On the Aspects of Simulation in the RoboCup Mixed Reality Soccer Systems

Reinhard Gerndt [1], Carsten Schridde [1], Rodrigo da Silva Guerra [2]

[1] University of Applied Sciences Braunschweig/Wolfenbuettel, Wolfenbuettel, Germany
[2] Graduate School of Engineering, Osaka University, Osaka, Japan
r.gerndt@fh-wolfenbuettel.de c.schridde@fh-wolfenbuettel.de
rodrigo.guerra@ams.eng.osaka-u.ac.jp

Abstract.

This paper describes the different aspects of simulation in the Mixed Reality (MR) (previously physical visualization) sub-league of the RoboCup soccer simulation league. Unlike other leagues, the Mixed Reality combines simulation and reality. The classical RoboCup simulation leagues offer a test bed for abstract behaviors and complex environments. Leagues with real robots allow interacting with physical environments. Mixed Reality incorporates all these aspects. It thus bridges the gap between simulated and real robot leagues. In this paper we will present the overall structure of the system and point out the different aspects simulation plays in the MR system.

1 Introduction

The Eco-Be!-activities, started by a co-operation between Osaka University and Citizen corporation [1], resulted in a considerably new sub-league of the RoboCup Soccer simulation. The Mixed-Reality (MR) approach combines physical and virtual aspects and thus relies on simulation of multiple components as one of the main operational principles [2, 3]. The RoboCup MR system is based on miniature robots, operated on a horizontally mounted screen. The screen may display any type of environment and other, simulated objects, even other robots. Figure 1 shows a soccer field as an example for a typical RoboCup environment. In the bottom half three real robots are placed with a virtual ball at the kick-off point and a virtual marker close to one of the robots. The real robots may interact with each other and the virtual and real objects, placed or displayed on the screen.

Simulation controls the virtual objects and robots, whilst the real robots are controlled by software agents. Simulation is also used for software development and self-learning approaches, e.g. reinforcement learning, which rely on a high number of learning cycles. Carrying out the learning phase only with real robots would not be feasible due to the extensive period of time this would need. Thus simulating the environment, simulating the robots and simulating the learning examples plays an important role in the RoboCup MR sub-league.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

Fig. 1 System setup with a 42" display (example soccer field).

The major objectives of the approach are research and education. Education is addressed by an easy access to the programming and low costs of the overall system with a considerably high number of robots. Major research aspects are interoperation of a large set of autonomous robots in multiple applications and environments.

A first approach to the system is considerably easy. In a basic soccer setup, modules for physical control of the real robots and simulation of the virtual ball are readily available. It can be easily adapted if more sophisticated behavior is required. The approach is based on simple vector arithmetic and makes use of a freely available physics engine for simulation.

2 The RoboCup Mixed Reality System

The basic hardware setup of the RoboCup MR system consists of a horizontally mounted display and a set of micro-robots. The left part of figure 2 sketches the hardware setup. One or more computers control the virtual environment, displayed on the screen, and the micro-robots. A camera that is mounted above the screen captures the scenery with all virtual and real objects. Image processing allows determining the position and orientation of the robots and other real objects on the screen.

The size of the robots is approximately 2.5 centimeters in all three dimensions (fig. 3). The robots can move freely on the screen. They are driven by two independently controllable wheels, according to the differential drive paradigm. Robots are controlled by individual software agents. For identification, each robot is equipped with an individual marker (not shown in figure 3). The markers allow determining the ro-

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

bots positions and poses. The number of individually distinguishable robots depends on the markers used and on the resolution of the camera.

An IrDa link is used for wireless data exchange. Other wireless data links have been used previously and RF modules have been proposed for future systems.

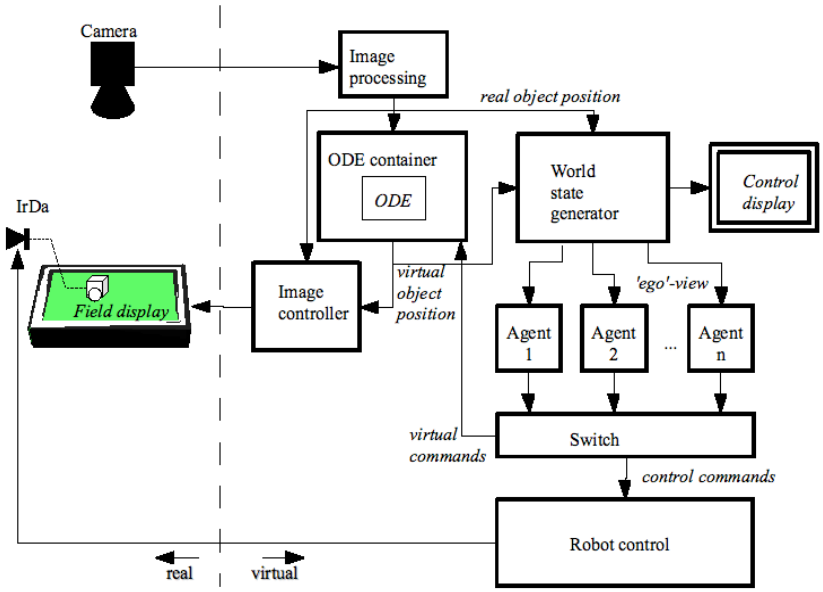

Fig. 2 Structure of RoboCup Mixed Reality system.

The size of the display can be chosen with respect to the overall application. Initially, systems used a 20'' screen, however, nowadays 42'' and larger screens are used. 20" displays were sufficient for an environment for up to 4 or 6 micro-robots. 42'' is a reasonable size for up to 12 robots. A future goal is using two full soccer teams with 22 robots, which may then use a large glass screen and back projection.

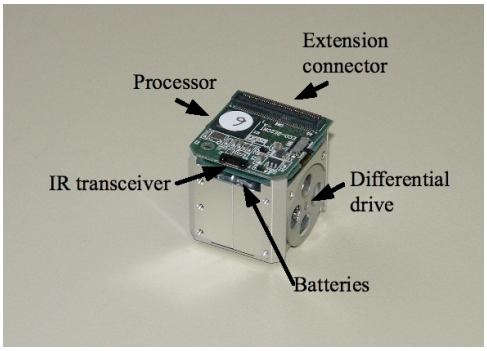

Fig. 3 3rd generation Eco-Be! Robots.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

A crucial part of the system, limiting the size of the scenery and the number of robots are the number and type of camera(s). The cameras need to capture the entire screen and need to have a sufficient resolution to distinguish among the individual markers of the robots. For 20'' displays and the markers used, a single one-megapixel camera was used. The frame rates of the cameras need to match with the speed of the robots. Currently, frame rates of 15 fps (frames per second) are sufficient for most applications. The image processing currently is very straightforward, only distinguishing between robot markers, which can be identified by a simple pattern-matching algorithm with a predefined library. However, with new software, identification of other real objects is feasible.

Processing is done on standard PC, which is typically running a Linux operating system. The right part of figure 2, marked 'virtual', shows the structure of the software. There is an overall framework with a number of modules for in- and output, simulation and control. The image-processing module captures the camera output and provides information on positions and poses of the robots and possibly other real objects to the other software modules. The world state generator generates an individual view for every single robot in the system. A control display can be attached for debugging and development purposes. Figure 4 shows an example for the 'ego-view' for a specific robot, as it is available at the control display.

Fig. 4 Screen-shot of 'ego-view' control display (for robot with 'myid_0').

The individual views are then communicated to the agents that control the robots. There is one individual agent module for every robot. The switch module separates the commands issued by the agents into commands that affect virtual objects, like kicking a virtual ball and control of real robots.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

The robot control module takes care of interfacing and communicating with the robots. The ODE container wraps the Open Dynamics Engine (ODE) [4] physics engine and takes care of simulation of the virtual objects. It processes data of real objects, like position and space occupied, and commands that affect virtual objects. It outputs information on new poses and positions of the virtual objects. The image controller displays all virtual objects on the screen.

3 Role of Simulation in MR System

Simulation plays a central role in the entire MR system. The virtual part of the system, e.g. the ball in a soccer game, is a result of simulation. Simulation is also used to reduce use of the robot hardware, like batteries and drives or to completely avoid it during software development. And simulation is used for automated development of robot behavior and game strategies by learning algorithms.

Mixed Reality consists of at least one real and one virtual or simulated component. The simulation is based on vector arithmetic as a mathematical principle. Simulation of the virtual objects takes place in the ODE container (fig. 2). It contains the physics engine for the physical behavior and interfacing and communication functionality. The freely available Open Dynamics Engine (ODE) [4] was selected for a physics engine because other RoboCup groups already used it and it was freely available. The ODE container receives information on the position of all objects in the scenery and commands from the robot agents. Based on these data, it generates new values for the virtual objects position and orientation and sends them to the image controller. Simulation is based on'ego-centric' vector arithmetic, i.e. other objects are referred to relatively to the own pose vector (fig. 5).

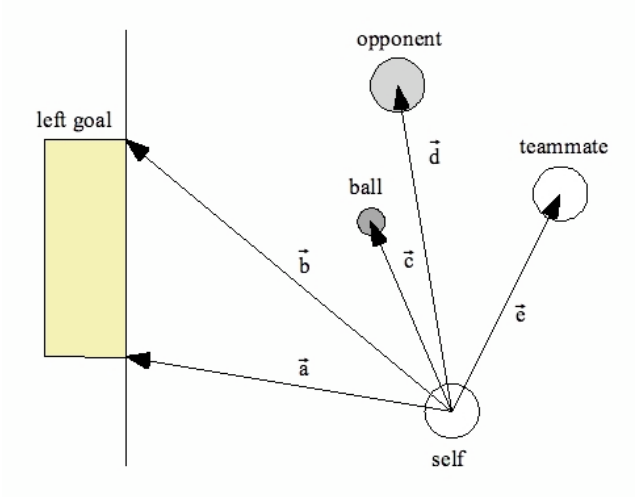

Fig. 5 Vector representation of game situation.

In figure 5 we illustrate how one can take advantage of the two-dimensional vector representation for a very visual strategy planning. As an example, we want our agent

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

to mark an opponent by placing itself between the opponent and the ball. The vector from the ball to the opponent is given by the expression c - d. One could simply scale down this vector, let's say, half-way (c–d)/2, and sum to the vector representing the direction to the ball, yielding the expression (c+c–d)/2. This last expression would be the direction the agent should go. Similarly, if an agent were to kick the ball (c) into the center of the goal (a+b)/2 , it would need to go towards $c+(c–(a+b)/2)(r_r+r_b)$, with r_r the radius of the robot and r_b of the ball.

In a typical MR soccer game, the ODE container calculates the position of the soccer ball and takes care of collisions with robots and virtual objects like goal posts etc. It may also incorporate a virtual referee in order to automatically monitor compliance with soccer rules and to count goals. The degree of accuracy of the simulation may vary. In the example system, there is a collision detection between real and virtual objects and very basic physics. The soccer ball has a momentum and friction is applied for deceleration.

However, not only simulation of virtual objects is an issue in MR. In order to reduce use of the robots and to improve the robots capabilities by learning algorithms, a sufficiently accurate physical simulation of the robots, including aspects like communication delays with the software agents, slippage of the robot wheels on the screen and possibly inaccuracy of the camera and image processing (sensor noise) is required. If done with sufficient care, the simulation can be used instead of the real robots. During the initial development phase many MR RoboCup teams developed an individual simulator, however, with different functionality.

In order to establish learning algorithms, a simulator is a crucial prerequisite. In a self-learning system, the system performs an action and receives a reaction from the environment. It also receives or generates a reward to encourage good actions and discourage unfavorable actions (fig 6).

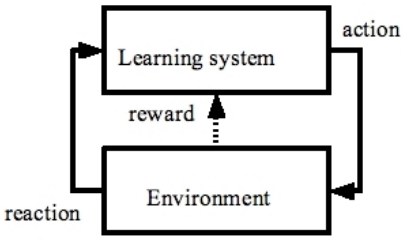

Fig. 6 Structure of learning system.

This kind of learning typically requires several hundred behavioral cycles, which may be drawn e.g. from real games, pre-recorded games or simulations. Physical speed of the real objects is a limiting factor for using real or recorded games. Only simulation allows executing a sufficiently large number of training exercises for learning algorithms. Of course, there still need to be real sequences to identify differences between simulation and real behavior and to verify results.

Another aspect of simulation is pure visualization. There are proposals for remote games with teams at different locations with only their own real robots playing a projection of the other, remote teams robots. Only by combining both 'half' games the overall game comes together.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

4 Structure of simulated system

Using a standard hardware platform allows to have a common simulator without extensive configuration. However, a broad spectrum of applications requires to emphasis different aspects of the simulation. Trade-offs between simulation speed, accuracy and ease of use may serve as an example.

If used without real robots, camera, image processing and the robot control module and robot communication infrastructure are not needed any more. In this case, the robot control module calculates the position and pose of the robots based on the agents control commands. Figures 7 shows the adapted system structure for a pure simulation approach. The only remaining real part is a monitor for visualization. With respect to the virtual part of the system and the software structure, only few changes result, as it is obvious from comparing figures 2 and 7. For pure simulation, a robot simulator module with its own physics module replaces the robot control module. Instead of generating control output from the agent commands, it immediately generates the new robot positions. This of course requires calculation of physical behavior of the now virtual robots. Collision detection is the minimum requirement. However, for learning algorithms more details like slippage are required. In our implementation of the robot simulator we also make use of the Open Dynamics Engine [4], which is already used for simulation of the virtual objects.

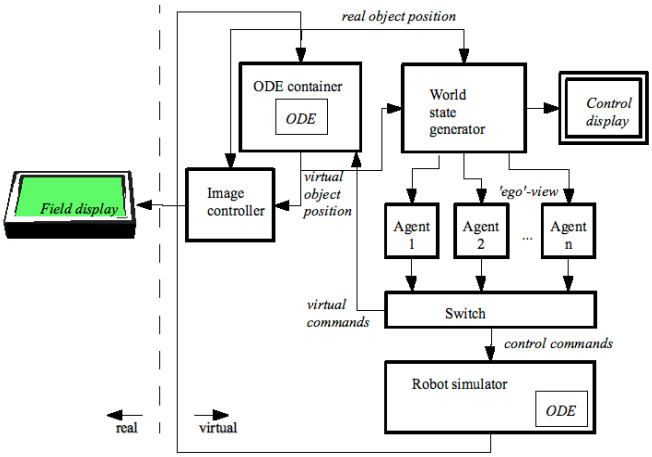

Fig. 7 MR simulation system.

The simulation, used for the virtual part of the basic system is kept very simple. It uses the ODE that is embedded into the ODE container module. The physical aspects of the simulation are derived from the ODE. Data representation is done according to the vector arithmetic approach. For the world state, Cartesian coordinates are used, whilst each robot receives a view in individual polar coordinates. In the example ODE container additional simplifications are done after calculating the physical behavior for the soccer ball, where the 3-D behavior was mapped to a 2-D behavior. Simulation of the robots is implemented in a similar way. The robot simulator module is written

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166

in JAVA. The ODE module is written in standard C code for reasons of simplicity. An ODE wrapper is used to allow native calls to the ODE [5]. In the future a JAVA module may be available. With respect to simulation of physical behavior, both simulation modules rely on the functionality provided by the ODE.

5 Conclusion

The RoboCup MR system, with a close coupling of simulation and real components offers an ideal test bed for different aspects of simulation. With its high degree of standardization, it is ideally suited to share a common simulation tool but individual developments of the teams require adaptations, such that robustness of the simulation may become an interesting issue. The overall system with robots controlled by intelligent agents, proved very robust against simple simulation models and even inaccurate simulation, e.g. soccer ball with unusual physics, like minimal friction, disability to bounce and so on.

The MR system also is suitable to get students involved with robotics and simulation. Simulation is a key aspect of the system to control the behavior of the virtual objects. Simulation also is a crucial prerequisite for the development and improvement of robotic applications and for implementation of self-learning systems. The Robocup Mixed Reality approach with an easy to understand vector arithmetic and a freely available physics engine, offers students a straightforward access to simulation. After they gained a deeper insight into the subject, they then know how to apply simulation for further improvements of the system and for other applications.

Easy access to the simulation within the MR system may have been a reason for most teams to rather tailor the existing approach to the individual needs than to look for a possibly more sophisticated but harder to integrate simulator, as used by other RoboCup leagues. This even holds for teams that are involved in multiple leagues. However, many key aspects are drawn from previous RoboCup experiences.

For the future a Mixed Reality league may bridge the gap between pure simulation leagues with elaborated game strategies for complete teams and the physical robotic leagues with real interaction, but a limited number of robots. It may thus help to transfer simulation-league results to the real world and verify its robustness in a real world.

Reference

[1] Eco-Be! Homepage in the Internet: www.eco-be.com.

[2] Rodrigo da Silva Guerra, Joschka Boedecker, Shinzo Yanagimachi, Minoru Asada. Introducing a New Minirobotics Platform for Research and Edutainment. 4th International Symposium on Autonomous Minirobots for Research and Edutainment. October 2007.

[3] R. da Silva Guerra, J. Boedecker, N.M. Mayer, S. Yanagimachi, H. Ishiguro, and M. Asada. A New Minirobotics System for Teaching and Researching Agent-based Programming. In Computers and Advanced Technology in Education, Beijing, 2007.

[4] Open Dynamics Engine (ODE) in the Internet: www.ode.org

[5] JAVA-ODE Wrapper. in the Internet: https://odejava.dev.java.net/.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 159-166