

SIMULATION BASED NEURAL MOTION PLANNER LEARNING

Radek Hrbáček

Bachelor Degree Programme (3), FIT BUT

E-mail: xhrbac01@stud.fit.vutbr.cz

Supervised by: Jiří Krejsa

E-mail: krejsa@fme.vutbr.cz

Abstract: This paper deals with design and implementation of mobile robot simulator and neural network based motion planner. Data gained from the simulation are used to train neural planner, where the simulated motion can be driven either by any existing motion planner or manually using a joystick. This allows the planner to learn many types of planning strategies ranging from simple ones to complex human-inspired behaviors.

Keywords: mobile robot, simulation, motion planner, neural network

1 ÚVOD

Reklamní robot Advee je produktem firmy Bender Robotics s.r.o. Jedná se o nové reklamní médium, viz obrázek 1. Robot založený na Ackermanově podvozku je autonomní, disponuje dotykovým panelem, tiskárnou na letáky, reproduktory, kamerou a senzory pro pohyb ve vnitřním prostředí, konkrétně ultrazvukovými senzory pro detekci překážek a přijímačem infračervených majáků sloužícím k lokalizaci. Účelem robota je upoutat pozornost kolemjdoucích a předat jim reklamní sdělení. Se zákazníkem komunikuje prostřednictvím hlasu a dotykové obrazovky.

Tento článek se zabývá simulací prostředí, ve kterém se robot pohybuje, a jeho senzorů. Simulace je dále využita k učení plánovače pohybu založeného na neuronové síti.

2 ARCHITEKTURA SOFTWAREVÉHO VYBAVENÍ ROBOTY

Při návrhu architektury software robota byl kladen důraz na modularitu a spolehlivost řešení. Celková koncepce vychází z hardwarového uspořádání, Advee je vybaven dvěma počítačovými deskami: vestavěnou deskou TS-7800 s OS GNU/Linux pro sběr dat ze senzorů a pohyb robota a deskou Zotac IONITX-F-E s OS Windows 7 pro interakci s uživatelem. Tyto komponenty jsou propojeny sítí standardu Gigabit Ethernet. Uspořádání je rozšiřitelné, pokud by vzrostla výpočetní náročnost.

Při volbě komunikačního mechanismu bylo nutné vzít v úvahu několik kritérií: propustnost a latenci komunikace, podporu více platforem (minimálně GNU/Linux a Windows, jazyky C/C++, C#.NET) a možnost definovat jednoznačná rozhraní mezi moduly. Po zvážení více možností byla vybrána knihovna LCM (*Lightweight Communications and Marshalling* [1]), splňující všechny naše požadavky kromě podpory C#.NET – tu bylo nutné přidat vlastními silami. Komunikace pomocí LCM zahrnuje tři hlavní části: definici zpráv, serializaci/deserializaci zpráv a vlastní komunikaci. Ta využívá UDP *multicast* a umožňuje tak vytvoření sběrnice architektury.



Obrázek 1: Advee

Software robota je rozdělen na tři logické celky od nejnižší úrovně (komunikace s hardware) po nejvyšší (komunikace s uživatelem). *Lowlevel* se stará o sběr dat ze senzorů (ultrazvukové senzory, odometrie, infračervené majáky) a ovládání pohybu robota (dopředný pohyb a zatáčení). Vrstva *middlelevel* implementuje estimátor polohy (poloha je určována pomocí Kalmanova filtru [2], na vstupu jsou data z odometrie a infračervených majáků) a plánovač pohybu (viz schéma na obrázku 2).

3 SIMULÁTOR PROSTŘEDÍ A SENZORŮ

Vzhledem k rozměrům robota bylo užitečné vytvořit simulátor prostředí, ve kterém se robot běžně pohybuje, společně se senzory, které využívá pro orientaci v prostoru. Díky tomu je možné testovat jednotlivé softwarové moduly bez nutnosti fyzického pohybu robota. Simulátor nahrazuje celou vrstvu *lowlevel* a díky dobře navrženému rozhraní mezi jednotlivými moduly dokáže spolupracovat se zbytkem softwarového vybavení bez zásadních změn v kódu.

Při implementaci simulátoru byla využita knihovna SIMLIB/C++ [3], která poskytuje vše potřebné pro diskrétní, spojitou i kombinovanou simulaci. Aby bylo možné spouštět simulaci v reálném čase, bylo nutné upravit algoritmus *next-event* přidáním čekání před naplánováním další události. Naopak pro simulaci zrychlenou bylo potřeba implementovat synchronizaci mezi simulátorem a estimátorem polohy a plánovačem pohybu. Tato synchronizace byla vyřešena opět pomocí LCM zpráv a pro pozastavení procesů a opětovné uvolnění jejich činnosti byl navržen mechanismus využívající *mutex*.

Simulátor je složen ze dvou základních částí: modelu prostředí, ve kterém se robot pohybuje, a modelu samotného robota. Jednotlivé procesy odpovídají skutečným procesům uvnitř robota, jedná se zejména o ultrazvukové senzory, infračervené majáky a odometrii. Tyto procesy vytvářejí stejné LCM zprávy jako skutečné senzory, vrstva *middlelevel* tedy nepozná, že se jedná o simulaci.

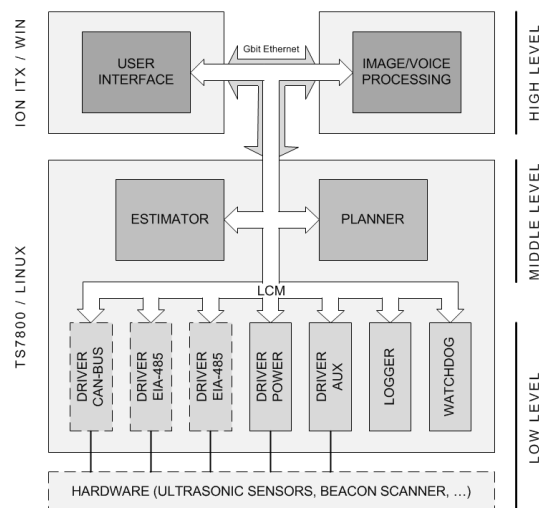
Možnosti využití simulátoru jsou značné – od pouhého sledování chování vyšších vrstev software v reálném či zrychleném čase, sběru statistických dat o procesech, až po získání dat k trénování plánovače pohybu založeného na některé z metod strojového učení, například neuronové síti.

4 PLÁNOVAČ POHYBU ZALOŽENÝ NA NEURONOVÉ SÍTI

Neuronová síť (NN) je biologií inspirovaný výpočetní model používaný především v umělé inteligenci. Je složena z neuronů, které jsou vzájemně propojeny a paralelně zpracovávají signály přivedené na jejich vstupy. Tyto signály transformují přenosovou funkcí na jediný výstup neuronu. Existuje značné množství topologií sítě, v této práci byla použita klasická dopředná třívrstvá síť.

Neuronová síť pracuje jako obecný aproximátor funkce ve dvou fázích. Při fázi tréninkové využívá relací mezi známými vstupy a výstupy k nastavení vnitřních parametrů (vah na synoptických spojeních) tak, aby byla minimalizována chyba mezi výstupem sítě a požadovaným výstupem. Z řady možných algoritmů k nastavení parametrů byl v práci vybrán Levenberg-Marquardtův.

Běh NN po ukončení trénovací fáze je velmi rychlý. Toho lze využít k aproximaci plánovače, který je výpočetně neúnosně náročný, jako tréninková data se použijí data získaná během simulace s takovým

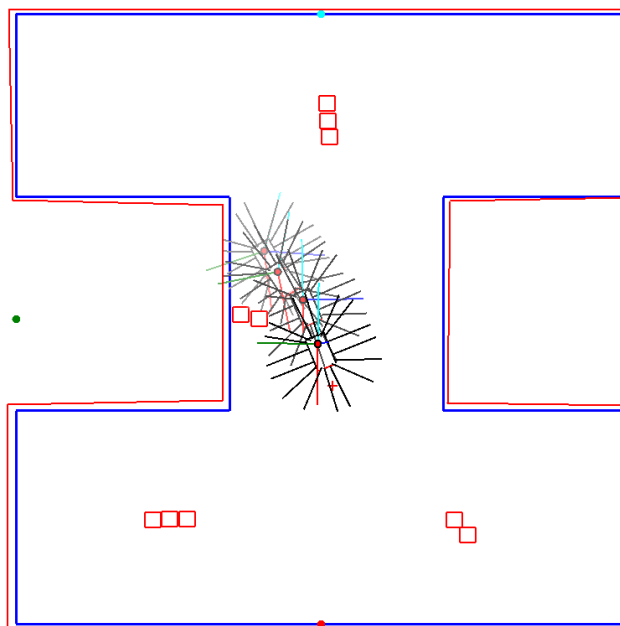


Obrázek 2: Schéma architektury softwarového a hardwarového vybavení robota. Simulované části jsou zvýrazněny čárkovaným okrajem.

plánovačem. Dalším možným využitím je zpracování dat nedeterministického plánovače, například lidského operátora. Tento případ je dále stručně popsán.

Výstup plánovače nezávisí pouze na aktuálních sensorických vstupech, ale také na stavu robota. Data pro trénování NN jsou proto upravena tak, aby obsahovala nezbytnou zpětnou vazbu. Konkrétně tak vstupy do NN tvoří údaje proximitních sensorů, úhel k cíli robota a výstupy plánovače (zobecněné translační a rotační rychlosti) v minulých krocích.

NN trénovaná na datech získaných logováním výstupů lidského operátora je po tréninku schopna robot řídit ve statickém prostředí, viz ilustrační obrázek 3. Pohyb v dynamickém prostředí je nyní předmětem dalšího zkoumání.



Obrázek 3: Pohyb robota ve virtuálním prostředí simulátoru

5 ZÁVĚR

Výše popsáný simulátor umožnil kromě testování jednotlivých softwarových modulů robota i přenesení schopnosti lidského operátora řídit simulovaný robot do softwarového samoučícího se plánovače založeného na neuronové síti, což je úloha na reálném robotu nerealizovatelná. Při nasazení tohoto plánovače v reálném robotu musí být ovšem doplněn o bezpečnostní vrstvu, například zastavení v případě překážky bližší než stanovený limit.

REFERENCE

- [1] HUANG, A.S.; OLSON, E.; MOORE, D.C. LCM: Lightweight Communications and Marshaling. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2010*, Taipei, Taiwan. 6 s.
- [2] KREJSA, J.; VĚCHET, S. Odometry-free mobile robot localization using bearing only beacons. *Proceedings of EPE-PEMC 2010 Conference, Republic of Macedonia*. T5/40-T5/45, ISBN 978-1-4244-7854-5, IEEE cat num: CFP1034A-DVD.
- [3] PERINGER, P. *SIMLIB Home Page* [online]. 2002, 2010-12-15 [cit. 2011-02-28]. SIMLIB/C++ == SIMULATION LIBRARY for C++. Dostupné z WWW: <<http://www.fit.vutbr.cz/peringer/-SIMLIB/>>