Mathematical and Computer Modelling of Dynamical Systems, 2003, vol. 9, no. 2, pp. 137-150.

*

Robot Soccer Collision Modelling and Validation in Multi-Agent Simulator

G. KLANČAR*, M. LEPETIČ*, R. KARBA* AND B. ZUPANČIČ*

SUMMARY

The paper deals with mathematical modelling and simulation of collisions in robot soccer representing ideal playground for studying multi-agent mobile systems. It involves robot and ball dynamic behaviour and focuses mainly on their collisions study and their realization. Some vital parts of the simulator are explained and modelled in more detail, beginning with the simple model of ball and robot motion and continuing with more complex collision models. Special consideration is given to collision between robots. The design of such model takes two steps. In the first, information about possible collision is obtained. The second step realizes collision by determining appropriate force impulse. The results from model verification are presented. It is shown that the developed model represents a good basis for realistic, yet simple enough, collision simulation. The paper concludes with some remarks and ideas for future work.

Keywords: simulator, multi-agent system, collision detection, modelling, discontinuous simulation.

1. INTRODUCTION

In the last two decades the concept of multi-agent mobile systems has been observed in many computer simulations, laboratory examples and in some practical applications. Inspiration for the design of such systems could be found in nature, as for example: incredible group organization of ants, bees, group of hunting predators etc. Here the importance of organization, work and information sharing as well as communication can easily be identified.

Researchers try to realize at least piece of this idea by multi-agent robot systems applications [12]. Among them robot soccer is very popular and serves as a perfect example of multi-agent systems in the last few years [4] [11]. It gives the possibility to study multi-agent related topics [12] such as: robot soccer, group formations, robots pushing objects, study of social science aspects, study of cooperation paradigm, learning methods and algorithms as well as mechanisms for adaptations and behaviour

*Address correspondance to: G. Klančar, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia.

assignment in multi-agent systems, opponent plan or strategy identification, reactive and cognitive capabilities of agents behaviour. However, also the possibility to study the capabilities of a single agent is enabled: optimal path planning and following algorithms, static and dynamic obstacle avoidance methods and prediction algorithm study (to predict proper position of other agents and the ball taking into account capabilities of the agent and to predict plans of the opponents). The reason for robot soccer popularity originates in the fact that it efficiently combines many research interests [14] besides the already mentioned it also involves multi-agent cooperation, game strategy, real-time data and image processing, robotic vision, artificial intelligence and control. The area has also proven to be very usable in engineering education not only because of the reasons stated above but also because of its attractiveness [10].

The paper presents a methodology to model the collision between soccer robots which is an important part of a multi-agent simulator for robot soccer game. It is implemented in Matlab Simulink and C++ environment. Both implementations are used for design and verification of control algorithms as well as for appropriate robot soccer competitions if they are organized in the simulation environment. Important feature, provided collisions are solved realistically, is that control algorithms designed on simulator can later be used in real game situation without major changes. The reprogramming of algorithms when testing them on real playground is thus not needed. Main motivation for the development of such a simulator is to design and study multi-agent control and strategy algorithms in FIRA Small League MiroSot category (3 against 3 robots). However on FIRA (Fedration of International Robot soccer Association) official site (www.fira.net) there exists a simulator for SimuroSot league, which could only be used in Middle League MiroSot (5 against 5 robots). Similar simulator was build in Taiwan [8] where robot motion is simulated by dynamical model while the collisions are oversimplified. A number of different collisions can appear in robot soccer game. Their realization is undoubtedly essential for realistic simulator. However, the most challenging and problematic from the modelling point of view is collision between robots which is presented in more detail.

Good mathematical background in rigid body collisions modelling and simulation can be found in [1]. Another useful contribution in the field of robotic simulators is [7] where collisions are treated by spring-dumper approach rather than by impulse force only. The use of spring-dumper linkage in the collision makes change of velocities continuous, which is less problematic for simulation than discontinuous change of velocities [5] obtained by impulse usage. However, spring dumper coefficients are not easy to identified and also collisions when observed from macroscopic time scale (as it is simulation) are indeed discontinuous events.

In the paper some novel ideas of collision formulation and realization are used. The presented simulator is developed mainly as a tool in control and strategy design of multi-agent system in real world and therefore needs to be realistic. Strategy design could be performed also on real plant but there are some important reasons which

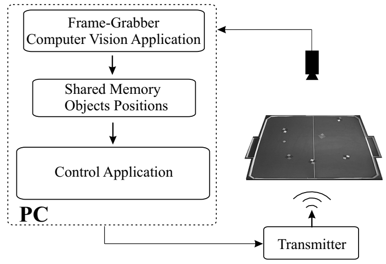

Fig. 1. Robot soccer setup.

benefit the usage of realistic simulator. They are stated in the paper. Collisions are simply solved by mathematically correct discontinuous change of velocities (states of the velocity integrators), which is more convenient for realization than simulating collisions by applying impulse force [1] [7]. The problem of collision detection and the method of finding exact time of the collision are exposed too.

The paper is organized as follows. First a brief system overview is revealed, followed by the mathematical model derivation of basic agents (robots and ball) and short description of different collisions in the game. Then the proposed model of robots collision, namely collision detection and its realization, is explained in more detail. Validation results of the developed mathematical model describing collisions between robots is depicted in section 5. The paper ends with conclusions and some ideas for future work.

2. SYSTEM OVERVIEW

The robot soccer set-up (see Figure 1) consists of six MiroSot category robots (for two teams) of size 7.5*cm* cubed, orange golf ball, rectangular playground of size $1.5 \times 1.3m$, *JAI* MCL-1500 camera, frame-grabber Matrox Meteor II, and personal computer. The vision part of the program processes the incoming images to identify the positions and orientations of the robots and the position of the ball. Finally, the control part of the program calculates the linear and angular speeds, v and ω that robots should have in the next sample time according to current situation on the playground. Calculated speeds are sent to the robots by radio connection.

As seen from Figure 1 there are two applications running on personal computer. Namely computer vision and control application. The communication among them is realized through shared memory.

Figure 2 depicts the situation where real system (playground with robots and ball) is replaced by simulator. Robots and ball movements are simulated in Matlab Simulink

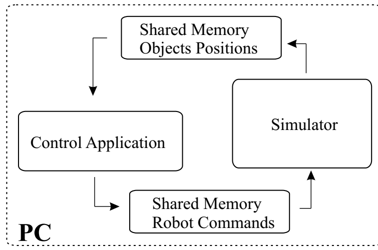

Fig. 2. Simulator setup.

environment or in C++ language while control algorithm and shared memory for object positions remain unchanged. Communication between control and simulator is again realized through another shared memory for calculated robot speeds (commands).

Which are the important advantages of the simulation environment? In real game robot positions and orientations of the object from the playground are obtained by camera and computer vision program. The role of simulator is therefore to avoid the usage of hardware (except PC), which is expensive and needs a large place to be set up. In addition such system is in general not mobile and it is time consuming to manipulate with. Finally the organization of mass competitions is expensive and problematic.

Simulator runs in real time so that data coming from simulator appear in the same time intervals as in real set-up. However, it is possible for simulator to run faster in order to speed up experiments or slower than real time to enable easy visualization of the scene. Another advantage of the structure presented in Figure 2 is that all modules for both competitors (simulator, two control applications) can run on one computer. As already mentioned the same control program can without any change be applied to real or simulated game.

3. MATHEMATICAL MODELLING

To simulate robot soccer game first mathematic motion equations should be derived. The playground activities consist of two kinds of moving objects: robot and ball. Therefore their motion modelling [9] is presented in the sequel.

3.1. Robot Model

The robot has a two-wheel differential drive located at the geometric centre, which allows zero turn radius and omni-directional steering because of nonholonomic constraint [6]. Inputs to the model are angular velocities of right and left wheel (ω_R, ω_L)

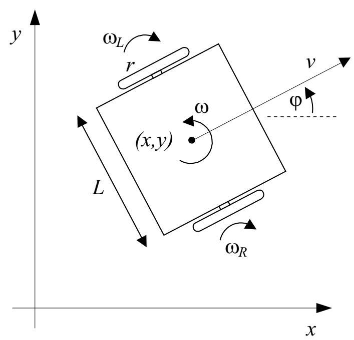

Fig. 3. Symbol description.

while the outputs are position data (x,y,φ). Robots angular (ω) and linear (v) velocities are obtained from the following set of equations

$$\begin{aligned} v &= (\omega_R + \omega_L)\tfrac{r}{2} \\ \omega &= (\omega_R - \omega_L)\tfrac{1}{L} \end{aligned} \tag{1}$$

where L is the robot size (see Figure 3) and r is wheel radius. Dynamics of both motors can be modelled by first order systems

$$\begin{aligned} \dot{\omega}_L &= \tfrac{1}{T}(u_L - \omega_L) \\ \dot{\omega}_R &= \tfrac{1}{T}(u_R - \omega_R) \end{aligned} \tag{2}$$

where T is time constant and u_L and u_R stand for voltage values (reference angular velocities of wheels) applied to motors. The robot kinematics is finally defined as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & 0 \\ \sin(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

3.2. Ball Model

Model of the ball rolling across the playground can be treated as independent of both directions. Mathematical modelling of ball motion can be efficiently derived using Lagrangian equations [15]

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}_s}\right]-\frac{\partial L}{\partial q_s}+\frac{\partial P}{\partial \dot{q}_s}=F\left(t\right) \tag{4}$$

where Lagrangian L represents difference between kinetic and potential energy, P is power function (dissipation function), q_s stands for generalized coordinate and $F\left(t\right)$ is external force respectively. For dimension x the following equation is obtained

$$L=W_K-W_P=\frac{1}{2}m\dot{x}^2+\frac{1}{2}J\dot{\varphi}^2=\frac{1}{2}\left(m+\frac{J}{R^2}\right)\dot{x}^2 \tag{5}$$

where m stands for ball mass, J for its moment of inertia, φ for angle of ball rotation and R for ball radius. The power function is

$$P=\frac{1}{2}f_v\dot{x}^2+K_c m g\dot{x} \tag{6}$$

wheref_v is viscos friction coefficient, K_c is Coulomb coefficient of rolling friction and g gravitation acceleration. Moment of inertia of the ball is defined as

$$J=\frac{2}{5}mR^2 \tag{7}$$

After inserting Equations (5) and (6) in Equation (4) for $q_s=x$ the following relation is obtained

$$\ddot{x}=\frac{F(t)-\dot{x}\cdot f_v-K_c m g}{m+{}^{J}\!/_{R^2}}=\frac{F(t)-\dot{x}\cdot f_v-F_f}{m+{}^{J}\!/_{R^2}} \tag{8}$$

In the simulation the Coulomb friction force ($F_c=K_c m g$) has to be used only when the ball is moving otherwise this force will start to push the ball in opposite direction. Because external force $F(t)$ takes nonzero impulse values only when collisions appear and its values are nearly always larger than F_c, the Karnopp's model of Coulomb friction [2] [3] is not required. Friction force (F_f) is thus determined by

$$F_f=\begin{cases}0, & \dot{x}=0\\ \dot{x}/k, & 0<|\dot{x}|<k\\ sign(\dot{x})F_c, & |\dot{x}|\geq k\end{cases} \tag{9}$$

where k is a correspondingly chosen small positive number. If $k=0$, classical Coulomb friction model [2] is obtained, which introduces oscillation problems in simulation [3]. Similar model can be written also for dimension y.

4. COLLISION MODELLING

During the motion of robots and ball on the playground several collisions between them are possible. The latter are given as submodels and describe the collision between moving objects: the robot-ball collision model, the robot-boundary collision model, the ball-boundary collision model and the collision between robots model. Most of the listed models are relatively simple for realization and are here just briefly mentioned. In the collision between ball and boundary elastic collision is supposed where the tangential velocity component to the boundary remains the same while the normal velocity component changes sign and is multiplied by a factor less than one representing energy loss. Similar procedure is followed in robot-ball collision except that relative ball velocity according to the robot is calculated. Also actual robot shape can be considered. The robot-boundary collision can under certain presumptions be solved by modified model describing collisions between robots. Among mentioned collisions in robot soccer game the most challenging one is collision between robots. Thus its modelling background and realization in the realistic simulator is presented in the sequel.

4.1. Collisions Between Robots

The collision of two or even more robots is undoubtedly problematic from the modelling point of view. However, the complexity of the model must be strongly dependent on the demands of the realistic simulator where the compromise between realism and simulation speed must be found according to the simulation usage aims. During simulator design a few more or less approximate solutions were tested until finally the best one was implemented. When designing control strategy of the robot soccer game, it may seem that collisions between robots are not so important because one focuses mainly on shots on goal, on passes, organizing defence and similar actions while collisions between robots are more or less undesired. However, collisions between robots are quite frequent in the game and in the case of defence also very important and must therefore be correspondingly treated in a realistic simulator.

4.2. Collision Detection

Collision detection algorithm consists of two steps. In the first only information about possible collision is obtained. The second step is then performed only if possibility obtained from the first step exists. In the second step separating plane between objects is found. In the simulation environment also the penetration of one robot to another is normally possible. The reason why collision detection is performed in two steps is only due to lower computational burden. Thus the second step is performed only in situations where collision is almost inevitable.

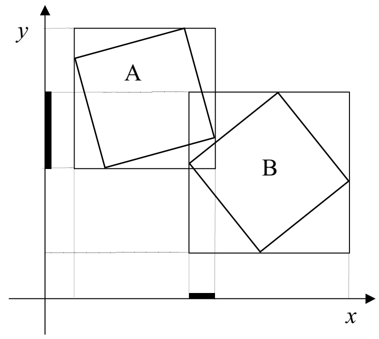

Fig. 4. Overlapping of bounding boxes in both directions.

The first step is performed by analysing robots bounding boxes. The latter have their sides parallel to the global coordinate axes, thus representing the rectangle in which robot in its current position is enclosed (see Figure 4). The possibility of two objects colliding exists only if bounding boxes overlap. Overlapping between two bounding boxes is determined by checking if their sides overlap in both axis directions (x and y) at the same time. As mentioned before the second step is performed only if overlapping of bounded boxes from the first step exists. The separating plane is calculated so that one object (convex polyhedron) is on one side and the other object on another side of separating plane. The latter always exists if two objects do not invade.

4.3. Calculation of Separating Plane

In a two-dimensional space separating plane is a straight line and should thus contain the side of one of the two objects which participate in collision (see Figure 6). Suppose A and B are two convex polygons. Let R_i^A be i-th corner of the polygon A and R_j^B is j-th corner of the polygon B. Let us further on define orientation of a triangle defined by three points (R_1, R_2, R_3) with coordinates $(x_1, y_1)\,,(x_2, y_2)\,,(x_3, y_3)$ by the following equation

$$\det \begin{vmatrix} x_1 - x_2 & x_1 - x_3 \\ y_1 - y_2 & y_1 - y_3 \end{vmatrix} = 2 \cdot o \cdot S \tag{10}$$

where o is the orientation of the triangle with values 1 or –1 while S is triangle area. Only the sign of determinant in Equation (10) is thus important.

Let us first observe the side of the polygon A, which is determined by the neighbour corners R_i^A in R_{i+1}^A. For all corners R_k^A of the polygon A, with $k \neq i$ and $k \neq i+1$,

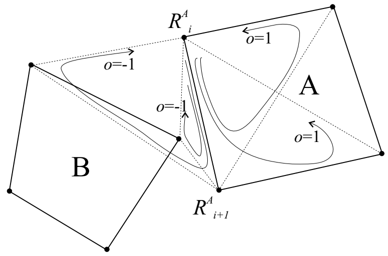

Fig. 5. Separating plane determination.

the triangles $\left(R_i^A, R_{i+1}^A, R_k^A\right)$ have then the same sign (see Figure 5). The neighbour corners R_i^A in R_{i+1}^A define separate plane only if the following statement is true:

- for all corners R_j^B of polygon B the orientation of the triangle $\left(R_i^A, R_{i+1}^A, R_j^B\right)$ has to be of the same sign but different as for triangle $\left(R_i^A, R_{i+1}^A, R_k^A\right)$.

If for one corner R_j^B triangle area S is zero, the sign of this triangle is ignored. If none of the sides of polygon A defines separating plane then the above procedure is repeated so that sides of polygon B are examined.

4.4. Collision Realization

Collision between two robots is realized by force impulse $\vec{J} = \vec{F}\Delta t$, which acts in normal direction $\vec{n}$ of the collision (also normal of the separating plane at the time of the collision, see Figure 6) of two frictionless bodies

$$\vec{J} = j\vec{n}(t_0) \tag{11}$$

where t_0 is time of the collision and j is amplitude of the force impulse. For the normal direction of the collision the following relation can be written

$$v_{rel}^{+} = -\varepsilon v_{rel}^{-} \tag{12}$$

meaning that absolute value of relative velocity in normal direction after collision v_{rel}^{+} remains the same or is lowered for energy loss factor ε in comparison with absolute value of relative velocity in normal direction before collision v_{rel}^{-}. From the property (12) the amplitude of force impulse j in Equation (11) can be estimated according to procedure described in [1]. Let $\dot{\vec{p}}_a^{-}(t_0)$ be the velocity of contact point of robot

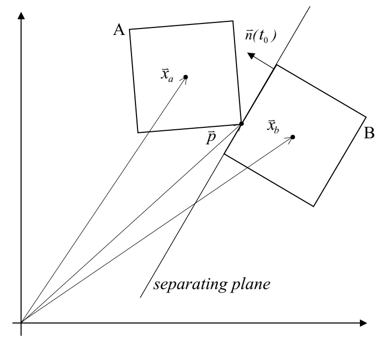

Fig. 6. Collision of two robots.

A before impulse $\vec{J}$ is applied and $\dot{\vec{p}}_a^+(t_0)$ velocity of contact point of robot A after applying impulse. Similarly notations $\dot{\vec{p}}_b^-(t_0)$, $\dot{\vec{p}}_b^+(t_0)$ are used for the second robot B taking part in the collision. Relative velocity in normal direction before applying impulse is thus

$$v_{rel}^- = \vec{n}(t_0) \cdot (\dot{\vec{p}}_a^-(t_0) - \dot{\vec{p}}_b^-(t_0)) \tag{13}$$

and after applying impulse

$$v_{rel}^+ = \vec{n}(t_0) \cdot (\dot{\vec{p}}_a^+(t_0) - \dot{\vec{p}}_b^+(t_0)) \tag{14}$$

Defining

$$\vec{r}_a = \vec{p} - \vec{x}_a(t_0) \tag{15}$$

where $\vec{r}_a$ is the displacement vector between mass centre $\vec{x}_a$ of the robot A and collision point $\vec{p}$. Further let $\vec{v}_a^-(t_0)$ and $\vec{\omega}_a^-(t_0)$ be the liner and angular velocity of robot A before and $\vec{v}_a^+(t_0)$ and $\vec{\omega}_a^+(t_0)$ after applying force impulse. The following velocities can be written for mass centre of robot A and for the point of collision

$$\vec{v}_a^+(t_0) = \vec{v}_a^-(t_0) + \frac{j\vec{n}(t_0)}{M_a} \tag{16}$$

$$\vec{\omega}_a^+(t_0) = \vec{\omega}_a^-(t_0) + I_a^{-1}(\vec{r_a} \times j\vec{n}(t_0)) \tag{17}$$

$$\dot{\vec{p}}_a^+(t_0) = \vec{v}_a^+(t_0) + \vec{\omega}_a^+(t_0) \times \vec{r_a} \tag{18}$$

Here $M_a$stands for mass of robot A and I is the corresponding moment of inertia. The same notation is used for robot B. Inserting Equations (16) and (17) to Equation (18), the following relation is obtained

$$\dot{\vec{p}}_a^+(t_0) = \dot{\vec{p}}_a^-(t_0) + j \cdot (\frac{\vec{n}(t_0)}{M_a} + I_a^{-1}(\vec{r_a} \times \vec{n}(t_0))) \times \vec{r_a} \tag{19}$$

The velocity in the contact point of robot B considering opposite direction of impulse force is thus

$$\dot{\vec{p}}_b^+(t_0) = \dot{\vec{p}}_b^-(t_0) - j \cdot (\frac{\vec{n}(t_0)}{M_b} + I_b^{-1}(\vec{r_b} \times \vec{n}(t_0))) \times \vec{r_b} \tag{20}$$

Inserting Equations (19) and (20) into Equation (14) and then combining obtained equation with Equation (12) the amplitude of impulse is finally calculated as

$$j = \frac{-(1+\varepsilon)v_{rel}^-}{\frac{1}{M_a} + \frac{1}{M_b} + \vec{n}(t_0) \cdot (I_a^{-1}(\vec{r_a} \times \vec{n}(t_0))) \times \vec{r_a} + \vec{n}(t_0) \cdot (I_b^{-1}(\vec{r_b} \times \vec{n}(t_0))) \times \vec{r_b}} \tag{21}$$

When the impulse is calculated, velocities of the robots are determined according to Equations (16) and (17) which are then used to determine new initial states of the integrators in the simulator. It is namely equivalent to impulse force because of collision simulation but more suitable and accurate for realization. To obtain accurate t_0 zero crossing algorithm implemented in Matlab Simulink [13] could be used in order to assure accurate integration of discontinuous velocities signals. This algorithm simply changes integration step by bisection, according to some input variable (distance between robots multiplied by a sign which is negative if robots penetrate), until exact time of discontinuity is achieved. However, the problem of high frequency oscillations around a discontinuity (chattering) appears when two or more robots stay in contact (robots pushing each other). Therefore step size of simulation becomes very small which results in halting of the simulation. Thus a better solution is to check for correspondingly small distance between one robot corner and the separating plane belonging to another robot. If separating plane does not exist, the time before penetration of the simulated robots must be taken into account.

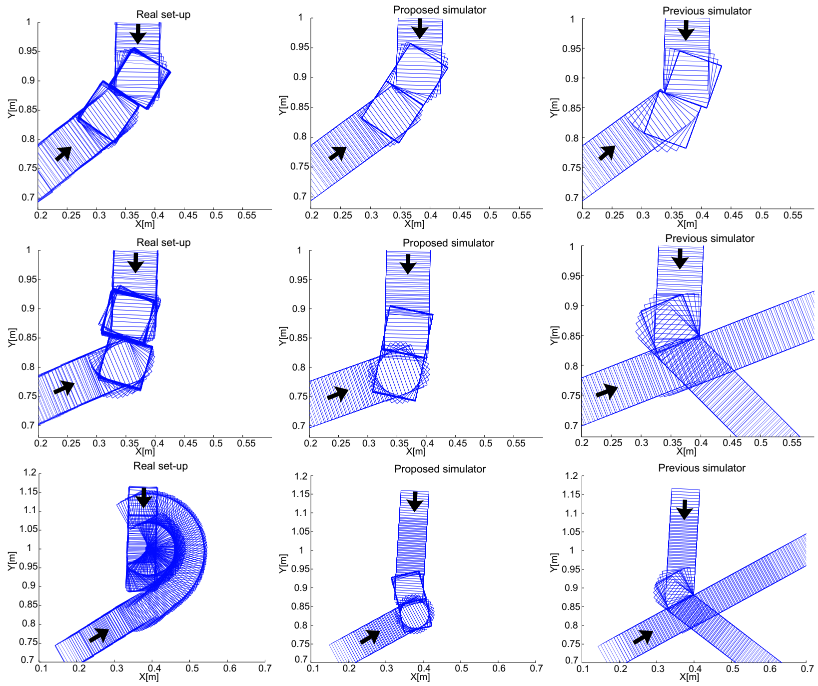

Fig. 7. Course of collision between two robots - real setup and two simulators.

5. VALIDATION

Model of the robot collision from section 4 is included in the proposed realistic simulator. The validation of the developed new version of the simulator is done by the aid of two-fold comparison. The first is among proposed simulator and real set-up where the experiments with the same initial conditions (starting positions, orientations and velocities) were performed. The second one, however, compares the proposed simulator with the previous version, including simpler robots collision model, for which can be stated that it is at least of the same quality (but probably better) than the simulator available on the FIRA official site. The given visual presentation in Figure 7 is very illustrative showing the difference between the compared subjects (first column – real set-up, second column – proposed simulator and third column – simpler version of simulator) for three different situations (rows in Figure 7). From the proposed representation also the estimation of robots course and their speeds in certain time (sample time is 33 *ms*) can be observed. The first row of Figure 7 shows the situation where all compared subjects are relatively equal. In the second row simple version of the simulator gives entirely wrong situation while the real set-up and proposed simulator give sufficiently similar results. The real situation with sliding wheels of the robot on real set-up is shown in the third row in Figure 7. Here of course both simulators give wrong results.

From Figure 7 it is evident that the proposed robots collision model improves the behaviour of the simulator to the reasonable extent, which means that simulated situations cover a vast majority of collisions in real game sufficiently well. However, a lot of factors in real set-up are of significantly stochastic character what means that their modelling is not justifiable from the usable simulator point of view (fast enough on available personal computers, simple enough, *etc*.). The mentioned factors are: nonuniform friction, dirt or dust on the playground or wheels, shape of the robot, robot strength which depends on battery status, wheel sliding, friction is different for the direction along or perpendicular to the direction of wheels, *etc*.

If validation is performed over longer time interval shown results are useless due to above reasons. Main goal of the paper however is to present reasonably accurate robot collision model and thus contribute to obtain more realistic simulator, which would be used as a tool in the process of strategy and control algorithms design. Therefore, the validation of the simulator as a whole should be done through transferability of obtained strategy algorithms to the real system. It can be confirmed that the behaviour of simulator is similar enough to the real setup what means that the designed algorithms (strategy and low level control) can without modifications directly be used also in real games. Simulator was also tested in the local robot soccer simulation league organized at the faculty. Fifteen students participated in seven teams. In short time (two months) they manage to build their own strategy application entirely on the simulator. The winning team of the simulation league took part in European championship organized in Vienna, Austria in April 2002. They won the second place in Small MiroSot league (real robots) with the same strategy application as developed on the simulator.

6. CONCLUSIONS

In the work the improvement of the existing robot soccer game simulator is presented. The main effort is put in the modelling of two robots collision which was found out to be the most problematic.

The designed simulator has significant improvements in comparison with the available simulator in MiroSot leagues (simulator for SimuroSot) and other available simulators. The advantages are: realistic shape of robots inclusion, which gives better simulation of robot ball interactions, collisions with robots, robots and boundary interactions and situations where ball is captured between two objects (it cannot invade any object). The presented simulator validation of collisions shows good agreement of the proposed simulator with the real set-up. However, it would be nonsense to tend to include also the phenomena which are of significant stochastic nature. The main goal of the simulator development was namely to enable acceptable transferability of certain strategy algorithms to real system. According to this goal it came out that the strategies and low level control developed on the proposed simulator can without modifications and directly be used successfully also in real games.

Our group has worked with the robot soccer for three years and from our experiences robot soccer is not only an ideal playground for studying multi-agent systems but also represents a good and attractive plant in education process. It is of interest for undergraduate and postgraduate students, because it speeds up the progress of knowledge and experiences in robotic soccer. By involving more students in this area new interesting ideas are expected, which could not be achieved only by experimenting on real system.

The introduced Small League MiroSot category simulator is not the final version, although it proves to be good approximation of the system. The robot-boundary, robot-robot and robot-ball collisions are realistic and they need no further improvements, which is not the case for multiple robots collision. The latter needs to achieve a more accurate representation of reality.

REFERENCES

1. Baraf, D.: An Introduction to Physically Based Modeling: Rigid Body Simulation II – Nonpenetration Constraints. *SIGGRAPH '97 Course notes*, Carnegie Mellon University, 1997.
2. Bonsignore, A., Ferretti, G., and Magnani ,G.: Analytical Formulation of the Classical Friction Model for Motion Analysis and Simulation. *Mathematical and Computer Modelling of Dynamical Systems*, 5, 1 (1999), pp. 43–54.
3. Egeland, O., Gravdahl, J.T.: *Modeling and Simulation for Automatic Control*. Marine Cybernetics, Trondheim, Norway, 2002.
4. Ferber, J.: *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, Essex, England, 1999.
5. Fremond, M.: Rigid bodies collisions. *Physical Letters A*, 1 (1995), pp. 34–41.
6. Kolmanovsky, I. and McClamroch, N.H.: Developments in Nonholonomic Control Problems, *IEEE Control Systems* 15 (1995), pp. 20–36.
7. Larsen, E.: A Robot Soccer Simulator: A Case Study for Rigid Body Contact. *Sony Computer Entertainment America R& D*, March 2001.
8. Liang, T.C. and Liu, J.S.: A Distributed Mobile Robot Simulator and a Ball Passing Strategy. *Technical Report TR-IIS-02-007*, Institute of Information Science, Academia Sinica, Nankang, Taiwan, 2002.
9. Matko, D., Karba, R. and Zupanèiè, B.: *Simulation and Modelling of Continuous Systems, A Case Study Approach*. Prentice-Hall, Englewood Cliffs, USA, 1992.
10. Minoru, A., D'Andrea, R., Birk, A., Kitano, H. and Veloso, M.: Robotics in Edutainment. *Proceedings of the 2000 IEEE International Conference on Robotics & Automation (ICRA'2000)*, San Francisco, USA, April 2000, pp. 795–800.
11. Moss, S. and Davidsson, P.: *Multi-Agent-Based Simulation*, Springer-Verlag, New York, 2002.
12. Stone, P. and Veloso, M.: Multiagent Systems: A Survey from a Machine Learning Perspective, *Au-*

tonomous Robots 8 (2000), pp. 345–383.
13. The Math Works, Inc.: *Simulink, Dynamic System Simulation for Matlab*. Natick, USA, 1998.
14. Topolov, A.V. and Tzafestas, S.G.: Fuzzy-Neural-Genetic Layered Multi-Agent Reactive Control of Robotic Soccer. In: Braha, D. (ed.):*Data Mining for Design and Manidacturing: Methods and Applications*, Kluwer, Boston, 2001, pp. 417–442.
15. Welles, D.A.: *Lagrangian Dynamics.* Schaum's Outline Series, McGraw Hill Book Company, 1967.