Engineering Applications of Artificial Intelligence 56 (2016) 14–29

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Fictitious play for cooperative action selection in robot teams

M. Smyrnakis *, S.M. Veres

Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom

ARTICLE INFO

Article history:
Received 13 June 2015
Received in revised form
23 February 2016
Accepted 9 August 2016

Keywords:
Robot team coordination
Fictitious play
Extended Kalman filters
Game theory
Distributed optimisation

ABSTRACT

A game-theoretic distributed decision making approach is presented for the problem of control effort allocation in a robotic team based on a novel variant of fictitious play. The proposed learning process allows the robots to accomplish their objectives by coordinating their actions in order to efficiently complete their tasks. In particular, each robot of the team predicts the other robots' planned actions, while making decisions to maximise their own expected reward that depends on the reward for joint successful completion of the task. Action selection is interpreted as an n-player cooperative game. The approach presented can be seen as part of the *Belief Desire Intention* (BDI) framework, also can address the problem of cooperative, legal, safe, considerate and emphatic decisions by robots if their individual and group rewards are suitably defined. After theoretical analysis the performance of the proposed algorithm is tested on four simulation scenarios. The first one is a coordination game between two material handling robots, the second one is a warehouse patrolling task by a team of robots, the third one presents a coordination mechanism between two robots that carry a heavy object on a corridor and the fourth one is an example of coordination on a sensors network.

1. Introduction

Recent advances in industrial automation technology often require distributed optimisation in a multi-agent system where each agent controls a machine. An application of particular interest, addressed in this paper through a game-theoretic approach, is the coordination of robot teams. Teams of robots can be used in many domains such as mine detection (Zhang et al., 2001), medication delivery in medical facilities (Evans and Krishnamurthy, 1998), formation control (Raffard et al., 2004) and exploration of unknown environments (Simmons et al., 2000; Madhavan et al., 2004). In these cases teams of intelligent robots should coordinate in order to accomplish a desired task. When autonomy is a desired property of a multi-robot system then self-coordination is necessary between the robots of the team. Applications of these methodologies also include wireless sensor networks (Makarenko and Durrant-Whyte, 2004; Kho et al., 2009a; Zhang et al., 2004; Kho et al., 2009b), smart grids (Voice et al., 2011; Ayken and Imura, 2012), water distribution system optimisation (Zecchin et al., 2006) and scheduling problems (Stranjak et al., 2008).

Game theory has also been used to design optimal controllers when the objective is coordination, see e.g. Semsar-Kazerooni and Khorasani (2009). Using this approach, in Semsar-Kazerooni and Khorasani (2008, 2009) the agents/robots eventually reach the Nash equilibrium of a coordination game. Another approach is presented in Bauso et al. (2006), based on agents' cost functions, which use local components and the assumption that the states of the other agents are constant.

Fictitious play is an iterative learning process where players choose an action that maximises their expected rewards based on their beliefs about their opponents' strategies. The players update these beliefs after observing their opponents' actions. Even though fictitious play converges to the Nash equilibrium for certain categories of games (Robinson, 1951; Miyasawa, 1961; Nachbar, 1990; Monderer and Shapley, 1996; Fudenberg and Levine, 1998), this convergence can be very slow because of the assumption that players use a fixed strategy in the whole game (Fudenberg and Levine, 1998). Speed up of the convergence can be facilitated by an alternative approach, which was presented in Smyrnakis and Leslie (2010), where opponents' strategies vary through time and players use particle filters to predict them. Though providing faster convergence, this approach has the drawback of high computational costs of the particle filters. In applications where the computational cost is important, as the coordination of many UAVs, the particle filters approach is intractable. An alternative, that we propose here, is to use extended Kalman filters (EKF) instead of predicting opponents' strategies using particle filters. EKFs have much smaller computational costs than the particle filter variant of fictitious play algorithm that has been proposed in Smyrnakis and Leslie (2010). Moreover, in contrast to Smyrnakis and Leslie (2010), we provide a proof of convergence to Nash equilibrium of

* Corresponding author.
E-mail addresses: m.smyrnakis@sheffield.ac.uk (M. Smyrnakis), s.veres@sheffield.ac.uk (S.M. Veres).

http://dx.doi.org/10.1016/j.engappai.2016.08.008
0952-1976/

the proposed learning algorithm for potential games. Potential games are of particular interest as many distributed optimisation tasks can be cast as potential games. Hence convergence of an algorithm to the Nash equilibrium of a potential game is equivalent to convergence to the global or local optimum of the distributed optimisation problem.

Thus the proposed learning process can be used as a design methodology for cooperative control based on game theory, which overlaps with solutions in the area of distributed optimisation (Chapman et al., 2011a). Each agent i strives to maximise a global control reward, as negative of the control cost, through minimising its private control cost, which is associated with the global one. The private cost function of an agent i incorporates terms that not only depend on agent i, but also on costs associated with the actions of other agents. As the agents strive to minimise a common cost function through their individual ones, the problem addressed here can also be seen as a distributed optimisation problem. In this work we enable the agents to learn how they minimise their cost function through communication and interaction with other agents, instead of finding the Nash equilibrium of the game, which is not possible in polynomial time for some games (Daskalakis et al., 2006). In the proposed scheme robots learn and change their behaviour according to the other robots' actions. The learning algorithm, which is based on fictitious play (Brown, 1951), serves as the coordination mechanism of the controllers of team members. In the proposed cooperative control methodology there is an implicit coordination phase where agents learn other agents' policies and then they use this knowledge to decide on the action that minimises their cost functions. Additionally the proposed control module can be seen as a part of the BDI framework since agents update their beliefs about their opponents' strategies given the state of the environment.

The remainder of this paper is organised as follows. We start with a brief description of relevant game-theoretic definitions. Section 3 presents some background material about rational agents. Section 4 introduces the learning algorithm that we use in our cooperative game-based robot cooperation controller, Section 5 contains the main theoretical results and Section 6 contains the simulation results in order to define the parameters of the proposed algorithm. Section 7 presents simulation results before conclusions are drawn.

2. Game theoretical definitions

In this section we will briefly present some basic definitions from game theory, since the learning block of our controller is based on these. A game Γ is defined by a set of players $\mathcal{I}$, $i \in \{1, 2, \ldots, \mathcal{I}\}$, who can choose an action, s^i, from a finite discrete set S^i. We then can define the joint action s, $s = (s^1, \ldots, s^{\mathcal{I}})$, that is played in a game as an element of the product set $S = \times_{i=1}^{i=\mathcal{I}} S^i$. Each player i receives a reward, r^i, after choosing an action s^i. The reward, also called the utility, is a map from the joint action space to real numbers, $r^i: S \to R$. We will often write $s = (s^i, s^{-i})$, where s^i is the action of player i and s^{-i} is the joint action of player i's opponents. When players select their actions using a probability distribution they use mixed strategies. The mixed strategy of a player i, σ^i, is an element of the set Δ^i, where Δ^i is the set of all the probability distributions over the action space S^i. The joint mixed strategy, σ, is then an element of $\Delta = \times_{i=1}^{i=\mathbb{I}} \Delta^i$. A strategy where a specific action is chosen with probability 1 is referred to as pure strategy. Analogously to the joint actions we will write $\sigma = (\sigma^i, \sigma^{-i})$ for mixed strategies. The expected utility a player i will gain if it chooses a strategy σ^i (resp. s^i), when its opponents choose the joint strategy σ^{-i}, is denoted by $r^i(\sigma^i, \sigma^{-i})$ (resp. $r^i(s^i, \sigma^{-i})$).

Table 1
Rewards of two players in a zero sum game as function of the outcome of throwing a coin: matching pennies game.

	Head	Tails
Head	1, −1	−1,1
Tails	−1,1	1, −1

Table 2
Rewards of two players in a simple coordination game as function of joint moves to the left and up (L,U) or right and down (R,D).

	L	R
U	1,1	0,0
D	0,0	1,1

A game, depending on the structure of its reward functions, can be characterised either as competitive or as a coordination game. In competitive games players have conflicted interest and there is not a single joint action where all players maximise their utilities. Zero sum games are a representative example of competitive games where the reward of a player i is the loss of other players. An example of a zero-sum game is presented in Table 1. On the other hand in coordination games players either share a common reward function or their rewards are maximised in the same joint action. A very simple example of a coordination game where players share the same rewards is depicted in Table 2. Even though competitive games are the most studied games, we will focus our work on coordination games because they naturally formulate a solution to distributed optimisation and coordination.

2.1. Best response and Nash equilibrium

A common decision rule in game theory is best response. Best response is defined as the action that maximises players' expected utility given their opponents' strategies. Thus for a specific mixed strategy σ^{-i} we evaluate the best response as:

$$\hat{\sigma}^i_{pure}(\sigma^{-i}) = \underset{s^i \in S}{\arg\max} \quad r^i(s^i, \sigma^{-i}) \tag{1}$$

A joint mixed strategy $\hat{\sigma} = (\hat{\sigma}^i, \hat{\sigma}^{-i})$ is called a Nash equilibrium. Nash (1950) showed that every game has at least one equilibrium which satisfies:

$$r^i(\hat{\sigma}^i, \hat{\sigma}^{-i}) \geq r^i(\sigma^i, \hat{\sigma}^{-i}) \quad \text{for any } i \text{ and } \sigma^i \in \Delta^i \tag{2}$$

Eq. (2) implies that if a strategy $\hat{\sigma}$ is a Nash equilibrium then it is not possible for a player to increase their utility by unilaterally changing its strategy. When all the robotic players in a game select their actions using pure strategies then the equilibrium is referred to as pure Nash equilibrium.

2.2. Distributed optimisation by potential games

It is possible to cast distributed optimisation problems as potential games (Arslan et al., 2007; Chapman et al., 2011b), thus the task of finding an optimal solution for a distributed optimisation problem can be seen as the search for a Nash equilibrium in a game. An optimisation problem can be solved distributively if it can be divided into $\mathcal{D}$ coupled or independent sub-problems with the following property (Bertsekas, 1982):

$$r(s) - r(\tilde{s}) > 0 \Leftrightarrow r^i(s^i) - r^i(\tilde{s}^i) > 0, \quad i \in \mathcal{I}, \quad \forall\, s, \tilde{s} \tag{3}$$