

Szimulált robot raj vezérlése

Sapientia

Erdélyi Magyar Tudományegyetem, Marosvásárhely

Patka Zsolt-András

2020

Kivonat

Abstract

Extras

Tartalomjegyzék

1. Bevezető	1
2. A dolgozat célja	2
3. Szakirodalmi háttér bemutatása	3
4. Felhasznált Szoftverkeretrendszerek	4
4.0.1. Robot szimulátorok adta lehetőségek tanulmányozása	4
5. Gyakorlati megvalósítás	6
5.1. Célkövetés és akadálykerülés a push-pull erők használatával	6
5.2. Célkövetés és akadálykerülés QLearning használatával	6
5.2.1.	6
6. A rendszer tesztelése	7
7. Következtetések	8

1. fejezet

Bevezető

A rajintelligencia alapötlete, hogy számos limitált számításikapacitással rendelkező agensek képesek egy komplex feladat megoldására. Az ötlet az állatvilágban megvizsgált szószerinti rajok tanulmányozásából született. Rajokba szerveződnek a hangyák, méhek, madarak, halak és sok más állatfajta. A méhek esetén kimutatható, hogy egy olyan komplex feladat megoldásában, mint például a kaptár optimális helyének a kiválasztásában, az esetek hozzávetőlegesen 80 százalékában optimális megoldásra jutnak. Az állatvilágot utánozva jelentek meg ennek az alapötletnek számos alkalmazásai nem csak robotikában, hanem a számítástechnika számos ágazatában (TODO: példák sorolása). E ötlet követése által lehetséges olyan osztott algoritmusok kidolgozása, amelyek implementálása által sok limitált számításikapacitással rendelkező agens egy komplex feladat megoldására képes.

2. fejezet

A dolgozat célja

3. fejezet

Szakirodalmi háttér bemutatása

4. fejezet

Felhasznált Szoftverkeretrendszerek

Egy szimulációskörnyezet szükséges és előnyös rajntelligenciás osztott algoritmusok kidolgozásához. Jóval lerövidíti a fejlesztési időt, mivel egyrészt nem szükséges számos robot megvásárlása, másrészt a robotok felprogramozása összemérhetetlenül gyorsabban történik.

4.0.1. Robot szimulátorok adta lehetőségek tanulmányozása

A robotszimulációs környezet világában két nyílt forráskódu, elterjedtebb szoftver között lehet választani: az ARGoS3 és a V-REP. A következőkben szemléltetve lesz a két szimulációs környezet előnyei, hátrányai.

V-REP <http://www.coppeliarobotics.com/>

- Támogatott nyelvek robot kontrollerlogikájának megírására: C/C++, Python, Java, Lua, Matlab, Octave
- Rendkívül valósághű
- Ingyenes, van fizetős verzió is
- Robotrajokra nincs optimalizálva
- Aktívan fejlesztik, jelennek meg új verziók
- Nagy a felhasználók száma
- Nyílt forráskódu

ARGoS3 <https://www.argos-sim.info/index.php>

- Támogatott nyelvek robot kontrollerlogikájának megírására: C/C++ és Lua
- Nagyobb az absztrakciós szint, nem annyira valósághű
- Robotrajokra teljesen optimalizálva van
- Ingyenes
- Fejlesztés alatt áll, bár az új verziók nem konszisztensen érkeznek (2019 júliusán jött ki új verzió, az ezelőtti verzió 2016-os)
- Viszonylag kevesen használják, nehezebb a hibákra megoldást találni

- Nyílt forráskódu

A két robotszimulátor között a választás az ARGoS-ra esett, mivel egy nagyobb absztrakciós szintet ajánl, ezért könnyebb magára az algoritmusra, a logikára fektetni a hangsúlyt. Ez a nagyobb absztrakciós szint ahhoz is vezet, hogy egy sok robotot tartalmazó robotraj esetén a számítási kapacitás közel se olyan sok, mint a V-REP-nél.

5. fejezet

Gyakorlati megvalósítás

5.1. Célkövetés és akadálykerülés a push-pull erők használatával

A push-pull erők elv követésével felírhatóak azok az erők amik a cél fele mutatnak, mint húzó erők, és azok amik az akadály felé mutatnak, mint taszító erők.

5.2. Célkövetés és akadálykerülés QLearning használatával

5.1. táblázat. Jutalommatrix, Akkadálykerülő és célkövető agensnek

Állapot	<i>STOP</i>	<i>TURN_LEFT</i>	<i>TURN_RIGHT</i>	FORWARD
FOLLOW	-1	0	0	1
UTURN	-1	0	0	0
OBST_LEFT	-1	0	0.1	0
OBST_RIGHT	-1	0.1	0	0
OBST_FORWARD	-1	0.1	0	0
WANDER	-1	0	0	0.1
IDLE	-1	0	0	1

5.2.1.

6. fejezet

A rendszer tesztelése

7. fejezet

Következtetések

Irodalomjegyzék