Wheeled Robots playing Chain Catch: Strategies and Evaluation

(Extended Abstract)

Garima Agrawal
International Institute of Information Technology
Hyderabad, India
garima.agrawal@research.iiit.ac.in

Kamalakar Karlapalem
IIIT Hyderabad/ IIT Gandhinagar
India
kamal@iiit.ac.in

ABSTRACT

Robots playing games that humans are adept in is a challenge. We studied robotic agents playing Chain Catch game as a Multi-Agent System (MAS). Chain Catch is a combination of two challenges - pursuit domain and robotic chain formation. In this paper, we present a Chain Catch simulator that allows us to incorporate game rules, design strategies and simulate the game play. We developed cost model driven strategies for each of Escapee, Catcher and Chain. Our results, simulation and robots implementation show that Sliding slope strategy is the best strategy for Escapees whereas Tagging method is the best method for chain′s movement in Chain Catch.

General Terms

Design, Algorithms, Experimentation, Performance

Keywords

Strategies, Multi-agent games, Simulation, Robots, Heuristics

1. INTRODUCTION

We implement robotic agents playing Chain Catch, which is a common multi-player playground game that requires strategic decision making and cooperation among chain members to stay together (as a chain) while catching another player whereas other players to compete with chain to escape from getting caught. Simulating robot games like Robosoccer and Robot pursuit evasion games have been a topic of extensive research in the field of Multi-Robot systems [6]. Our game starts as simple Catch-Catch or "tag" game that falls under pursuit domain problems. In our Chain Catch game (i) the Catcher Catches one of the Escapees, (ii) the Catcher and caught Escapee form a chain to Catch other Escapees and (iii) step (ii) is repeated until all Escapees are caught and become one chain. Chain Catch requires complex and efficient strategies for the Escapee and chain, we also developed techniques for robotic chain formation and movement suitable in game scenario. Our Chain Catch agents are autonomous and compute their strategy in a decentralized manner.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J.Thangarajah, K.Tuyls, C.Jonker, S.Marsella (eds.), May 9–13, 2016, Singapore.*

Strategy name	Cost function Description with respect to Catcher
Max distance	maximize their distance
K circle	Form a circle with radius equal to "K"
K circle with rotation	form a K circle and rotate around it
Sliding slope	K circle strategy along with sliding slopes at the corners

Table 1: Summarizing strategies for Escapees.

1.1 Related Work

Korf suggested a standard solution to the pursuit problem [4], which is a motivation behind Max Distance strategy for our Escapees. Game theoretical approaches can also be used for prey-predator games [3]. But however this approach is centralized unlike our decentralized multi-agent system. Robot control aspects of forming chain is discussed in [5].

2. AGENT STRATEGIES

We use a cost model to develop strategies for each of Escapee, Catcher and Chain. Lesser the cost of a cell, better it is for the agent to move into it. Catcher's strategy is based on computing the cell closest to the nearest Escapee. Escapees' strategies involve maintaining safe distance from the Catcher/chain while achieving an implicit formation among fellow Escapees. Table 1 summarizes all strategies designed for Escapee agents . Member agents of the chain have dual objectives- (i) Catch an Escapee (ii) maintain chain formation. We have designed two strategies (Table 2) for chain members keeping the two objectives under consideration.

3. ROBOT SIMULATION

We use production quality Nex Robotics platforms Fire Bird- V ATMEGA2560 with Xbee API module. Our robotic setup does not have localization mechanism therefore, we implement virtual localization through communication. These robots are similar in terms of size, speed (same and constant) and behavior. Users have to place the robots onto the specified starting location to begin the game. Once the game begins, the robots compute the best move possible depending upon information it has about other agents and by using its Strategy Engine module. We have six robots; and imple-

Strategy name	Description
Tagging method (CC)	Leader moves towards nearest Escapee. Other members tag themselves to their neighbour in direction of Leader and move to the cell closest to it
Variance method (CC)	All members try to attain different *variance* distances from Escapee to surround it while maintaining safe distance R_c from their neighbours
Tagging method (AC)	Leader moves towards nearest Escapee. Other members to its left and right tag themselves to their neighbour in direction of Leader
Variance method (AC)	All members try to attain different *variance* (as defined for case-2) distance from Escapee while maintaining R_c from neighbours

Corner catch = CC, Any one catch = AC

Table 2: Chain strategies

mented Catcher, Escapee and Chains algorithms on these robots in different scenarios and examined the performance (see Figure 1).

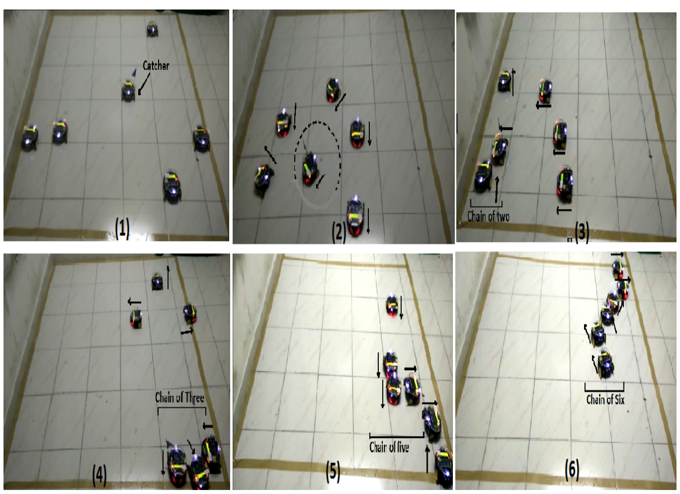

Figure 1: Robot simulation of Chain game with chains Tagging method and Escapees moving with K circle strategy. (1) Initial condition. (2) K circle around Catcher. (3) First Catch (4) Chain of three (5) Chain of five. (6) Chain completion

Live videos of some of the game play experiments can be found in reference section [1].

4. EXPERIMENTAL RESULTS

We built a front-end simulator to design and experiment with various Chain Catch game strategies. Total time (T_c), is the time taken till the last Escapee gets caught. Chain aims to lower T_c while Escapees aim at higher T_c. We performed over 100 empirical experiments, with varying number of agents from 3 to 100, and different starting locations of agents in each case. Transition table shown in Table 3 gives overall analysis of game results as average number and standard deviation of steps taken for the complete chain formation. The data >3000 implies that the strategy is insignificant; as it takes more than maximum possible steps (3000) to finish the game. Sliding slope is the best strategy for Escapees while Tagging method is the best strategy for chain.

escapee / chain	Naive strategy	K circle strategy without rotation	K circle strategy with rotation	Sliding slope strategy	Random strategy
Tagging strategy (CC)	197.39 ± 88.44	566.17 ± 314.88	290.6 ± 124.29	673.6 ± 323.98	139.41 ± 79.09
Variance strategy (CC)	308.78 ± 277.42	720.25 ± 307.62	503.66 ± 246.02	837.38 ± 278.22	249.89 ± 200.12
Random strategy	>3000	>3000	>3000	>3000	962.44 ± 639.91
Tagging strategy (AC)	136.9 ± 107.73	475.6 ± 300.33	250.14 ± 114.83	553.17 ± 320.14	110 ± 89.23
Variance strategy (AC)	240.17 ± 151.14	613.5 ± 289.24	449.66 ± 203.34	710.44 ± 249.78	200.2 ± 144.53

Corner catch = CC, Any one catch = AC

Table 3: Average number and standard deviation of steps (T_c) taken.

5. SUMMARY

We built a Multi-robot system where robotic agents are capable to play Catch-Catch and Chain Catch. We implemented the system both as simulation framework and in physical environment with real robots. An example of Chain Catch is where, to trap a terrorist the robots might have to form a chain and move in a coordinated fashion or even surround it with a circular formation as done by our chain and escapees. [2] is the extended version of this work.

REFERENCES

[1] Simulation and robot implementation of various game strategies. *https://goo.gl/8nIyBG*, 2015.

[2] G. Agrawal and K. Karlapalem. Wheeled robots playing chain catch: Strategies and evaluation (full paper - arxiv). *http://arxiv.org/abs/1602.06460*, 2016.

[3] R. E. Korf. A game theoretic approach to the pursuit problem. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, pages 195–213.

[4] R. E. Korf. A simple solution to pursuit games. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, pages 183–194, 1992.

[5] R. Mead, J. B. Weinberg, , and J. R. Croxell. A demonstration of a robot formation control algorithm and platform. *22nd National Conference on Artificial Intelligence, Vancouver, BC*, 2007.

[6] F. L. Wurtz. Ai in prey predator domain. Master's thesis, Technical University of Denmark, 2008.