Human-Robot Interaction: Group Behavior Level

Lev Stankevich, Denis Trotsky
Saint Petersburg State Technical University
29, Politechnicheskaya, St. Petersburg, 195251, Russia
den_tr@mail.ru

Abstract
This paper describes a new approach to problem of human-robot interaction for controlling behavior of robots group. A solving this problem is actual for providing teamwork of robots and other unmanned vehicles controlled in command and control manner. At group level of interaction, a human operator can define strategy and tactics of robot teamwork. It is proposed to use special multi-agent 3D game simulation environment (Basketball Server), which provides teamwork of basketball agents under control of the operator, in order to study principles of interaction for human operator and robot team. Using the environment, the operator can change strategy and tactics of team operatively using special agent and modules embedded into agent’s program, for interaction with a basketball agent. The environment has 3D visualization of game. This allows for defining effectiveness of individual and collective behaviors of the agents and their ability to solve complex tasks of attack and defense in basketball environment. Examples of realization and investigation of different tactics for teamwork are considered.

1. Introduction
At present there is large interest to problem of control for autonomous vehicles performing common works in group. These can be mobile robots, participating in rescue and military operations, or unmanned vehicles performing various tasks in air, water and space. These can be also software robots that are agents solving industrial, economical, social tasks cooperatively, or agents playing in soccer or basketball and participating in rescue team competitions.

Interaction of human operator and robots working in group is possible at individual (with every single humanoid robot) and group levels. Currently interaction of human and humanoid robot is most interested. The humanoid robots are mobile robots of new class having human-based form and behaviour. These must be able to act in dynamic and not structural environment and to interact with human, which not having skills of control robots. That’s why creation of friendly human-robot interface is critical for effective using such robots.

There are attempts of creating human-robot interface of individual level based on specialized audio- and video-means. In common case, such means must allow for detecting human on a scene, segment human’s face or hands, to understand facial gestures or lip movement patterns corresponding of pronounce of words, determine meaning of pose and gestures of man, and know voice and speaking message of a person. However an effective solution of these tasks is very difficult problem.

At the group level the operator can define (assign) strategy (type of group behavior and role of robots) and tactics (way and step sequence of behavior realization), as well to evaluate quality their realization by watching. At individual level there is interaction of human operator and single robot as a separate act in which the operator forms a message consisting of instructions (commands) for robot and checks their executions using channel of watching or message from the robot. At the group level of interaction especial interest presents a problem of interaction for operator and robot group, performing teamwork, requiring carry out common intentions of all team members to achieve common goal.

In this work it’s supposed to use specialized virtual game environment for studying general principals of such interaction. This environment may be as universal mean to model and to investigate interaction of operator and robot group in what the robots have complex collective and individual behavior. The 3D environment for virtual basketball developed provides possibilities to operatively change strategy and tactics of robotic agent teamwork. Visualization of the game on monitor of computer allows for controlling behavior agents of team by the operator. Usage of competition environment provides investigation of different strategy and tactics of teamwork and choosing best of them. Such type of environment can be used not only to investigate, but also to model control for team of really robots or autonomous vehicles in central office of control. It’s important for effective performing rescue, transport, reconnaissance and military tasks in command & control systems.

Remainder of the paper has following content. In the second section, principals of human-robot interaction for group robot control system organization are discussed. In the third section of this paper the 3D Basketball Server is described. In the fourth section it’s shown how the basketball agents can be built. In the fifth section experiments related to change tactics and strategy are presented.

The work is based on author private experience on design of special soccer and basketball agents for game application [4] and intelligent control systems of robots [5, 6, 7].

2. Principals of group interaction human-robot
In command & control systems operator must have possibility to input instruction for changing strategy and tactics teamwork of robots. In this work it is proposed to do it through terminal having specialized program agent-supervisor connected to robots of group. In order to model such system can be used multi-agent approach [3]. It is allowed for developing multi-agent systems that can be viewed as distributed control...
systems based on intelligent agents, which interacts with each other. Currently the multi-agent technology is often used for creating control system of intellectual robots [5, 6, 7].

In our case a special virtual game environment is used. The environment consists of the server and group of intelligence agents modeling robots. Such environment is multi-agent environment including system of human-robot interaction.

Structure of human-robot interaction system at group behavior level is presented in fig. 1. In our system the agents model the robots and they are realized as the autonomous control systems with added coordination functions. Every agent realizes a set of functions, which provide individual and collective robot’s behaviors. The agents are built on the basis of the layered reactive architecture. This architecture allows for doing agent’s behavior similar to human one. The more complex agents can be built on the basis of BDI architecture [1]. Interaction between the agents is determined by its upper level of collective behavior. The special functions that realize the coordination mechanisms must be used for agent’s cooperation. They provide ability of the agents to be cooperated and/or competed.

In our case it’s supposed using 3D game environments of soccer and basketball for modeling group interaction operator-robot. For example, in [2] a variant of basketball simulation environment called RoboNBA (National Basketball Association) was discussed. Unfortunately, this variant use 2D visualization. However in our case, more appropriate is 3D variant. Therefore a development of full 3D basketball simulation is more relevant for our case. Namely such variant of Basketball environment was developed to use as modeling one in the interaction system.

3. **Structure and components of Basketball Server**

Basketball Server was developed to be the heart of the environment for competition of agent teams. The server was developed in Delphi 7. The server is compatible with OS Windows XP, and also with other OS of Windows family. It requires Hardware with not less than 64MB RAM, 1 GHz CPU speed, 20 Mb free disk space.

3.1. **The structure of the server**

The server includes Communication, Logical, and Graphic modules.

The Communication module provides connection with clients, data transfer over TCP/IP protocol, and interaction with the logic module.

The Logical module realizes mathematical model of the environment. The basic stages of functioning of the logical module:
- Change of a status of agents in the environment;
- Processing of a simulation step of the environment according to a new status of agents;
- Preparation of the sensor information for agents;

The Graphical module visualizes objects in the environment.

The schema of interaction of modules is presented in fig. 2.

3.2. **Model of the environment**

The environment consists of the court, two basket rings with backboard, ball and 10-players. The size of the court is $x_{max}$ by $y_{max}$ (fig. 3).

$$P = \{ p_i \} ,$$

where $p_i = (x_i, y_i, z_i)$.

The position of the ball is also defined in 3D space:
Ball = \{x, y, z\} \tag{3}

The ball may have status FREE (the ball is free) and BUSY (the ball is taken by player). The player is identified by the number of its team as TeamID = \{0,1\} and by player number in the team - PlayerID = \{1,2,3,4,5\}.

3.3. Action model

In the server, the time is update in discrete steps. A simulation step is 100 ms.

The server can process the limited number of actions that defined as commands sent by a player (one command of each step is 100 ms.

- SHOOT (power Pow, direction DirXY and DirZ). The player shoots the ball with the power Pow, in direction of horizontal plane DirXY and in direction of vertical plane DirZ.
- PASS (power Pow, direction DirXY and DirZ). The player passes the ball with power Pow, in direction of horizontal planes DirXY and in direction of vertical plane DirZ. The ball, moving with the power Pow in direction DirXY and DirZ, is switched in state FREE.
- RUN (power Pow). The player runs with power Pow in current direction.
- TURN DIRECTION (direction DirXY). The player changes its body direction to Dir XY.
- CATCH. The player captures the ball. If distance between the ball and the player is less than CatchDist, the ball belongs to the player. If more then one player is within distance CatchDist to the ball, the ball will go to the nearest player. Catch action is executed only when the ball is free.
- BLOCKSHOOT. Player blocks shoot or pass of opponent.

3.4. Model of movement

Position of the player with coordinates \( \{x, y, z\} \), power \( pow \), and a direction \( DirXY \) in the next simulation step is calculated as follows:

\[
\begin{align*}
x_2 &= x_1 + pow \cdot \cos(dirXY \cdot Pi/180) \\
y_2 &= y_1 + pow \cdot \sin(dirXY \cdot Pi/180) \\
z_2 &= z_1
\end{align*}
\]

where \( \{x_2, y_2, z_2\} \) is the new coordinates of the player.

Position of the ball with coordinates \( \{x, y, z\} \), power \( pow \), direction \( DirXY \) and \( DirZ \) in the next simulation step is calculated in the following way:

\[
\begin{align*}
x_2 &= x_1 + pow \cdot \cos(-dirZ \cdot Pi/180) \cdot \cos(dirXY \cdot Pi/180) \\
y_2 &= y_1 + pow \cdot \cos(-dirZ \cdot Pi/180) \cdot \sin(dirXY \cdot Pi/180) \\
z_2 &= z_1 + zspeed
\end{align*}
\]

where \( \{x_2, y_2, z_2\} \) is the new coordinates of the ball, \( GRAVITY \) is the acceleration of free falling, \( zspeed \) is the vertical speed of the ball, calculated in the moment of shoot or pass by the formula (6).

\[zspeed = pow \cdot \sin(dirZ \cdot Pi/180)\] \tag{6}

3.5. Sensor model

The server can send the following information to players:

- Own coordinate;
- Coordinates, TeamID, PlayerID, DirXY parameters of all partners and opponents;
- Coordinates and status of the ball, TeamID and PlayerID of player, who controls the ball, if status of the ball is BUSY.

3.6. Structure and components of the program of the server

The server is implemented as object-oriented program. The program of the server consists of the following modules:

- Main.pas is the Communication module, which performs the start of the TCP/IP server, organizes connection and data transfer to clients;
- Environment.pas is the Logic module, which realizes mathematical model of the environment, simulates of actions of agents and movements of a ball;
- DGraph.pas is the Graphic module, which contains functions for graphic display of the environment;
- Basket.pas is the file of the description of protocol and commands of the server;
- Constants.pas is file containing constants of the server;

4. The basketball agent

To provide complex individual and collective behaviors at teamwork, the basketball agents must be built using multi-layered architectures. The basketball agent has three-level architecture that is similar to the soccer agent architecture used in [4]. Note, that one of the variants of the agent with described architecture was used for creation of the soccer agent of team STEP (Soccer Team of ElectroPult) that has became by winner of World Championship RoboCup-2004 in Simulation 2D Soccer League.

The low level of the basketball agent has several executive reactive layers. Number of the layers can be changed at agent’s behavior tuning. Each executive layer has its priority and reacts to given situation by forming the corresponding actions in response on input information. Set of such reactions defines some primitive executive behaviors (agent’s skills). Sequence of the selected reactions corresponds to current intention of the agent.

The middle level of the agent has set of production rules defining individual behavior of the agent. These rules use the conditions in form of data, facts, and situations that are known for the agent. They carry out decisions for selection of primitive behaviors that must be realized at low level of the agent.

The upper level of the agent also has set of production rules that form corresponding collective agent’s behavior. At this level the agent first makes decision on selection of
whether individual or collective behavior. If the agent selects a collective behavior, then it must take into account positions of partners and opponents. In case of arising conflicts the current collective behaviors of agents are formed. The conflict is solved using special rules for conforming agent’s actions.

Agent has module of interaction with operator. This module receives commands of operator through the Server and according to one change tactics and strategy of behavior of agent. For example, operator can point out agents that it’s long throwing, if opponent mark agents not dense on 3 point line. In this case, module of interaction change set of production rule of upper level.

4.1. Structure of the agent

The structure of the agent is presented in fig. 4.

![Diagram of agent structure](image)

4.2. Tactics of the agent

At the second level a player could make decision on attack, defense, or catch the ball. The goal of the team in the attack is to score a ball in a basket ring. The goal of the team in the defense is to not allow the opponents to finish the attack. There are some variants of the organization of the defense. In the given example a personal marking of players is used. During the defense, players have the following tasks:

- To block the free moving opponent to the basket;
- To intercept the ball while opponent passes;
- To deliver the ball to opponent basket;
- To perform an accuracy shoot;

Catching ball actions should take into account first of all a status of the ball and position of the player related to the ball. At the team tactic level the player makes a decision on current action in the team. Formally algorithm of decision making is described as (7):

\[
((\text{BALL}=\text{FREE}) \land (t_0 \leq t_1) \land (t_0 \leq t_1))
\Rightarrow (\text{Go to get the ball})
\]

\[
\text{else } (t_1 \leq t_2) \Rightarrow (\text{Go to the attack half})
\]

\[
\text{else } (\text{Go back to the defense half})
\]

where \( I_0 \) is distance between the player and the ball, \( I_1 \) is distance between the ball and partner nearest to the ball, \( I_2 \) is distance between the ball and opponent nearest to ball.

The given algorithm is the same as described in [2]. If the ball is busy, the player has two options (8):

\[
((\text{BALL}=\text{BUSY}) \land
\text{(BALLTeamId}=	ext{PlayerTeamId}))
\Rightarrow (\text{Go to get the ball})
\]

\[
((\text{BALL}=\text{BUSY})
\land (\text{BALLTeamId} \neq \text{PlayerTeamId}))
\Rightarrow (\text{Go to the attack half})
\]

where \( \text{BALLTeamId} \) is number of team, who controls the ball, \( \text{PlayerTeamId} \) is number of player in the team.

**Attack actions.** Goal of the team in attack is to score the ball in the basket ring. First it is necessary to deliver the ball up to the basket of opponent. The ball can move on the court using two of the ways:

1) Player runs with ball;
2) Player passes the ball to partner;

Two selection strategies can be used such as selection of partner nearest to opponent basket ring and selection of partner, whose position is optimal. The algorithms implementing these strategies are described in detail in [2]. Optimality of the position of each of partners on the court is described by some value. This value is named an evaluation of the player. When a player needs to pass the ball, he selects teammate with the highest evaluation.

Evaluations of all partners are defined in following way:

\[
t_1 = f(d_{1}^n) + \sum_{m} g(d_{2m}^n)
\]

where \( d_{1}^n \) is the distance between \( n \)th teammate and basket; \( d_{2m}^n \) is the distance between \( n \)th teammate and \( m \)th opponent; \( f(x) \) is a function that evaluates the goodness to shoot for the teammate; \( g(x) \) is a function that evaluates the threat from opponents.

When partner is defined, ball’s trajectory is calculated. Initial parameters of a pass are calculated such a way that the ball has not been intercepted by opponent and the given partner will be the first player who can intercept the ball.

1) The horizontal direction of the ball is defined as:

\[
\text{DirXY} = \frac{\text{ArcCos}((\text{Partner}.x - \text{Agent}.x))}{\text{Dist}((\text{Partner}, \text{Agent}))*180 / \pi}
\]

if \((\text{Partner}.y < \text{Agent}.y)\)

then \((\text{DirXY} = -\text{DirXY})\)

where \( \text{DirXY} \) is horizontal direction of pass; \( \text{Partner} \) is the coordinate of the partner who receives the pass; \( \text{Agent} \) is the
coordinate of the partner which passes; \( \text{Dist} \) is function defining distance between two points on a court.

2) Definition of ranges of change of initial parameters of a pass is made as follows:

\[
\begin{align*}
\text{Pow} &= 1..30 \\
\text{DiZ} &= 0..75 \\
\text{Time} &= 0..50
\end{align*}
\]

where \( \text{Pow} \) is the power of pass, \( \text{DiZ} \) is the vertical speed of pass, \( \text{Time} \) is count of time steps.

3) Initial ball’s parameters are calculated as:

\[
\begin{align*}
\text{Ball}.x &= \text{Agent}.x \\
\text{Ball}.y &= \text{Agent}.y \\
\text{Ball}.z &= \text{PlayerHeight} \\
V_z &= \text{Pow} \cdot \sin(\text{DiZ} \cdot \text{Pi} / 180)
\end{align*}
\]

where \( \text{Ball} \) is the coordinate of the ball; \( \text{PlayerHeight} \) is the constant of the server defining height of the player; \( V_z \) is vertical speed of the ball.

4) The ball’s parameters are calculated as follows with time:

\[
\begin{align*}
V_z &= V_z - G \\
\text{Ball}.x &= \text{Ball}.x + \text{Pow} \cdot \cos(\frac{\text{DirZ} \cdot \text{Pi}}{180}) \cdot \cos(\frac{\text{DirXY} \cdot \text{Pi}}{180}) \\
\text{Ball}.y &= \text{Ball}.y + \text{Pow} \cdot \cos(\frac{\text{DirZ} \cdot \text{Pi}}{180}) \cdot \sin(\frac{\text{DirXY} \cdot \text{Pi}}{180}) \\
\text{Ball}.z &= \text{Ball}.z + V_z
\end{align*}
\]

where \( G \) is acceleration of free falling.

5) A hit of the ball in the given zone is defined using the following condition:

\[
\text{Dist}(\text{Ball}, \text{Partner}) < \text{CatchableDist}
\]

Coordinates of a ball and initial parameters of a pass are saved as:

\[
\begin{align*}
\text{BallFinal} &= \text{Ball} \\
\text{MinDirZ} &= \text{DirZ} \\
\text{MinPow} &= \text{Pow}
\end{align*}
\]

6) Search of the nearest \( \text{BallFinal} \) to the partner is realized by the following way:

\[
\begin{align*}
\text{MinDirZ}, \text{MinPow} &\xrightarrow{\Delta} 0 \\
\text{where} \ \Delta &= \text{Min}(\text{BallFinal} - \text{Partner})
\end{align*}
\]

Parameters \( \text{MinDirZ}, \text{MinPow} \) are selected, at which distance between the ball and partner is the least one.

Graphic interpretation of the given algorithm is presented in fig. 5. The given algorithm builds several trajectories of a throw (curves 1, 2, 3 in fig. 5) and chooses one of them, which is closer to the partner. In case shown in fig 5, such curve is the trajectory 3. This algorithm also calculates parameters of shoot to the basket ring.

![Fig. 5. Minimization of pass parameters](image)

**Defense actions.** As defensive strategy, the personal marking of players is used. The player \( i \) chooses of the opponent \( j \) for marking if the condition (17) is satisfied.

\[
\text{PlayerID}_i = \text{PlayerID}_j
\]

Further it is necessary to choose a position on a field, to which the player should move to mark the opponent. Coordinates of this position are calculated by the formula (18):

\[
P_i = (P_j + (P_j + \text{CircleOwn} / 2)) / 2
\]

where \( P_j \) is coordinate of the \( j \)th opponent; \( \text{CircleOwn} \) is coordinate of the own basket ring; \( P_i \) is required position.

At the moment of a pass or a throw when the ball is switched in a status FREE, the marking player can make interception of the ball. If it was possible, then the player is switched to attack, else player is switched to defense.

## 5. Experiments on strategy and tactics investigation

Experiment was directed to investigation of teamwork effectiveness using handling team strategy and tactics by operator in real time. Attack and defense strategies were examined. And tactics of pass and dribble were also examined. For example, the two functions of selection of a partner for a pass were considered (see section 3.4):

1) Function of nearest partner (selection of partner nearest to opponent basket ring);

2) Function of optimal position (selection of partner, whose position is optimal).

Efficiency of each system of a pass was estimated through tactical and technical parameters of each team. The tactical and technical parameters include: the score, 2 points throws (success/all), 3 points throws (success/all), pass (success/all), accuracy of a pass (%), possession of a ball (%).

To investigate efficiency of each system was made match of two teams, each of which uses one of functions of a pass. Team A used the function of optimal position, and the Team B 2 used the function of nearest partner. In fig. 6 the fragment of game is shown.
In the table the tactical and technical parameters of experiment are presented.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Team A</th>
<th>Team B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>2 points throws (suc. / all)</td>
<td>2 / 8</td>
<td>1 / 2</td>
</tr>
<tr>
<td>3 points throws (suc. / all)</td>
<td>2 / 7</td>
<td>1 / 1</td>
</tr>
<tr>
<td>Pass (suc. / all)</td>
<td>33 / 64</td>
<td>20 / 65</td>
</tr>
<tr>
<td>Accuracy of a pass (%)</td>
<td>52</td>
<td>31</td>
</tr>
<tr>
<td>Possession of a ball</td>
<td>52</td>
<td>48</td>
</tr>
</tbody>
</table>

Several evaluations can be concluded from this table.

**Number of the passes and accuracy of the pass:** the higher parameters in accuracy of a pass for the Team A are caused by using the function of optimal position; this function first of all prefers those players, who mark of the opponent less, and hence probability that the ball will be intercepted will be less.

**Possession of the ball:** the higher parameter in possession of the ball for the Team A is consequence of the superiority in accuracy of a pass. The team better plays a pass because it more holds the ball.

**The score and throws:** Team A has demonstrated better game in a pass; therefore it has advantage in the score as well as throws. The Team B only 3 times has finished the attack with a throw on a basket ring while the Team A has executed 15 throws. Low accuracy of a throw of the Team A has not allowed having considerable advantage in the score. The given algorithm of throws was not found effective since in the result of experiment the low parameter of accuracy of a throw was obtained.

### 7. References


