenést řídicí systém (umělou neuronovou síť) vyvinutý v simulátoru na reálného robota. To by umožnilo provést časově a hardwarově náročnou evoluci v simulovaném prostředí s modelem robota a na reálném robotovi použít až výsledný řídicí systém. V případě úspěšného přenesení řídicího systému lze usuzovat, že se nejen dobře adaptoval na konkrétní model robota a prostředí, ale také je dostatečně robustní a dokáže zpracovat i data z reálných senzorů a správně na ně reagovat povely pro motory.

Robot

Pro reálné experimenty byl vybrán robot Morbot [5, 12] Tato volba vychází z jeho poměrně malé velikosti, diferenciálního podvozku a vhodného senzorkého vybavení (4 infračervené dálkoměry SHARP).

Senzory SHARP, kterými je vybaven, mají velmi nelineární charakteristiku měření vzdálenosti od překážek. Její tvar je na obrázku 2.13. V simulátoru Stage je charakteristika lineární a jelikož ovladače robota Morbot poskytují naměřené hodnoty jako napětí na výstupu A/D převodníku, bylo zapotřebí převést tyto hodnoty až v klientské aplikaci, která robota řídí.

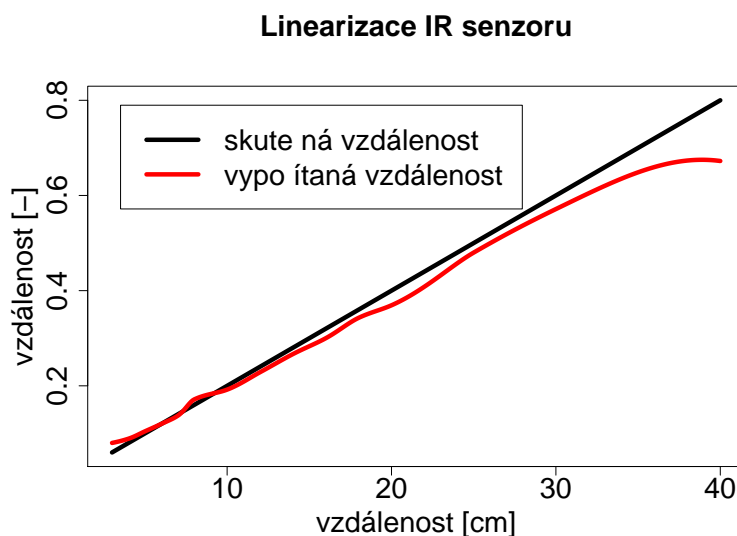


Obrázek 2.13: Převodní charakteristika IR dálkoměru GP2D120XJ00F, převzato z [10].

Pracovní oblast IR dálkoměru, za kterou jsou považovány hodnoty vzdálenosti od 3 cm výše, byla přibližně aproximována hyperbolou

$$f = \frac{12}{x}$$

a vypočítané hodnoty vzdálenosti byly oříznuty do intervalu $\langle 0; 0,8 \rangle$. Výsledná převodní charakteristika po linearizaci vstupních signálů touto funkcí je spolu s ideální zobrazena v grafu na obrázku 2.14.



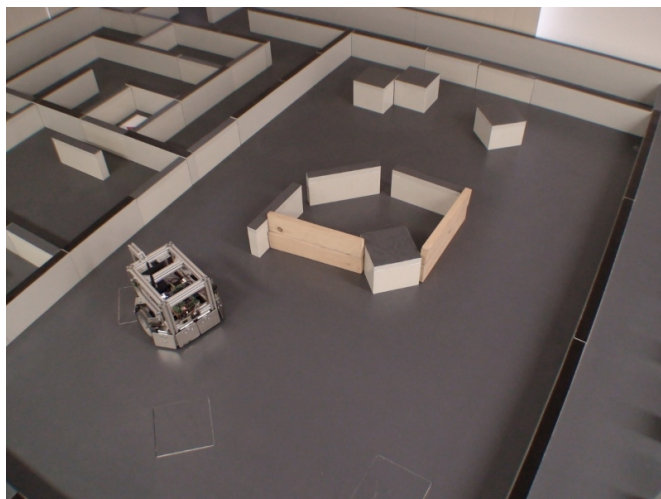
Obrázek 2.14: Linearizovaná převodní charakteristika IR dálkoměru GP2D120XJ00F (podle převodní charakteristiky (obrázek 2.13), zobrazena pouze pracovní oblast).

Infračervené dálkoměry byly na robotovi umístěny podobně jak je zobrazeno na obrázku 1.1. Pouze zadní senzor byl kvůli konstrukci robota připevněn na hliníkovém nosníku asi 7 cm od rámu. Toto umístění mohlo přinést další problém, způsobený charakteristikou senzorů. Ta totiž není funkcí prostou a pro dvě různé vzdálenosti vrací jedinou hodnotu napětí. Pokud se robot pohybuje dál od překážek, není tento problém patrný. Když se ovšem vzdálenost snížila pod určitou hranici (přibližně 3 cm, ale jednotlivé senzory se od sebe svými charakteristikami mírně liší), zdá se robotovi, že je překážka naopak výrazně dál, což může vést ke kolizi s touto překážkou. Vyhnout se tomuto problému lze umístěním senzorů dovnitř robota tak, aby byly překážky snímány pouze v pracovní oblasti senzoru. V provedeném experimentu jezdil robot v dostatečné vzdálenosti ode všech překážek, takže se změnou umístění senzorů nebylo potřeba zabývat.

Prostředí

Experiment byl prováděn v robotické aréně umístěné v místnosti KN:E-132. Tato aréna je součástí projektu SyRoTek (Systém pro robotickou televýuku), který se zabývá výzkumem postupů metod k realizaci unikátní modulární multi-robotické platformy pro

distanční vzdělávání [11]. Některé překážky v této aréně jsou pohyblivé — lze je zasunovat do podlahy — čímž je možné měnit charakter prostředí, ve kterém se robot pohybuje. Této možnosti bylo využito i v experimentu, jak je vidět na obrázku 2.15, kde jsou v levé dolní části vidět tři překážky zasunuté a v horní části několik vysunutých. Ty spolu s několika dřevěnými stěnami tvoří jednu kompaktní překážku zhruba ve středu arény.



Obrázek 2.15: Reálný robot v testovacím prostředí.

Experiment

Jako řídicí systém byla použita neuronová síť vyvinutá v prostředí *Tunel*, protože v porovnání s ostatními neuronovými sítěmi vykazovala nejlepší chování (jak vyplývá z tabulky 2.2). Na její vstupy byly přivedeny linearizované signály z dálkoměrných senzorů reálného robota. Rychlost byla z bezpečnostních důvodů snížena na polovinu, aby při případných kolizích nedošlo k poškození robota nebo prostředí.

Robot byl postupně umístěn na šest různých pozic v aréně a ponechán volně se pohybovat. Z šesti provedených pokusů byly čtyři úspěšné. Jako úspěšný pokus byl považován takový, při kterém byl robot schopen projet arénou alespoň jednou dokola bez kolize s překážkou. Neúspěšné pokusy byly způsobené tím, že řídicí systém byl vyvíjen v prostředí, ve kterém se nevyskytovaly konkávní rohy a řídicí systém nebyl naučen správně na ně reagovat. Pokud byl robot umístěn do podobného prostředí v simulátoru, docházelo k podobným situacím. Kvalita řízení v simulátoru a v reálném prostředí je tedy srovnatelná, z čehož lze usuzovat, že vyvinutý řídicí systém je robustní a jeho přenesení na reálného robota úspěšné.

Závěr

Experimentem bylo ověřeno, že neuronová síť řídicí robota Morbot zvládá téměř bezkolizní jízdu v popsáném prostředí. Tento výsledek je zajímavý především z toho důvodu,

že byla vyvíjena v odlišném prostředí a na rozdílném modelu robota. Zatímco v simulátoru byly až na uměle přidávaný šum do senzorů všechny vlastnosti ideální, v reálném prostředí se vyskytovaly nepřesnosti na všech úrovních (od drobných nerovností podložky, odlišných vlastností pohonu robota až po jiné charakteristiky a natočení infračervených senzorů). Se všemi těmito odlišnostmi se řídicí systém dokázal celkem úspěšně vypořádat. Hlavním problémem tak byla pouze neschopnost řídicího systému vyhnout se konkávním rohům, která byla způsobena charakteristikou prostředí, ve které byl tento řídicí systém vyvíjen. Z dosaženého výsledku lze usuzovat, že pro vývoj řídicího systému je možné využít simulátor a na reálném robotovi použít až hotové vyvinuté řešení.

Je pravděpodobné, že jednu z klíčových rolí v úspěšném přenesení řídicího systému na reálného robota hraje šum, jenž byl během evoluce uměle superponován k užitečnému signálu z dálkoměrných senzorů. Vyvinutý řídicí systém se proto neadaptoval konkrétně na vlastnosti těchto senzorů a byl schopen správně reagovat i na reálný signál. Také linearizační funkce tak stačila pouze jednoduchá a nebylo zapotřebí vyjadřovat složitou charakteristiku infračervenému senzoru SHARP přesněji.

2.8 Shrnutí experimentů

Roboti se během evoluce ve většině případů naučili jezdit prostředím a vyhýbat se překážkám. Bylo zjištěno, že roboti se velmi konkrétně adaptují na prostředí, ve kterém jsou vyvíjeni. Schopnosti, které nejsou zapotřebí k úspěšné navigaci, robot nezíská, protože v daném prostředí jednoduše chybí prvky, jenž by takové schopnosti vyžadovaly. Pokud je cílem získat univerzálnější chování zvládající jízdu v různorodých prostředích, je zapotřebí nechat robota vyvíjet v takovém prostředí, které obsahuje co nejvíce různých prvků, na něž se bude moci adaptovat. V úloze obstacle avoidance to může být například různá šířka cest nebo různé tvary, velikosti a rozmístění jednotlivých překážek.

Ve výsledných chováních lze pozorovat několik emergentních vlastností, které se u robotů vyvinuly v průběhu evoluce. Zajímavý poznatek přineslo asymetrické uspořádání senzorů. I když byl robot z hlediska pohonu symetrický, v průběhu evoluce se u většiny nejlepších robotů vyvinul preferovaný směr jízdy tím směrem, kde měli roboti více senzorů. Takový výsledek lze logicky předpokládat, jelikož na straně, ze které dostávali lepší a přesnější sensorickou informaci, dokázali roboti lépe detekovat překážky a vyhnout se jim. Ti jedinci, kteří preferovali opačný směr, v průběhu evoluce zanikali. Nedokázali se vyhnout překážkám, naráželi a jejich hodnota fitness nebyla dostatečně vysoká, aby uspěli v selekci do dalších generací. Výjimku tvoří pouze řídicí systém vyvinutý v prostředí *SyRoTek*, který preferuje jízdu směrem opačným. Příčinu tohoto chování se ovšem nepodařilo objasnit.

Podobně jako došlo k adaptaci na uspořádání senzorů, bylo ověřeno také přizpůsobení se na rychlost vyčítání dat. Robot se v prostředí nepohyboval maximální možnou rychlostí, ale pouze tak rychle, aby byl schopen včas reagovat na překážky. Toto zjištění je dalším důkazem toho, že evoluční hledání řešení nalezne takový řídicí systém, který využívá všechny možnosti, které umožňují co nejlepší splnění zadaného úkolu.

Naproti tomu výsledky reálného experimentu ukazují, že nalezený řídicí systém je nejen konkrétně adaptován na danou konfiguraci prostředí a robota, ale dokáže správně pracovat také ve výrazně jiném prostředí s jiným modelem robota. Jak již bylo zmíněno, důležitou roli v úspěšném přenesení řídicího systému ze simulátoru na reálného robota hraje šum, uměle přidaný k infračerveným sensorům. Dále k tomu přispělo i použití umělých neuro-nových sítí. Ty již ze své podstaty mají schopnost generalizace a tím ještě více napomáhají správné interpretaci reálných dat.

Další pozorování vyplynulo ze srovnání jízdy robota v prostředí, kde byl vyvíjen jeho řídicí systém, s jízdou v prostředí jiném. Toto srovnání přináší za prvé informaci o kvalitě nalezeného řídicího systému, za druhé také informaci o prostředí samotném. Prostředí, ve kterých se vyvinou optimální řídicí systémy, lze pokládat za dostatečně různorodé nebo jinak motivující vývoj úspěšných chování.

Pro představu je vhodné zdůraznit, že vývoj jednoho řídicího systému v simulátoru trval na počítači Mobile AMD SempronTM3100+, 2 GB DDR, Radeon XPRESS 200M přibližně 43 hodin (pro délku kroku 300 ms, délku života jedince 50 kroků, 100 jedinců v generaci a 100 generací). Rychlost hledání řešení je omezena použitým simulátorem, protože kvalita simulace je s rostoucí rychlostí nižší. Při hledání řešení složitější úlohy, která by pravděpodobně vyžadovala delší dobu vyhodnocování jedince a větší počet jedinců, je s délkou evoluce nutno počítat. S tím je nutné počítat, pokud je záměrem hledat řešení složitější úlohy, která by pravděpodobně vyžadovala delší dobu vyhodnocování jedince a větší počet jedinců. Pokud je snahou vývoj řídicího systému přímo na reálném robotovi, je vhodné brát délku evoluce v potaz při jeho návrhu a výrobě.

Kapitola 3

Závěr

V bakalářské práci byl genetickými algoritmy úspěšně vyvinut řídicí systém pro model robota Morbot v simulátoru Player/Stage. Pro jeho vývoj byla implementována aplikace, jež celou evoluci kontroluje a umožňuje zaznamenávání všech podstatných dat ze svého průběhu. Řídicí systémy byly vyvíjeny současně v pěti různých prostředích a byl z nich vybrán jeden nejlepší. Ten byl následně úspěšně přenesen na reálného robota Morbot v aréně SyRoTek. Řídicí systém dokázal úspěšně ovládat reálného robota, čímž se ukázaly jeho schopnosti správně reagovat na různorodá vstupní data a na odlišné modely prostředí a robota. Dále byl učiněn experiment, kterým bylo ověřeno, že řídicí systém vyvinutý genetickými algoritmy je adaptován na konkrétní konfiguraci a dokáže ji optimálně využít ke splnění daného cíle.

Během provedených experimentů bylo zaznamenáno několik zajímavých výsledků. Nalezené chování robotů je silně závislé na fitness funkci, vlastnostech prostředí a možnostech a vlastnostech robota. Nemusí ovšem přesně odpovídat představě experimentátora při vytváření fitness funkce a při návrhu je proto výhodné postupovat iterativně a tvar fitness funkce upravovat podle výsledků průběžných experimentů. Řídicí systém je sice adaptován na konkrétní konfiguraci, ale dokáže úspěšně řídit i odlišný model robota v reálném prostředí. Z pozorovaných výsledků lze usuzovat, že pro vývoj řídicího systému autonomních mobilních robotů je výhodné využít genetických algoritmů a umělé neuronové sítě, jelikož oba tyto principy přinášejí do výsledného řídicího systému důležité prvky. Genetické algoritmy umožňují efektivní hledání optimálního řídicího systému a neuronová síť přináší robustní řízení samotného robota.

Použití genetických algoritmů je vhodné u problémů, jejichž analytické řešení je obtížné. Řešení úlohy obstacle avoidance je typickou úlohou pro genetické algoritmy, jelikož se jedná o úlohu optimalizace trajektorie neznámým prostředím.

Přirozeným pokračováním práce by mohl být vývoj dalších druhů chování (explorace prostředí, pursuit & evasion apod.) s rozšířením pro evoluci více robotů najednou. Vhodným vylepšením by bylo využití ontogenetické adaptace, při které by jedinci měli možnost se během svého života učit a měnit se.

Literatura

- [1] Darwin, C. R.: *The origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. London: John Murray, 6 vydání, 1872.
URL <http://darwin-online.org.uk>
- [2] Floreano, D.; Mondada, F.: Active Perception, Navigation, Homing, and Grasping: An autonomous Perspective. V *From Perception to Action Conference (PERAC'1994)*, editace P. Gaussier; J.-D. Nicoud, 1994, cA: IEE Computer Society Press, Ph. Gaussier and J.-D. Nicoud (eds.).
- [3] Floreano, D.; Mondada, F.: Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural Network Driven Robot. V *3rd International Conference on Simulation of Adaptive Behavior (SAB'1994)*, editace D. Cliff; P. Husbands; J.-A. Meyer; S. Wilson, MA: MIT Press, 1994, d. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson (eds.).
URL <http://www.isab.org.uk/confs/sab94.php>
- [4] Floreano, D.; Mondada, F.: Evolution of Homing Navigation in a Real Mobile Robot. *IEEE Transactions on Systems, Man and Cybernetics Part B : Cybernetics*, ročník 26, č. 3, 1996: s. 396–407, doi:NA.
- [5] Grimmer, V.: *Platforma pro výuku mobilní robotiky*. Bakalářská práce, ČVUT FEL, Praha, Česká republika, 2008.
- [6] Hynek, J.: *Genetické algoritmy a genetické programování*. Praha, Česká republika: Grada Publishing, a.s., 2008.
- [7] Mařík, V.; Štěpánková, O.; Lažanský, J.: *Umělá inteligence (3)*. Praha, Česká republika: Academia, 2001.
- [8] Nolfi, S.; Floreano, D.: *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines*. Cambridge, MA: MIT Press, 2001, 2001 (2nd print), 2000 (1st print).
URL <http://mitpress.mit.edu/book-home.tcl?isbn=0262140705>

- [9] R Development Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0.
URL <http://www.R-project.org>
- [10] Sharp Microelectronics, GP2D120XJ00F Sales and Technical Information. 2009.
URL <http://www.sharpsma.com/Page.aspx/americas/en/part/GP2D120XJ00F>
- [11] SyRoTek, Systém pro Robotickou Tele-výuku.
URL <http://lynx1.felk.cvut.cz/syrotek/>
- [12] Szücsová, H.: *Podpůrný software pro výuku mobilní robotiky*. Bakalářská práce, ČVUT FEL, Katedra kybernetiky, Praha, Česká republika, 2008.
- [13] The Player project.
URL <http://playerstage.sourceforge.net>
- [14] Wall, M.: GALib, A C++ Library of Genetic Algorithm Components.
URL <http://lancet.mit.edu/ga/>

Příloha A

Obsah CD

Přiložené CD obsahuje zdrojové kódy aplikace pro řízení evoluce a dalších pomocných programů a skriptů pro automatizaci experimentů. Dále obsahuje text bakalářské práce ve formátu PDF a zdrojové kódy celé práce pro systém L^AT_EX. V následující tabulce je stručně popsána struktura CD.

Adresář	Popis
<code>src</code>	zdrojové kódy aplikací
<code>doc</code>	zdrojové kódy bakalářské práce
<code>thesis.pdf</code>	text bakalářské práce

Tabulka A.1: Adresářová struktura na CD.