Hierarchical Group Decision-Making Model for Robot Soccer Based on Finite-State Transition

Shuliang Wang, Chen Ma, Bo Chen, Ming Zhong, Jigao Fu, Jiehao Chen, Pingfan He, Han Liang

Beijing Institute of Technology, School of Software, Beijing 100081, China
slwang2011@bit.edu.cn

Abstract - Aiming at the defect of centralized robot group control structure with large decision space dimension which causes heavy computation and communication burden, this page designs a hierarchical group decision-making model for robot soccer according to the multi-robot system hierarchical architecture and the finite-state of match states and robot roles. The mode significantly reduces MCU's computation burden and the communication burden, which can be used for the group-decision control in robot soccer match.

Index Terms - robot soccer; finite-state; hierarchical structure; group decision-making

1. Introduction

With the development of robotics, Multi-Robot System makes it possible to complete the task in parallel, improve working efficiency and expand capacity with the help of its features of spatial distribution, time distribution, resource distribution as well as information taxonomy. Resource sharing and function coordination help the system to make up individual robot's inadequacies. By taking advantage of the redundancy of information and knowledge, the system is also able to increase its fault tolerance, robustness and enhance the ability in complex environment. As a result, Multi-Robot System is receiving increasing attention by researchers.

The effective operation of Multi-robot System is heavily dependent on system organization and control, which requires us to organize the group of robots. How to form a robot group and achieve system's coordination and cooperation has currently become an important issue in the field of machine intelligence research, and it has important theoretical and practical significance.

Operation of the Multi-Robot Systems involves coordination and cooperation.

Coordination of Multi-Robot System refers to enforcing a coordinated constraint to the movement of all the robots within the system, with a major focus on system dynamics modeling, decomposition of movement and path planning.

The main emphasis of cooperation is multi-robot collaboration mechanism, that is, how to enable cooperation behaviors with given task, environment and a group of robots. The basic idea of cooperation are regarding the robots as a group so that we can observe the collaboration mechanism on the system perspective, giving full play to the intelligence of robots in the multi-robot system, in order to reorganize Multi-Robot System according to changes in environment and tasks as quick and flexible as possible.

The Group Architecture of Multi-Robot System refers to the topology relation among robots, including logical and physical Information Relations and Control Relations, as well as the distribution of problem solving ability, etc. The cooperative behaviors of Multi-Robot System is based on Groups Architecture of the system, which determines system's overall capacity. According to the control topology, Groups Architecture of the Multi-robot System can be divided into centralized, distributed and hierarchical architectures.

A MCU is often used by the Centralized Architecture to possess the information from the environment and the controlled robots, decompose and allocate tasks with programming algorithm and organize the robots to complete the task collectively. Generally speaking, Centralized Architecture possesses global information, which may help to generate the optimal solution after planning and coordinating. However, due to the oversize dimensions of the decision space, heavy computations and communication burden, the result of programming algorithm may cause poor real-time, adaptability and fault tolerance, which is hardly adapted to the complex changing circumstances.

Under the Distributed Architecture without MCU, the relationship among robots is relatively equal, and every robot can exchange information with each other and make its own decision. In that case, the robots enjoy a high degree of autonomy with better fault tolerance, stability and extendibility. But it's difficult to ensure global decision-making optimization.

The Hierarchical Architecture is a hybrid architecture between the Centralized Architecture and the Distributed Architecture. The Distributed Architecture is used in the coordination of Multi-Robot System while Hierarchical Centralized Architecture is used in the cooperation. We choose this solution as the architecture in the RoboCup.

RoboCup is an international robotics competition and academic activities, whose aim is to promote robotics and AI research. The RoboCup Middle Size League(MSL) is one of the RoboCup robot soccer leagues, in which human interference is excluded, while communication between the robots of a team or the robot and one remote computer system(herein after referred as Base Station) through wireless links is allowed. Therefore, it has been a major challenge to get robots well organized into teams to accomplish particular tasks, choose the forms of control and organization for Multi-robot System, accommodate the system to a changing circumstance, and ultimately realize a fully distributed robotic system of autonomous decision-making.

2. Hierarchical System Architecture

Based on the robots with abilities of visual image processing and Low-Level motion control, we adopt the Hierarchical Architecture of Multi-Robot System combined with and the Real-Time Information Maintenance Model, to design and implement a robot soccer group decision-making system based on the finite-state maintenance, realizing the Dynamic Role Assignment Strategy and Tactical Coordination in MSL.

The software system in MSL can be divided into two parts by different functions : the individual robot system and the Base Station system. The individual robot system, also known as subset decision-making system, is responsible for the coordination of individuals and the effective execution of competition function. The Base Station system, also called main decision-making system, is responsible for the coordination of Multi-Robot system.

A. Coordination Mechanism for Individual Robot

The independent decision-making process depends on the implementation of individuals' coordination mechanism, which organizes the coordination and execution of control commands in units of "behavior".

Individual coordination mechanism involves the composition and resolution of motion and the path planning, which are outside the range of this paper and there is no need to repeat them.

B. Coordination Mechanism for Multi-Robot System

Group coordination mechanism is suitable for the phases before organizing robots to place the place-kick ball, take a place kick, or being attacked by the opponents of the team with ball. These phases require much collaborative process of robots and the centralized control by the Base Station system.

Individual coordination mechanism is suitable for the phases during the opponents of the team's path planing, attacks with ball, automatic-obstacle-avoidance and shotting. Those phases just require that the robot players make decisions themselves by combing sharded information and the world model built by group vision according to their physical architecture and the sensor data they capture.

Group coordination mechanism is an important embodiment in the process of group decision-making, which is based on group strategy, roles, role effect, role definition and behavior commands.

Group Strategy for Robot Soccer is a complete implementation process of arranging robots to accomplish given tasks, which is triggered by finite states of the match.

Group Strategy implements one strategy or tactics, with the help of the processes of tactical deploy with abstract role strategy, correction of rule confliction, feasibility study of individual robot role, the strategic role distribution and positioning, and transmission of behavior commands.

Role strategy is an abstract concept in the tactical layout and it is also the fundament of group strategy. During the process of group strategy, roles take part in the aspects of generation, effectiveness of inspection and assignment. Role Generation refers to an ideal tactical layout according the compete condition. Role's Effectiveness Of Inspection refers to inspecting the role of each robot on the basis of its physical information. Role Assignment refers to assigning a role of strategy to individual robot using priority table in sequence.

3. Finite-State Model for MSL

Each match of MSL is controlled by the commands from Referee Box Program. In order to ensure the ordered execution of match complying with competition rules, the team Base Station receives commands from the Referee Box and send them to robots. Therefore, the implementation of detailed model during the match is an important prerequisite for group decision-making and cooperation.

A. Finite-State Model During The Match

The robot players are coached by the commands from their team's Base Station System, we can divide the match time into a number of states by the control commands. According to those commands, the Base Station update robot players' status to meet the changes and rules of competition.

As is shown in Figure 1, we divide the match phases and simplify the match states to observe their changes.

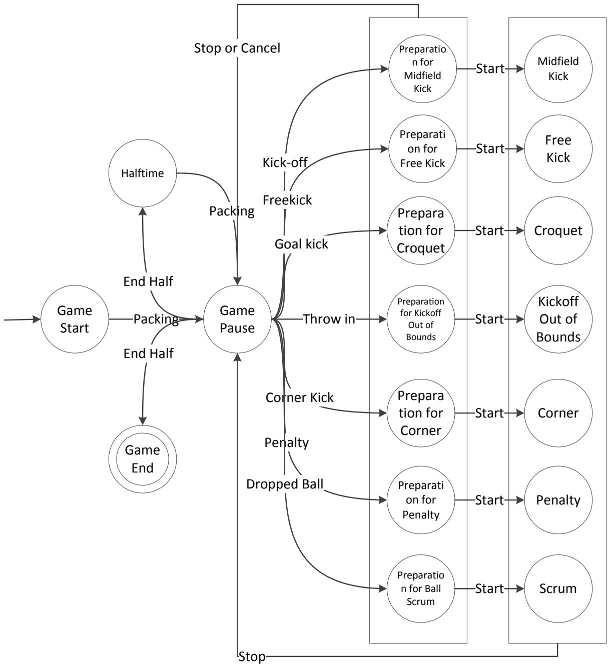

Fig. 1.State Transition Diagram in The RoboCup MSL

Match states can be described as a simple Finite-State Model using a formalized approach.

Definition 1: The Finite-State Machine during the MSL is a 5 tuple.

$$M = (S , \Sigma , f , S_0 , Z) \tag{1}$$

S: The finite set of match states $S_i (S_i \subset S)$, each match stateis an element of the set.

Σ: The finite set of control commands sent by Referee Box. Each complete control command is an element of the set.

f: The match state-transition function, which is a partial mapping on sets.

S_0: The start state of automaton-- i.e. the start state of the match, $S_0 \subset S$.

Z: The final state of automaton, $Z \subset S$.

Formalized approach helps us to inspect the system properties during the state transition. Aiming at the Finite-State Machine of match, the formalized approach based on Linear Temporal Logic(LTL) captures the Safety and Liveness properties during the state transition, which can deduce the information that FSM can't reach.

B. *Finite-State Model for Robot Behaviors*

As is shown in Figure 2, the states of robot behaviors is more complicated comparing to the states during the match. More possibilities of states transition cause the complication. According to the unpredictable specific situations, the system allows robots to covert between any two behavior states. Besides, the concurrency of robots' behavior states also expand the state space.

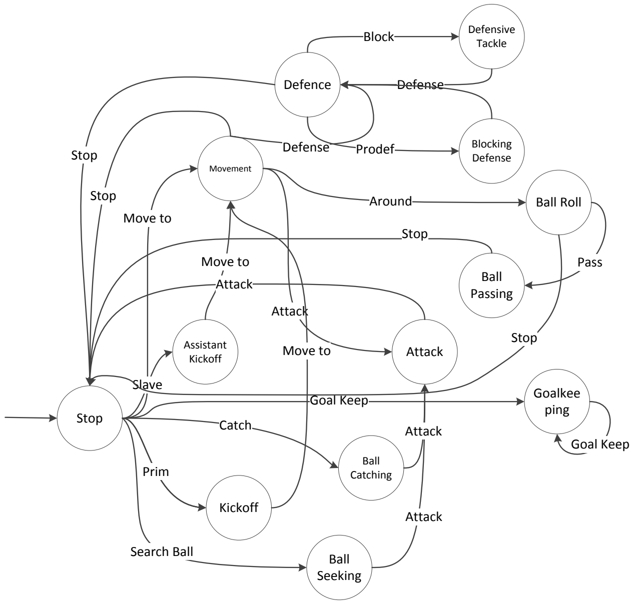

Fig. 2. The Transition Diagram of Robot Behavior States in The MSL

According to the actual game, there is not need to inspect all of the transitions among behavior states but just pay attention for a period of time, especially the transition from group decision-making state to individual decision-making state. As a result, robot behaviors modeling approach differs from the way that pays more attention to the match state of the substate space. At the same time, the modeling of the behavior of the robot will also introduce concurrency problems.

Definition 2: The Petri net of Multi-Robot Parallel System is a 4 tuple.

$$N = (P，T，F，U) \qquad (2)$$

P: Set of place, a prioritized sets of robot behavior states, where each behaviour state is an element, $P_i\,(P_i \subset P)$.

T: Set of transitions, a transport conditions set of robot behavior states, representing the atomic actions during the states transition when conditions for transition of robot behavior states are satisfied.

F: Set of input functions, a subset of $P \times T$, representing the mapping relationship between existing re-state and migration.

U: Set of output functions, a subset of $T \times P$, representing the mapping relationship between existing migration and following state.

With the Petri net model for Multi-Robot Parallel System, each player is regarded as a single token passed around the net. In the Petri diagram, the migration is allowed if all of its input places have tokens. One migration happens once it is allowed, after that the tokens from input places are consumed and new tokens are provided for output places. In the Collateral Multi-Robot System, one migration means that the concurrent behaviors will happen and the reasonable migration between Multi-Robot System state and Individual state can be implemented when the conditions of number and states of robots with concurrent behaviors are met.

4 Model Validation

This paper validates the model with the cooperation processes during the kick-off of place-kick as example, including passing and catching soccer statically, defensive counterattacking, and transforming from attack to defence.

A. *Passing and Catching Soccer Statically*

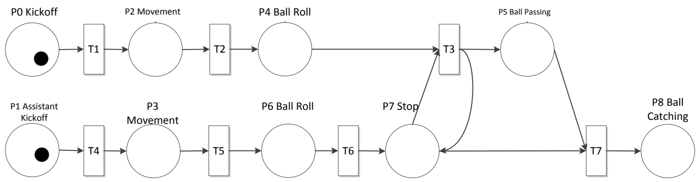

Fig.3. Petri-Net Model of Passing and Catching Cooperation

The analysis result of robot behavior states is shown in Figure 3.

The process of Passing and Catching soccer requires the participation of two robots, so there are two initial libraries and it requires the two representative robots' tokens to get involve. The intercurrent procedure of Passing and Catching soccer will perform automatically.

The main bullier turns from kick-off state into moving state, moving at a special angle and speed to the area in front of the soccer , then goes around the ball in order to adjust the passing direction and waits for the assistant bullier.

The assistant bullier turns from assistant state into moving state, moving at a special angle and speed to the area behind the soccer, then goes around the ball in order to adjust the catching direction and assigns the main bullier in adjusting direction until it is completed.

The main bullier passes soccer to the assistant bullier and turn into catching state. After that the assistant bullier turns into passing state and passes the soccer ball to the auxiliary robot, who immediately turns into catching state and catches the ball.

B. *Defensive Counterattacking*

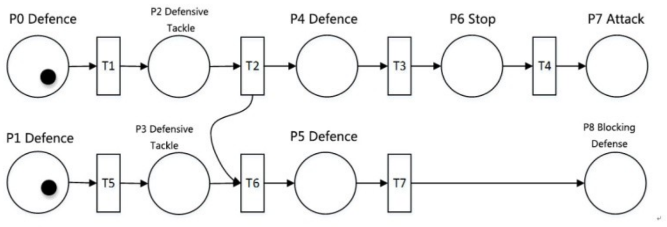

Fig.4. Petri-Net Model of Defensive Counterattacking Cooperation

The analysis result of robot behavior states is shown in Figure 4.

The tactics of Defensive Counterattacking requires the participation of two robots, so there are two initial libraries and it requires the two representative robots' tokens to get involve. The intercurrent procedure of Defensive Counterattacking will perform automatically after the success of an interception by a robot.

When the opponent robots holding ball mount an attack to us, two defenders would turn from defensive state into defensive counterattacking state, then beat the attackers with a defensive double team and try to steal and intercept them.

After a successful steal, the robot who is holding the soccer ball turns into terminal state while the another one turns back to defensive state.

Then the robot who holds the ball turns into attacking state and begin to mount an attack.

C. *Transforming from Attack to Defence*

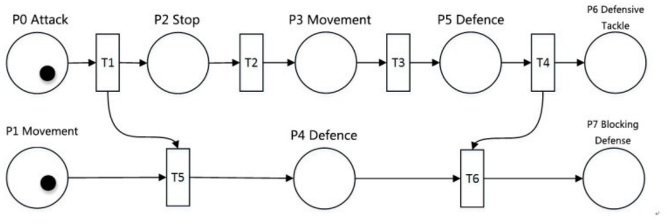

Fig.5. Petri-Net Model of Transforming from Attack to Defence

The analysis result of robot behavior states is shown in Figure 5

The process of Transforming From Attack To Defence requires the participation of two robots, so there are two initial libraries and it requires the two representative robots' tokens to get involved. The intercurrent procedure of Passing and Catching soccer will perform automatically after the two robots are divided into offensive player and moving player.

The offensive player turns into terminal state when its soccer ball is intercepted by opponents, and the moving player will be informed to turn into defensive state at the same time.

After laying the ball off, the offensive player turns into moving state, and again into defensive state to defend the opponents, then turns into intercepting defensive state, informing the moving player to turn into plugging defensive state at the same time.

References

[1] ZhengAqi. Qt4 Development Practices [M]. Beijing：Publishing House of Electronics Industry, 2011.

[2] XieXiRen. Computer Network [M]. Beijing, Publishing House of Electronics Industry, 2008.

[3] Han ChongZhao. Multi-source Information Fusion [M]· Beijing：Tsinghua University Press, 2006.

[4] Cui Lianhu. Research on RoboCup mid-sized robot cooperation [D]. Hunan：National University of Defense Technology, 2007.

[5] Fan Yong. Research on Multi-robot Collaboration and Control [D]. Beijing：the Institute of Automation of the ChineseAcademy of Sciences, 2000.

[6] Ben YongZeng. Study on Intelligent Mobile Robot and Robot Football Match Strategy [D]. Beijing：Beijing University Of Technology.

[7] Yang Fan. RoboCup Simulation Orientation and Online Adjustment Based on Scene and Strategic [J]. Control Theory and Application, 2004(21)：10-14.

[8] ZhongBiliang. Research on Robot Soccer Role Transformation Based on Petri-net [J]. Computer Engineering and Applications, 2003(24)：60-63.

[9] Zhao XiBin. RoboCup Strategy Analysis with Dynamics Goal-Driven [J]. Computer Engineering, 2003(6)：8-9.

[10] Du TaoJun, Huang Hong. Multi-agent Collaborative Planning Theory and Application in RoboCup [J]. Computer Simulation, 2004(7)：125-128.

[11] Nuno Lau. Multi-Robot Team Coordination through Roles, Positionings and Coordinated Procedure[R]. St.Louis, USA：IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.

[12] Mark Sims. Automated Organization Design for Multi-agent Systems [J]. Auton Agent Multi-Agent Syst, 2008(16)：151-185.

[13] John Atkinson. On-the-fly Generation of Multi-robot Team Formation Strategies Based on Game Conditions [J]. Expert Systems with Applications, 2009(36)：6082-6090.

[14] Joao Silva. World modeling on an MSL Robotic Soccer Team [J]. Mechatronics, 2011(21)：411-422.

[15] 2011 China Robot Competition &RoboCup China Open Mid-sized Robot Rules. RoboCup in China mid-sized Technology Committee, 2011