

Decision-Making In Real-Time, Noisy, Adversarial and Pressing Multi-Agent System

Gu Yang, Liu Junfeng, Wang Qinglin

Department of Automatic Control, Beijing Institute of Technology, Beijing, 100081

Abstract: The architecture and decision phase paradigm of decision making in real-time, noisy, adversarial and pressing multi-agent system is presented as the main contribution of this article. Intelligence phase is implemented as world model, design phase as task decomposition, off-line learning, and on-line programming, and choice phase as jointly decision. Robotic soccer simulation, which is a domain that fits the above characteristics, is taken as a case study to verify its effectiveness. In addition, our simulation team was runner-up at the RoboCup2002 competition, in which 44 teams participated. It achieves a total score of 91-13 over 14 different games, and successfully demonstrated its flexible structure in the combination of local optimization and adversarial consideration.

Keywords: decision-making, RTNAP, MAS, LOAC, RoboCup.

1 Introduction

Every decision must go through the 3 phases shown in Figure 1. These phases, called intelligence, design and choice, were first defined by Herbert Simon ^[1, 2]. The intelligence phase consists of finding, identifying, and formulating the problem or situation that calls for a decision. The design phase is where alternatives are developed. And in the choice phase, alternatives are evaluated so that one of them is chosen as final decision.

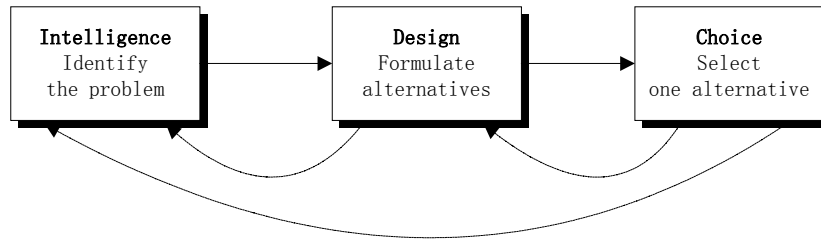


Fig. 1 Flow among Three Decision Phases

In a Multi-Agent System (MAS), especially a Real-Time, Noisy, Adversarial, and Pressing (RTNAP) domain, decision making of an individual agent is real-time and challenging ^[3]. In such a dynamically changing domain, agents cannot accurately perceive the world, nor can they accurately affect it. A group of agents coordinate in order to achieve a common long-term goal, while other agents share competing goals in the same environment.

Though the decision process still incorporates each phase at least minimally, the emphasis on each phase, and the relationships among the phases, often differ in various situations. Since agent is an entity that senses, thinks, and acts, the intelligence phase can be concluded as the stage of sense, and the design and choice phase as the successive stage of think.

In this paper, decision-making architecture is presented in a RTNAP MAS. And robotic soccer is a domain which fits the above characteristics while being both accessible and suitably

complex. The decision-making architecture is implemented in the development of Everest 2002 RoboCup simulation agent. As a case study, a high-level skill of forwarding is detailed and experiment is conducted to verify its effectiveness.

This article is organized as follows. Section 2 presents the architecture and decision phase paradigm of real-time decision making in RTNAP MAS. Section 3 gives the details of the full implementation of forwarding skill of an agent within the simulated robotic soccer domain. Section 4 is devoted to discussions and concludes.

2 Architecture in Real-Time Decision Making in RTNAP MAS

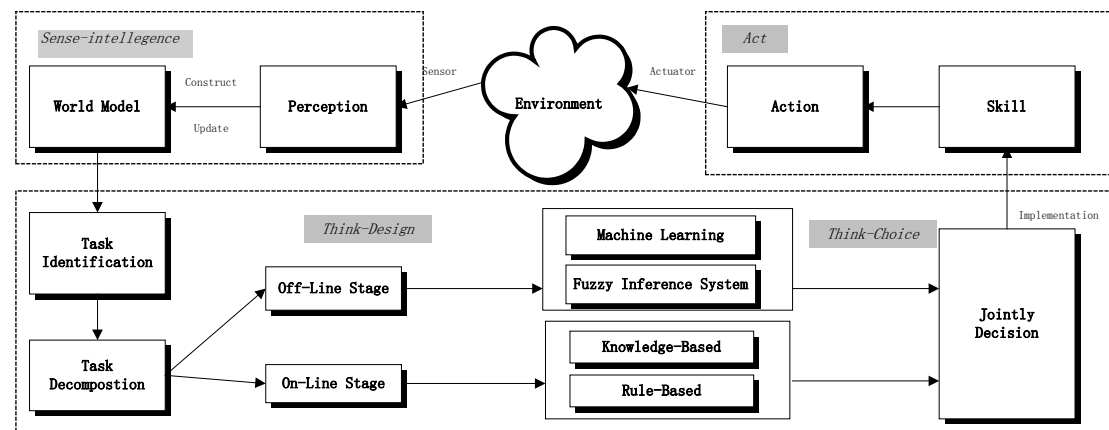


Fig. 2 Agent architecture and its relation to decision making in RTNAP domain

Flow among three decision phases is as follows:

- ◆ **In the intelligence phase, sensory data is interpreted as world model.**
Based on past and current perceptions of the environment, probabilistic representation of the world can be constructed and updated. It is obvious that, the more accurately world model represent the real world, the more effective decisions can be made.
- ◆ **In the design phase, goal is decomposed to hierarchical tasks, and then task is decomposed to those that can be learned or solved off-line and those that must realized on-line.**
Since the domain of RTNAP MAS is too complex for making decisions directly from an agent's sensory inputs to its actuator outputs, the long-term goal is decomposed to hierarchical sub goals, and each sub goal, called task, is divided into 2 stages, which differ in their time of implementation. Off-line stage is achieved by machine learning or fuzzy inference system, and this work can be frozen during actual task execution. However, on-line stage depends on knowledge-based or rule-based system to impose limitations on developed alternatives on-line.
- ◆ **In the choice phase, Local Optimization and Adversarial Consideration (LOAC) lead to the jointly decision ^[4].**
LO is the output of off-line stage method, which takes into consideration predicted world states as well as predicted effects of future actions in order to determine the optimal primitive action from a local perspective, both in time and in space ^[1]. On the other hand, AC is the output of on-line method, which takes into account nearby teammates and dangerous opponents from global perspective and limitations imposed by simulation environment such

as field area, and real-time strategy. Jointly decision is made via the combination of LOAC. Figure 2 shows the agent architecture and its relation to decision-making in RTNA domain.

3 RoboCup Simulation: A Case Study

RoboCup simulation is a suitable case where real-time decision-making can be examined, for the Soccer Server program provides a RTNAP MAS for that. The server calculates and updates the positions and movements of all objects on the field, sends audible and noisy, partial visual information to the players and receives commands from the players of both teams. Each player is an autonomous program that communicates with teammates and the server via a single and limited-bandwidth channel.

The world model of Everest 2002 agent is based on the confidence value to indicate the reliability of object in the field. Here emphasis is placed upon the thinking procedure of agent. In particular, the task of forwarding, a high-level ball-handle skill has been studied. Skills are sets of action commands to achieve desired short-term task, which are decomposed from the long-term goal of winning the game at all cost. Forwarding here is a kind of dribbling with fast speed. As is shown in Figure 3, this skill looks like that a player kicks the ball far ahead of him and then intercepts it as soon as possible. In implementing this skill, the agent whose basic position is on the wings often advances the ball with a fast speed cutting through the opponents' defense. So that it is an effective skill required by side forwarder, especially heterogeneous player with faster speed and greater acceleration.



Fig. 3 A process of forwarding

Forwarding is a challenging task for decision-making in RoboCup simulation for the following reasons.

◆ Real-Time

Each simulation cycle is only 100 msec, and agent has to make decisions and act in such a short period in response to a dynamically changing environment. If he can not react in a certain cycle, critical action opportunities may be missed and advantage conceded to the opponents.

◆ Noisy

In order to reflect unexpected movements of objects in real world, noise is added to the movement of objects and parameters of commands. And noise in the information renders the position and velocity of ball in the world model not accurate

◆ Adversarial

Opponents who have adversarial goals will impact the implementation of skill. In any cycle during the process of forwarding, the execution of skill maybe fail if the ball comes into opponent's kickable area.

◆ Pressing

In achieving a goal, the fewer cycle needed, the more effective it is. Forwarding is a skill to strive for winning time, so it is a pressing task, especially in the stage of preparation.

In this way, Efficiency and robustness become the 2 important factor on judging the goodness of the decision-making in forwarding.

The process of forwarding can be divided into 3 steps. First, the agent makes decision on the target position of forwarding ball. Second, he adjusts his position, body facing and neck angle to receive the ball easily and then kicks the ball out of his kickable area with adequate power. Third, he chases after the ball and catches it up as soon as possible in order to forward again. With good performance, a player can forward time and again to break through opponent's defense quickly.

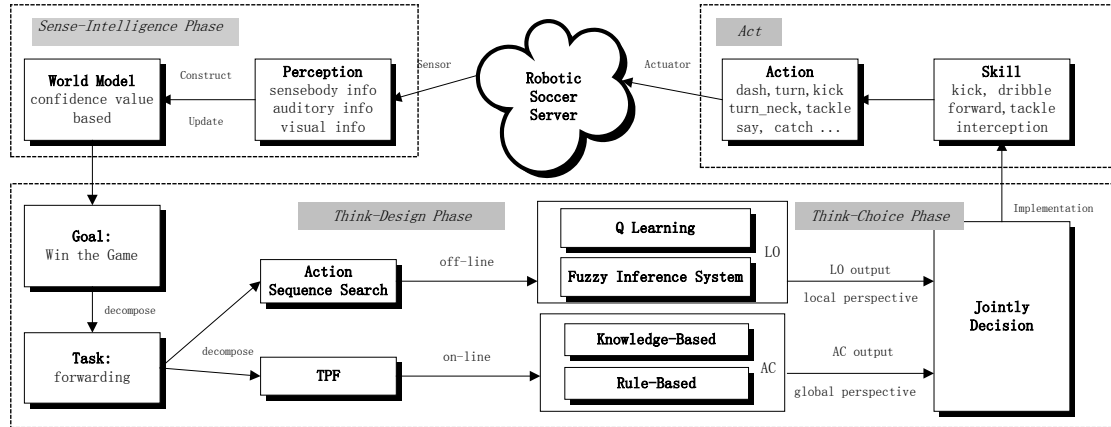


Fig. 4 Everest 2002 agent architecture in implementing the skill of forwarding

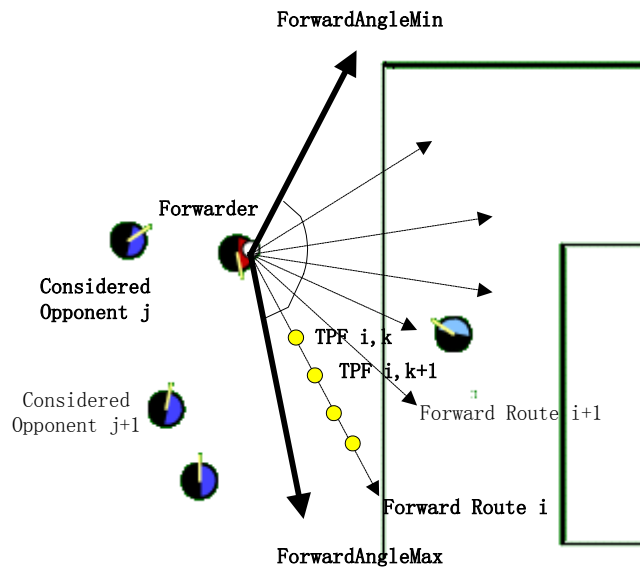


Fig. 5 A scene in decision making of TPF

The first step of making decision on Target Position of Forwarding (TPF) is based on soccer knowledge and AC. Shown as figure 5, agent has to decide on the forward angle and target ball position when catching it up next time. So TPF is determined by both the angle and the distance from the target position to himself. The basic idea of TPF is to balance between safety and velocity. That is, avoiding losing ball and moving forward as fast as possible, agent has to make it clear where to kick ball out and how much power needed in that kick. Discrete routes whose relative

angle to the agent's body facing are between *ForwardAngleMin* and *ForwardAngleMax* are taken into account. And each discrete point k on route i , denoted as $TPF_{i,k}$ are checked up by interception model assuming opponent j will try to intercept the ball before him. Eliminating the failed cases, in which ball will first enter the kickable area of considered opponent before him, agent evaluates the successful ones by knowledge-based system under that situation to get a global optimal TPF.

Q learning is used to locally optimize the next 2 steps. In the off-line learning of step 2, agent's final goal is to find a sequence of kicks and turns (the shorter, the better), such that eventually the ball leaves in a certain target direction with a certain target velocity that is pre-defined by TPF in the previous step^[5]. In most cases, at the time to kick the ball out, agent is already face the direction that the ball will leave at so as to save one cycle of turn. In step 3, chasing is a problem of interception; similarly, Q learning is used here to find an optimal sequence of dashes and turns. Figure 6 shows the sequence of kicks and turns to prepare forwarding, and the circle around the player means the kickable area.

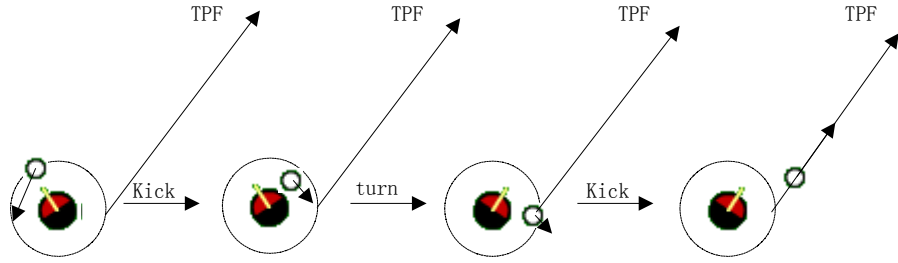


Fig. 6 A sequence of kicks and turns to prepare forwarding

4 Discussions and Conclusions

Team		Score	Average Cycles between forwarding and score	Number of effective forwarding
Chagamma	(Japan)	28-0*	14	13
Cyberoos	(Australia)	3-2*	18.7	3
UTUTD	(Iran)	2-0*	N/A	0
Virtual Werder	(Germany)	10-0*	17.1	7
United2002	(U. S. A)	9-0**	25	4
Helli-Respina	(Iran)	3-0**	12	3
WrightEagle	(China)	1-0**	N/A	0
Baltic Luebeck	(Germany)	11-0**	5	14
PolyteCS	(Iran)	15-0**	12	7
UvA Trilearn	(Netherlands)	3-2***	58	3
BrainStormers	(Germany)	2-1***	N/A	0
TsinghuAeolus	(China)	0-1***	N/A	0
BrainStormers	(Germany)	4-0***	19	2
TsinghuAeolus	(China)	0-7***	N/A	0
Total/Average		91-13	20.1	57

References

- [1] Simon, Herbert A. *The New Science of Decision Making*. Harper & Row, New York. 1960.
- [2] Efrim G. Mallach. *Decision Support and Data Warehouse Systems*. McGraw-Hill Inc. 2000.
- [3] Peter Stone, M. Veloso. *A Layered Approach to Learning Client Behaviors in the RoboCup Soccer Server*. Applied Artificial Intelligence. 12:165-188, 1998.
- [4] Gu Yang, Cui Lihui, Liu Junfeng, Wang Qinglin. *Local Optimization and Adversarial Consideration: Agent Skills in Everest 2002 RoboCup Simulation Team*. Submitted to Journal of Beijing Institute of Technology.
- [5] M. Riedmiller, A. Merke, D. Meier, A. Hoffmann, A. Sinner, O. Thate, Ch. Kill, and R. Ehrmann. *Karlsruhe Brainstormers: A Reinforcement Learning approach to robotic soccer*. In *Proceedings of RoboCup-2000: Robot Soccer World Cup IV*.