

Kognícia reálneho času

Andrej Lúčny

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského
Mlynská dolina, 842 48 Bratislava
lucny@fmph.uniba.sk

Abstrakt

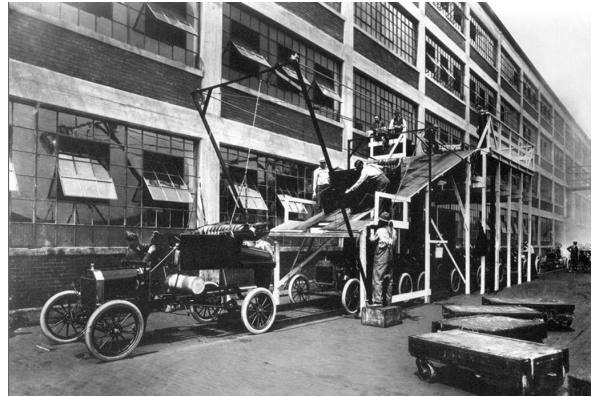
V rôznych modeloch kognitívnych systémov vystupuje čas buď ako spojitý alebo diskretný v podobe taktovania spoločného pre všetky zložky systému. V oboch prípadoch je vzťah času k prebiehajúcim výpočtom taký, ako keby akýkoľvek výpočet dokázal prebehnúť v čase obmedzenom dĺžkou taktu, či dokonca v nulovom čase. Tento prístup je cudzí pre tzv. systémy reálneho času, kde sa taktovanie jednotlivých modulov prispôbuje reálne dostupnej výpočtovej kapacite, t.j. berie sa do úvahy, že určité výpočty sú rýchlejšie a iné pomalšie. Ukážeme, že tento prístup k času vedie k alternatívnym vysvetleniam správania sa systémov a lepšie vyhovuje potrebám ich implementácie.

1 Úvod

Čím sa vyznačuje správanie živých organizmov? Odpoveď na túto otázku závisí – mimo iné - od časovej mierky. Zo strednodobého pohľadu je to účelnosť tohto správania, zatiaľ čo z dlhodobého pohľadu je to v prvom rade schopnosť toto správanie zdokonaľovať. Avšak my sa sústreďujeme na krátkodobý pohľad, ktorý odzrkadľuje najnižšie a najrýchlejšie pracujúce vrstvy riadenia, ktoré sú za správanie zodpovedné. Kým v porovnaní živých systémov s umelými môžeme (v prípade, že tie umelé systémy sú urobené dobre) vidieť aj rovnakú účelnosť, na krátkodobej škále jeden podstatný rozdiel uvidíme: stroje normálne generujú za rovnakých podmienok rovnaké správanie, zatiaľ čo živé tvory to nedokážu ani v prípade, že je to účelné. Zo strany živých systémov tu ide o určitú nepredvídateľnosť či tvorivosť ich správania.

Keď napríklad porovnáваме filmové zábery výroby Fordových závodov zo začiatku minulého storočia, kedy tam zaviedli pásovú výrobu (obr. 1) s analogickou automatickou linkou plnou priemyselných robotov zo súčasnosti (obr. 2), na prvý pohľad sú podobné, až na to, že v jednom prípade sú akčnými prvkami výroby ľudia a v druhom roboty. Je to však len preto, že vec posudzujeme zo strednodobého hľadiska: robot, rovnako ako človek urobí na správnom mieste budúceho auta správny zvar, zaskrutkuje správnu maticu a podobne. Pri detailnom pohľade však zistíme, že po chvíli sledovania konkrétneho robota, vieme veľmi dobre

predvídať dráhu jeho nasledovného pohybu. Avšak človek vždy vykoná svoj pohyb trochu inak, hoci sa zrejme snaží vykonávať pohyb stále ten istý.



Obr 1. Pásová výroba vykonávaná ľuďmi

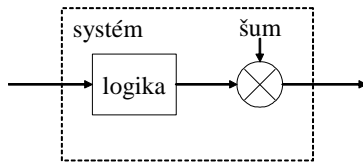


Obr 2. Pásová výroba vykonávaná robotmi

Hoci spomínaná krátkodobá tvorivosť sama o sebe k zdokonaľovaniu na dlhodobej úrovni nevedie, môže byť pre toto zdokonaľovanie cenným, ba dokonca nevyhnutným zdrojom.

2 Tvorivosť na základe šumu

Čo je zdrojom spomínanej tvorivosti či nepredvídateľnosti? V podstate každý mechanizmus, ktorý generuje správanie a dáva výstupy v nejakej spojitaj forme, je možné obohatiť o šum, ktorý spôsobuje rôzne variácie výsledného správania a pritom vďaka spätnej väzbe z prostredia, nemusí to pokaziť jeho kvalitu (obr. 3).



Obr 3. Tvorivosť na základe šumu

Systém tu sám o sebe chce konať dokonale, ale nepodarí sa mu to kvôli vlastnostiam jeho aktuátorov, dynamike prostredia a pod. Tento šum je možné generovať i umelo, s úmyslom monitorovať úspešnosť činnosti systému z nadradených úrovní a následkom tejto skúsenosti obohacovať jeho logiku. Tvorivosť je takto v podstate simulovaná, nevyplýva z vnútorných vlastností systému. Keď potlačíme šum, dostaneme dokonalé správanie.

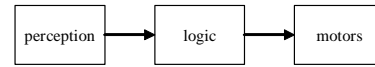
Je preto dosť pravdepodobné, že takéto predstava je skôr dobrým nástrojom na opis správania systému a nie na opis jeho skutočnej štruktúry, ktorá by sa podobala tej, čo v živých systémoch toto správanie generuje. Sľubným charakteristickým znakom takejto štruktúry by mohlo byť práve to, že nedokonalý nie je len výsledok, ale celý výpočet logiky systému. Z principiálnych dôvodov je preto možné variabilitu výsledku len znížiť, ale nedá sa úplne odstrániť.

3 Tvorivosť na základe reálneho času

Ako ukážeme, jednou z možností ako vysvetliť inherentnú prítomnosť tvorivosti v systéme je vziať v úvahu, že spracovanie informácie na rôznych miestach systému prebieha len konečnou rýchlosťou a v závislosti od zložitosti tohto spracovania môže trvať i významne dlhý čas.

Uvažujme nasledovný príklad [5]: mobilného robota vybaveného kamerou a dvojkolesovým podvozkom, ktorý má sledovať pingpongovú loptičku, ktorou pohybujeme. Už aj pri tejto jednoduchej úlohe, riešenie

obsahuje dostatočne veľa rôznych fáz spracovania na to, aby malo zmysel hovoriť o jeho modulárnom usporiadaní



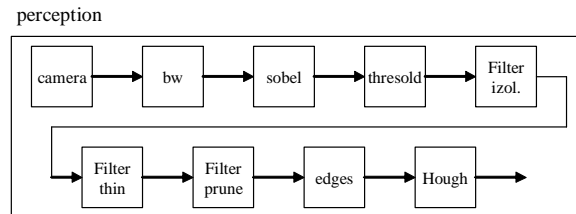
Obr 4. Základná štruktúra robota sledujúceho loptičku

Základná logika je síce jednoduchá (obr. 4):

- ak je loptička na obraze moc vpravo, vydávame príkaz na otáčanie sa doprava
- ak moc vľavo, doľava
- ak je loptička moc veľká, cúvame
- ak moc malá, ideme dopredu

ale samotné rozpoznanie stredu a polomeru vhodného kruhového objektu na obraze, ktorý reprezentuje loptičku, pozostáva z viacerých fáz (obr. 5) [3]:

- farebný obraz z kamery sa prevedie na čiernobiely
- čiernobiely obraz sa vyhraní, napríklad Sobelovým operátorom
- na základe prahu prevedieme vyhranený obraz na bitovú mapu obsahujúcu hrany
- odstránime izolované body
- hrany stenčovaním premeníme na čiary (*thinning*)
- orežeme ich (*pruning*)
- takto získanú mapu premeníme na množinu úsečiek
- pomocou Houghovej transformácie pre kružnicu získame stredy a polomery kružníc na obraze



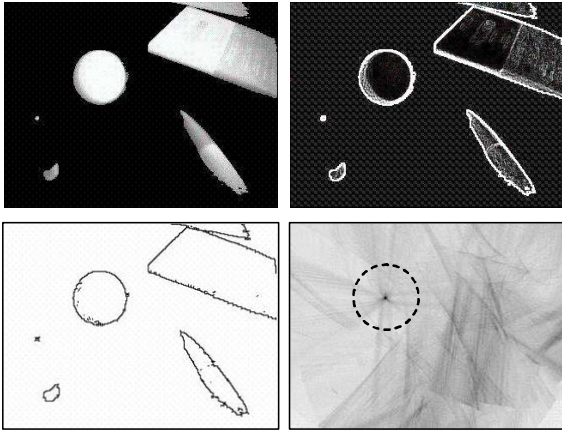
Obr 5. Štruktúra percepcie robota sledujúceho loptičku

Logika je teda rýchla a vnímanie pomalé, z menovaných fáz (obr. 6) je pomalá najmä Houghova transformácia [3].

Na úvod sme pracovali s riešením, ktoré sa snaží s reálnym časom bojovať tým spôsobom, že spomaľuje a prerušuje motorické akcie robota tak, aby malo vnímanie dostatočný čas na výpočet korektnej akcie (obr. 7)¹. Táto stratégia je inak presne to, čo nám pri porovnávaní videí zachycujúcich pásovú výrobu realizovanú robotmi a ľuďmi udrie do očí najviac: roboty robia ľahké úkony rýchlo, zatiaľ čo náročné veľmi

¹ Ako podvozok sme použili stavebnicu Boebot [1] s procesorom BasicStamp, bezdrôtovú kameru XCam2, na spojenie rádiomodemy a na ovládanie program v Jave na PC.

pomaly (obr. 2), ľudia naopak majú stále zhruba rovnakú rýchlosť².



Obr. 6. Základné fázy percepcie robota sledujúceho loptičku.

Pri tomto prístupe sme použili na ovládanie podvozku mobilného robota príkazy typu:

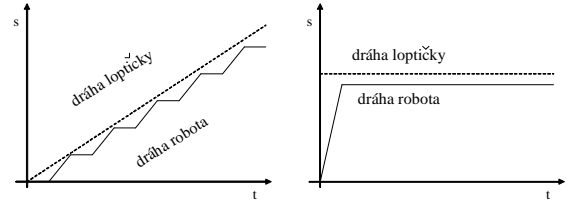
- pohni sa kúsok dopredu
- pohni sa kúsok dozadu
- otoč sa trochu doľava
- otoč sa trochu doprava

pričom sme zvolili tie správne konštanty, aby dráha, ktorú robot na jeden príkaz vykoná, nebola ani primalá, ani priveľká. Oba extrémny by totiž znefunkčnili sledovanie loptičky. Primalá dráha spôsobuje, že sa robot síce hýbe v smere loptičky, ale pomerne rýchlo stratí loptičku z dohľadu z dôvodu jej pohybu. Priveľká dráha spôsobuje, že sa robot síce pohne správnym smerom, ale preženie to a loptičku stratí z dohľadu z dôvodu vlastného pohybu.



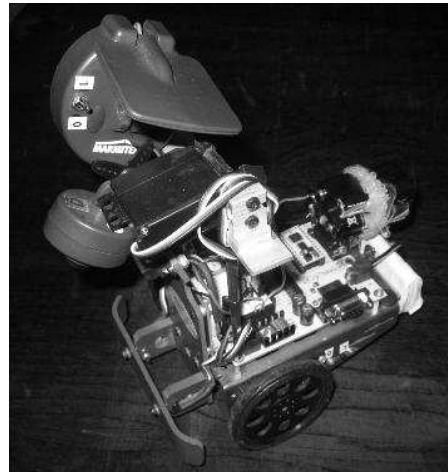
Obr. 7. Robot PingPong (1. riešenie)

Takýto robot síce dokáže sledovať pingpongovú loptičku, ale zvolený prístup sa prejavuje tým, že jeho pohyb je trhaný (obr. 8). Inak je pohyb úplne v poriadku a sleduje správnu trajektóriu.



Obr. 8. Strojové, ale korektné správanie 1. riešenia (vľavo sa loptička pohybuje kolmo k robotovi, v pravo loptička stojí)

Avšak tento pohyb v pozorovateľovi jednoznačne vyvoláva i pri tej najväčšej fantázii dojem, že sa pozerá na stroj, ktorý je svojim správaním veľmi vzdialený živému tvorovi. Preto sme sa snažili urobiť druhú verziu robota (obr. 9), ktorý by sa pohyboval kontinuálne².



Obr. 9. Robot PingPong II (2. riešenie)

Z hľadiska porovnávania videí zachycujúcich pásosú výrobu, sme sa teda snažili urobiť robota, ktorý by sa svojim pohybom viac podobal na ľudí, než na robotov. Do podvozku sme teda vysielali príkazy:

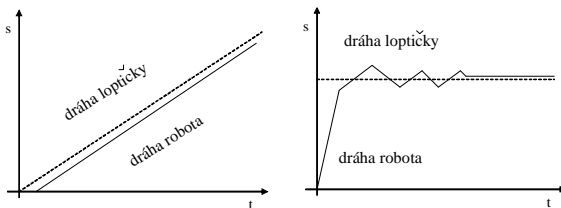
- zmeň pohyb na posun vpred
- zmeň pohyb na posun vzad
- zmeň pohyb na otáčanie doľava
- zmeň pohyb na otáčanie doprava
- stoj

² Opäť sme použili stavebnicu Boeobot [1], tentoraz však s procesorom Javelin, bezdrôtovú kameru XCam2, na spojenie bluetooth a na ovládanie program v Jave na PC

Pri takýchto príkazoch je podvozok autonómne pracujúca jednotka, ktorá reaguje na príkazy tak často, ako ich vie z vnímania odvodená logika pohybu dodávať.

Táto zmena si vyžiadala v prvom rade rozsiahle zmenu hardwaru, aby tento robot vôbec bol schopný. (Museli sme použiť iný procesor na ovládanie motorov, ktorý umožňuje v jednom vlákne vysielat' pulzy do servomotorov a v druhom komunikovať s nadradenou riadiacou jednotkou, ako aj dodávať podstatne viac energie.) Zásadný problém sa pritom vynoril v tom, že servomotory sa nedajú ľubovoľne spomaliť (pri učítom spomalení sa už proste zastavia) a použiť na to napríklad prevodovku nebolo možné kvôli miniatúrnym rozmerom robota. Keďže šlo o tie isté servomotory ako pri predchádzajúcej verzii robota a spomalili sme ich na minimum, v podstate sa točili ešte pomalšie ako predtým – lenže neustále.

Výsledkom napojenia takejto motorickej jednotky na logiku s vnímaním bolo, že robot loptičku úspešne a priamočiaro sledoval (t.j. robil rovnako úspešne to, čo predtým), pokiaľ sa hýbala určitou rozumnou rýchlosťou. Pritom jeho pohyby boli ladné, netňané, oveľa viac pripomínajúce živého tvora. Avšak pokiaľ bola loptička zastavená v pevnej pozícii, ukázalo sa, že rýchlosť pohybu je predsa len priveľká (obr. 10). Robot sa totiž priamo neobrátil k loptičke, ale začal svojím otočením oscilovať zľava doprava a späť tak, že držal loptičku niekde v strede zorného poľa. Po určitom čase náhodou dostal loptičku práve do stredu a zastal.

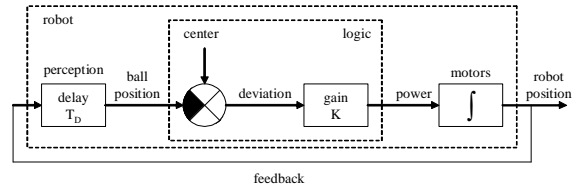


Obr. 10. Nie celkom korektné správanie 2. riešenia (vľavo sa loptička pohybuje kolmo k robotovi, v pravo loptička stojí)

Zo psychologického hľadiska je zaujímavé, že toto správanie nezmenšovalo alebo naopak umocňovalo dojem správania sa živého tvora. (Budilo to dojem, že si robot premeriava loptičku pohľadom, akým sa v prírode slabší snaží uchrániť svoje miesto pred silnejším, voči ktorému nie je úplne bezbranný.)

Z technického hľadiska je zaujímavé, že za túto osciláciu je zodpovedný výlučne ten fakt, že spracovanie vstupnej informácie (v tomto prípade obrazu) trvá určitý čas. Čo by sa stalo, keby sme teraz zabudli na fakt, že poznáme vnútornú štruktúru systému a toto správanie chceli vysvetliť modelom? Takýto model, nazvime ho – povedzme – kybernetický [2], vidíme na obr. 11.

Tento model dokáže generovať správanie hladké i oscilačné a to oscilačné tlmené i oscilačné netlmené, v závislosti od zosilnenia väzby medzi logikou a motormi K, ako aj v závislosti od oneskorenia T_D spôsobeného pomalou percepciou (obr. 12, na konci textu). Zodpovedná je za to spätná väzba realizovaná prostredníctvom robota.



Obr. 11. Kybernetický model 2. riešenia

V tomto modeli existujú pre každé T_D dve kritické hodnoty K. Pri prvej sledovanie loptičky prestane byť plynulé a nadobudne (významný) oscilačný charakter a pri druhej prejde tento oscilačný charakter z tlmeného do netlmeného priebehu.

Podľa tohto modelu by sme teda boli náchylní povedať, že pri konštrukcii 2. riešenia sme mali tú smolu, že aj to najmenšie K ktoré sme vedeli dosiahnuť, aby sa robot ešte hýbal, bolo medzi dvomi kritickými hodnotami pre T_D dané výpočtovým (a v našom prípade aj prenosovým) výkonom.

Toto však považujeme len za aproximatívnu pravdu, ktorá je pravdou len spojitom prípade, ktorý je len abstrakciou diskretnej skutočnosti. Tento model totiž v sebe neobsahuje žiadnu tvorivosť. Správanie systému, ktorý by sa podľa neho riadil, by bolo dokonale predvídateľné. Napríklad spomínaná oscilácia robota v prípade zastavenia loptičky by buď mala skončiť sa určitý presne definovaný a vždy rovnaký čas, alebo nikdy neskončiť. Stačí však robota chvíľu sledovať, aby bolo jasné, že to tak nie je. Zdá sa, akoby jeho pohyb tvorili netlmené, prípadne skoro netlmené kmity (obr. 12), ktoré náhle – v nepredvídateľný okamih – prestanú. Je pritom málo pravdepodobné, že by sme sa akurát náhodou trafili našim K do druhej kritickej hodnoty pre naše T_D (čo by pozorované správanie vysvetlilo). Samozrejme pridanie šumu na spôsob použitý v časti 2 by tú tvorivosť zachránilo, ale my sa zameriame na inú možnosť.

4 Architektúra reálneho času

Nastal čas prezradiť s akou vnútornou štruktúrou robota sme vlastne pracovali. Použili sme architektúru Agent-Space navrhnutú v [6] a použili sme ju ako pri 1. tak aj pri 2. riešení (obr 13). (Pri 1. riešení sme však nevyužili žiadnu z jej zvláštnych črt.)

Podľa tejto architektúry je aj sekvenčné spracovanie realizované spoluprácou paralelne a nezávisle na sebe pracujúcich modulov, tzv. agentov [4] [7]. Každý z nich cyklicky vykonáva výpočet transformujúci určité vstupné údaje na výstupné. Jeden prechod týmto cyklom môže nastať na základe zmeny hodnoty určitého vstupného údaju (*trigger*) alebo na základe uplynutia určitej doby od predchádzajúceho prechodu (*timer*), pričom táto druhá možnosť nie je z hľadiska modelovania v tomto článku zaujímavá. Komunikácia medzi týmito agentami je obmedzená na nepriamu: vstupno-výstupné údaje agenty preberajú z a odovzdávajú do tzv. blokov nachádzajúcich sa mimo nich. Detaily spôsobu, akým sa s blokmi pracuje, sú vo všeobecnosti dosť významné, ale nebudeme ich rozoberať a odkazujeme čitateľa na [6] alebo [5]. Pre účely tohto článku postačí vziať v úvahu, že agent zapisuje do bloku bez ohľadu na to, či prechádzajúcu hodnotu, ktorá bude týmto úkonom prepísaná, stihol iný agent prečítať alebo nie. Pokiaľ teda jeden agent zapisuje blok častejšie než ho druhý (s ohľadom na čas, ktorý potrebuje na výpočet svojej transformácie) stíha čítať, niektoré hodnoty zapísané prvým agentom sa proste stratia. Vďaka tomuto implicitnému vzorkovaniu, nie je možné robota globálne preťažiť: výsledná reakcia bude vždy reflektovať na najčerstvejší stav, ako sa len dá, vzhľadom na dostupnú výpočtovú kapacitu. Ale niekde vo vnútri, lokálne, preťaženie možné je. Napríklad pri našom 2. riešení by systém dokázal zrútať hrany obrazu častejšie, než by tento produkt mohol byť použitý Houghovou transformáciou (obr. 14).

Obr. 14 je však hlboko zavádzajúci v tom, že všetky agenty na ňom pracujú ako hodinky, t.j. že výpočet ich transformácie trvá vždy rovnaký čas. Pre niektoré agenty to pravda je: napríklad agent, ktorý mení farebný obraz na čiernobiely, vždy urobí toľko operácii, koľko je bodov na obraze. Ale agent, ktorý nachádza hrany (podobne aj Houghova transformácia) pracuje v závislosti od toho, koľko ich na obraze je. Napriek snahe všetkých agentov pracovať ako hodinky sa prostredie do tejto činnosti premietne a spôsobí, že agenty vôbec nebudú pracovať ako hodinky: ich práca sa bude vyznačovať časovou variabilitou. A tá sa premietne v konečnom dôsledku do variabilného pohybu motorov.

Navyše, vplyv prostredia na priebeh výpočtu v robotovi je závislý od polohy robota v prostredí a táto poloha je závislá od výsledku výpočtu robota. V zložitejšom prostredí – čím bohatšie, tým lepšie – sa preto môžu objavovať mnohoraké spätné väzby. Oscilácie robota nie sú tlmené, ale tvorivé. Pozorovaná nepredvídateľnosť pohybu robota nie je spôsobená tým, že by zámerne experimentoval, ale tým, že dokonalé správanie nie je – vzhľadom na obmedzenú výpočtovú a prenosovú kapacitu – v reálnom čase možné.

Pri zafixovaní polohy loptičky by takýto robot mohol okolo tejto loptičky oscilovať donekonečna len načisto teoreticky. Čím viac času uplynie, tým väčšia je pravdepodobnosť, že robot zachytí loptičku vo vhodnej polohe a zastane. A presne to aj pozorujeme, keď tohto robota sledujeme.

5 Zdokonaľovanie sa

Máme teraz určitú predstavu o zdroji tvorivosti v systéme reálneho času. Vieme ju teraz použiť k zdokonaľovaniu sa tohto systému? Keby sme to nevedeli, tak by naša štruktúra bola len neopraviteľným pokazením ideálneho kybernetického modelu. Našťastie to nie je ťažké, nakoľko architektúra Agent-Space [6] nám tu pomôže vlastnosťami, pre ktoré bola pôvodne vytvorená a ktoré podporujú inkrementálnu tvorbu systému.

Základná myšlienka spočíva v tom, že ak Houghova transformácia nedokáže dostatočne rýchlo pre logiku riadiacu motory pracovať, potom sa táto logika nemôže opierať výlučne o ňu. Ak sa aj logike nedostáva nových údajov o relatívnej polohe loptičky, môže dočasne pracovať s odhadom týchto hodnôt. Pokyn vydaný do motorov je bezpochyby použiteľný k odhadu pohybu loptičky na obraze, lebo ak sa robot hýbe napríklad doprava, loptička sa zákonite na obraze hýbe doľava. Keby sme vedeli, že loptička stojí, stačilo by raz odmerať konštantu vyjadrujúcu pomer rýchlosti motorov ku rýchlosti pohybu loptičky na obraze. Ale loptička sa môže aj sama o sebe hýbať. Predikcia je teda o niečo zložitejšia. Samozrejme, nie natoľko, že by bolo nemožné postaviť predikciu na čistej logike. Ale ponúka sa tu aj iná možnosť: využiť detekciu skutočnej polohy na kalibráciu predikcie. Variabilný pohyb neskalibrovaného robota dáva dostatočne bohaté dáta aby kalibrácia mohla úspešne prebehnúť a významne zapojiť predikciu do konania robota (obr. 15). Týmto smerom by sme sa chceli uberať v ďalšej práci.

Inú možnosť ako zlepšovať pohyb robota predstavuje opretie kalibrácie o rozpoznanie iného objektu na obraze, ktorý možno rozpoznať rýchlejšie a teda aj častejšie než loptičku. Napríklad pohyb vertikálnych hrán na obraze dobre vyjadruje ako sa na obraze loptička hýbe následkom pohybu robota (obr. 16). Čím komplexnejšie deje v riadiacej jednotke robota budú prebiehať, tým lepšie zlepšenie sa z toho dá vydolovať.

6 Záver

V tomto článku sme sa snažili poukázať na význam reálneho času pre tvorbu umelých systémov a štúdiu systémov živých. Pokiaľ sa priblížime s architektúrou umelého systému k decentralizovanému paralelnému spracovaniu informácie, ktoré môžeme pozorovať na

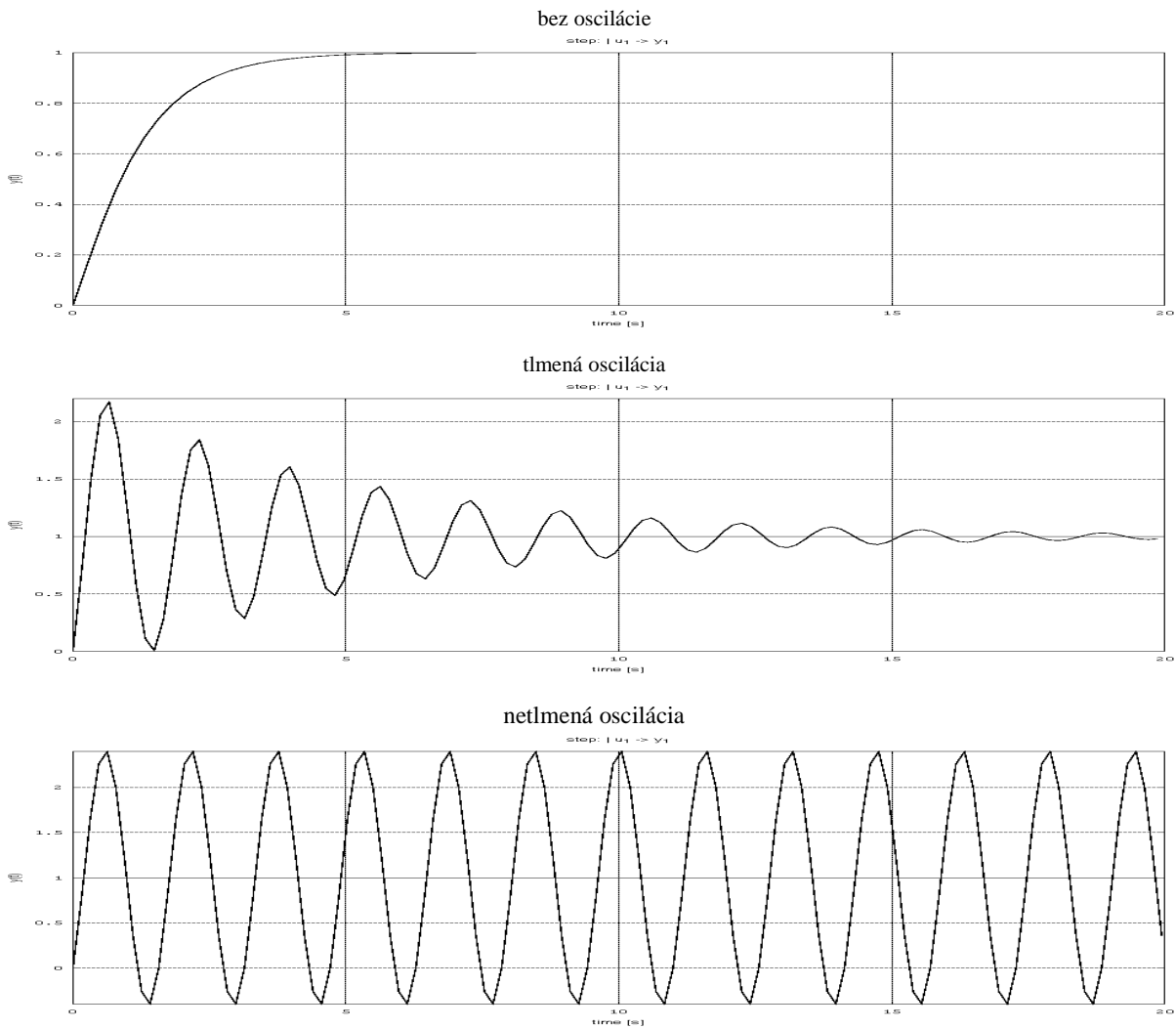
živých systémech v přírode, objavia sa aj v umelom systéme problémy s dosiahnutím správania, ktoré nevybočuje z optimálneho priebehu. Tieto vybočenia nepochádzajú pritom zo zlého nastavenia parametrov systému, ale z nemožnosti plne realizovať správne nastavenie v reálnom čase. Táto zdanlivá nevýhoda môže byť prínosom, pokiaľ k systému vieme postaviť ďalšie paralelné moduly, ktoré tieto vybočenia používajú na objavovanie rýchlejších a lepších postupov a prenášajú tieto objavy do výsledného správania. V podstate sme narazili len na to, čo nám aj bez experimentovania vraví zdravý rozum: že dokonalosť je najväčšou prekážkou zdokonaľovania sa. Nemali by sme sa teda báť faktu, že nedokonalosti sa objavujú už vplyvom práce v reálnom čase a v reálnom prostredí, ale naučiť sa tento fakt tvorivo využívať.

Pod'akovanie

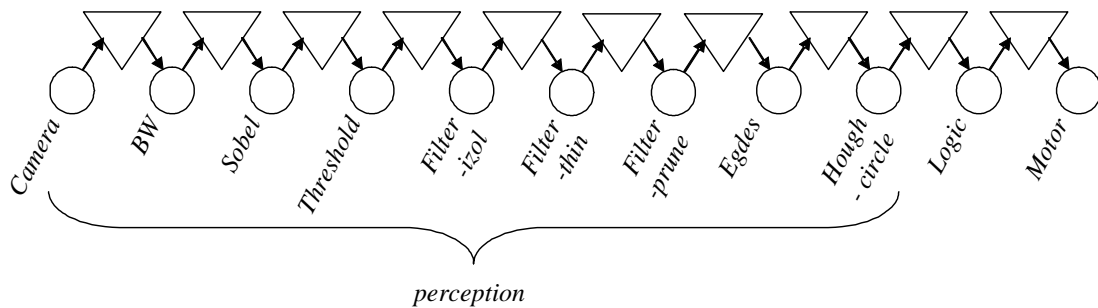
Tento príspevok vznikol s podporou grantu VEGA 1/0280/08 *Aplikácia multi-agentovej modularity pre tvorbu riadiacich systémov a počítačových modelov*. Ďakujem taktiež Richardovi Baloghovi z URPI FEI STU v Bratislave za know-how z oblasti kybernetiky a analýzu kybernetického modelu.

Literatúra

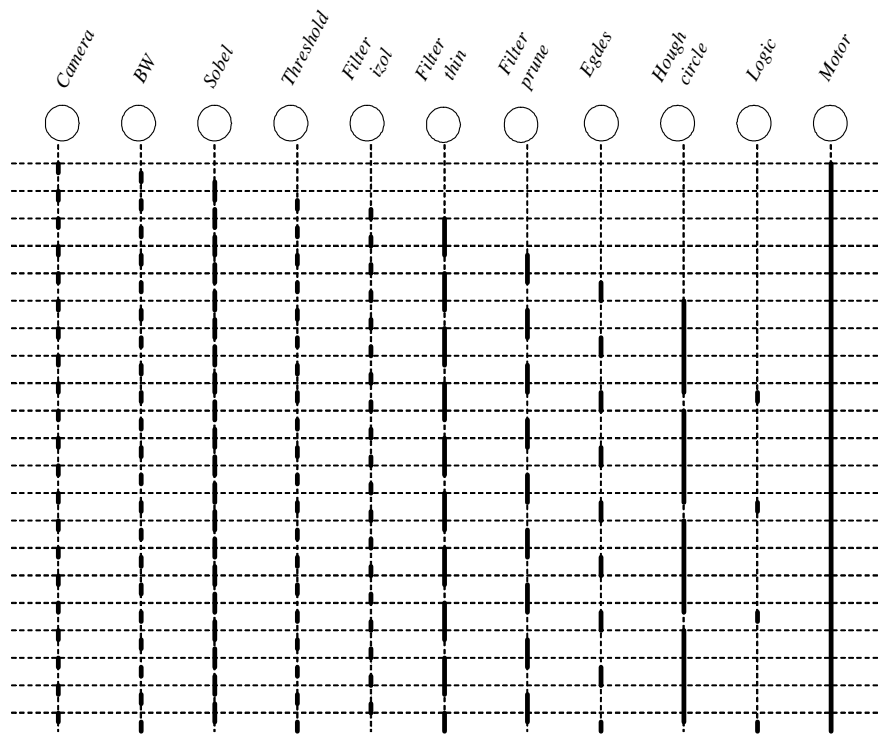
- [1] Balogh, R.: *Basic Activities with the Boe-Bot Mobile Robot*, In DidInfo 2008. 14th International Conference. FPV UMB, Banská Bystrica, Slovakia, 2008
- [2] Cruze, H.: *Neural Networks as Cybernetic Systems*. Brains, Minds & Media, Bielefeld, 2006
- [3] Davies, E. R.: *Machine Vision: Theory, Algorithms, Practicalities.*, Elsevier, 2005
- [4] Kelemen, J.: *Multiagent symbol systems and behavior-based robots*. Applied Artificial Intelligence 7, (1993), 419-432.
- [5] Lúčny, A.: *Od medzimodulových spojení k nepriamej komunikácii medzi agentami*. Znalosti, VŠE Ostrava, 2007.
- [6] Lúčny, A.: *Building Complex Systems with Agent-Space Architecture*. Computing and Informatics, Vol. 23 (2004), pp. 1001-1036
- [7] Minsky, M.: *Society of Mind*. Simon & Schuster, New York, 1986



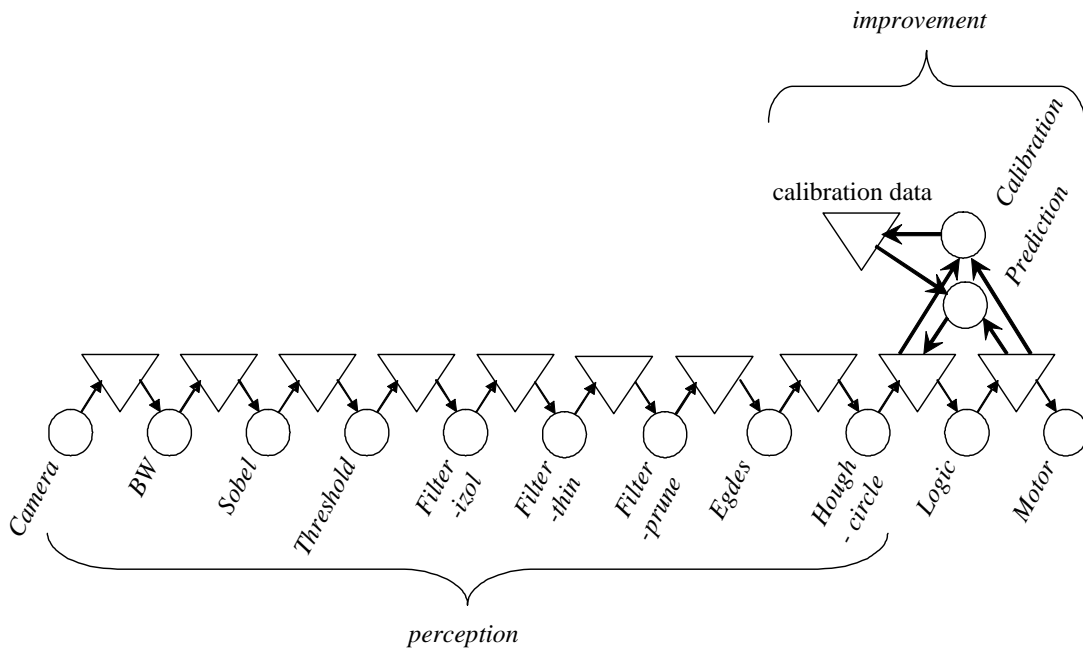
Obr. 12. Rôzne druhy pohybu robota podľa modelu



Obr. 13. Vnútna štruktúra riadiaceho systému robota



Obr. 14. Približný stavový diagram agentov v 2. riešení (čas plynie zhora nadol, plné zvislé čiary predstavujú čas po ktorý agent vypočítava svoju reakciu, prerušované čas po ktorý je agent nečinný, vodorovné čiary zodpovedajú príjmu nového obrázka z kamery, motory sú riadené kontinuálne podľa posledného údaju z logiky)



Obr. 15. Rozšírenie riešenia za účelom zdokonaľovania.