Tech United Eindhoven Team Description 2013

Middle Size League

F.B.F. Schoenmakers, G. Koudijs, C.A. Lopez Martinez,
M. Briegel, H.H.C.M. van Wesel, J.P.J. Groenen, O. Hendriks, O.F.C. Klooster,
R.P.T. Soetens, M.J.G. van de Molengraft

Eindhoven University of Technology,
Den Dolech 2, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
www.techunited.nl, techunited@tue.nl

Abstract. In this paper we discuss progress in mechanical, electrical and software design of our middle-size league robots over the past year. This includes a redesign of our ball handling mechanism, improvements in passing on the strategy layer, kinect enabled ball detection and advancements in alternative approaches for pass execution. Knowledge sharing among the robotics community has been facilitated by adding drawings of our soccer robot to ROP, an online wiki platform aimed at sharing robot design under an open hardware license. We will also elaborate on steps we took towards creating a low-cost middle-size league robot and on initial steps towards an autonomous middle-size league referee.

Keywords: RoboCup soccer, middle-size league, low-cost robot design, autonomous referee, ball handling, ball passing strategy

1 Introduction

Tech United Eindhoven is a RoboCup team of Eindhoven University of Technology. It consists of PhD, MSc and BSc students, supplemented with academic staff members from different departments. The team was founded in 2005, originally only participating in the Middle Size League (MSL). Six years later service robot AMIGO was added to the team, which now also participates in the RoboCup@Home league [6]. Knowledge acquired in designing hard- and software for soccer robots proved to be an important resource while creating our service robot. In 2012 also TUlip, competing in the adult-size humanoid league, joined the team.

This paper describes our major improvements regarding the mechanical, electrical and software architecture of the middle size league robots. It is part of the qualification package for the RoboCup 2013 World Championships in Eindhoven, and is split in four main sections. First an introduction of our robot platform is presented. Second we briefly describe progress towards the design of a low-cost soccer robot, followed by steps we are taking in using feedback from a top-cam to automate the job of the referee. In the final section main improvements regarding our own robot platform, are presented.

Many of the points of improvement described in this latter section are a direct result of the mid-line passing rule, introduced by the MSL technical committee in the 2012 rulebook. Enforcing teams to make a pass before reaching the other half of the field boosted dynamic team play[1] and brought our league significantly closer to the ultimate goal of RoboCup. The new rule pushes hardware design (subsection 5.1) and moves the competition towards more advanced levels of multi-agent coordination (subsection 5.2).

2 TURTLE platform

The robots of our RoboCup team have been named TURTLEs (acronym for Tech United RoboCup Team: Limited Edition). We are using our fifth generation of soccer robots, built in 2011 (Fig. 1).

[1] Video compilation of final match at RoboCup 2012 in Mexico City: http://goo.gl/0FyAJ

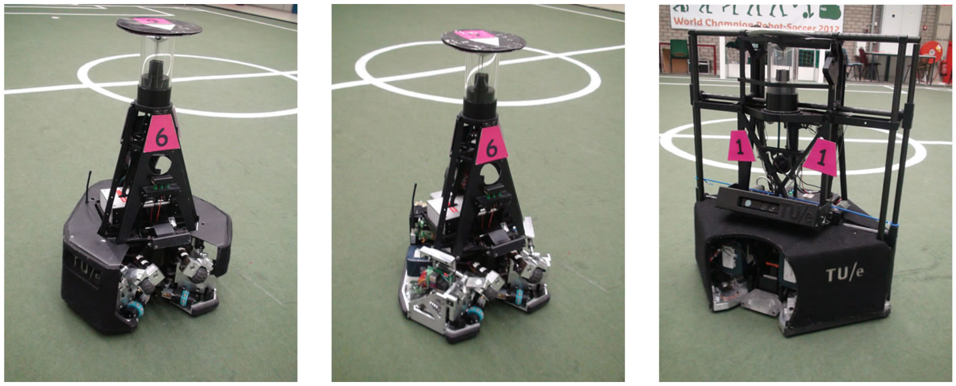

Fig. 1. Fifth generation TURTLE with cover attached (left), with cover removed (middle) and customized goalkeeper design (right).

2.1 Hardware

Three 12V Maxon motors, driven by Elmec Violin 25/60 amplifiers and two Makita 24V, 3.3Ah batteries, are used to power our omnidirectional platform. Each robot except for the goalkeeper, is equipped with an active ball handling mechanism, enabling it to control the ball when driving forwards, while turning, and even when driving backwards [1]. Previously our ball handling mechanism was optimized especially for dribbling the ball. Given the increased importance of being able to quickly catch an approaching ball, we redesigned this system (subsection 5.1).

The solenoid shooting mechanism, which is powered by a 450V, 4.7mF capacitor, provides an adjustable, accurate and powerful shot [7]. To acquire information about its surroundings, the robot uses an omnivision unit, consisting of a camera focussed at a parabolic mirror. An electronic compass is implemented to differentiate between omnivision images on our own side versus on the opponent side of the field.

A detailed list of hardware specifications, along with CAD files of the base, upper-body, ball handling and shooting mechanism, has been published on a ROP wiki.[2] The Robotics Open Platform (ROP) is an initiative of Eindhoven University of Technology [5]. It fosters sharing of hardware knowledge by providing a wiki platform where robotics researchers can share specs, electrical schemes and three-dimensional drawings of their design.

2.2 Data-Acquisition and Software

To facilitate data-acquisition and motion control, the robots are equipped with EtherCAT devices[3], connected to the onboard host computer via ethernet [4]. Each robot is equipped with a mini-PC running a pre-emptive Linux kernel. The software is automatically generated from Matlab/Simulink models via the RTW toolbox. Over the past couple of months we have upgraded our development environment, now using Matlab R2012a running on the latest long term support Ubuntu distribution (Ubuntu 12.04).

Our software is divided in three main parts: a vision-, worldmodel- and motion module. While the vision and worldmodel modules both run at 30Hz, the motion module uses a much higher sampling rate of 1000Hz. The vision module provides localization of ball, teammates and opponents. Hereafter the worldmodel combines information from all robots to get a unified estimation on peer and opponent positions. The motion module contains strategy, pathplanning and controllers.

[2] http://www.roboticopenplatform.org/wiki/TURTLE
[3] Beckhoff, main catalog 2009, http://goo.gl/0XbRR

3 Low-cost MSL robot

Ever since its genesis in 1997, the middle-size league has been pushing the scientific boundary in multi-agent coordination and mechatronic systems design [8]. Well-designed hardware and clever software are equally important in our league. But while advancements in software can be shared among teams cheaply, copying advances in hardware comes with the costs of hiring external craftsman and ordering parts. For new teams entering our league, this results in high startup costs and many hours of work before being able to participate at a reasonable level.

Therefore together with companies in the wider Eindhoven area, we are designing a low-cost MSL robot. After completion, this newly designed robot should provide a base-platform, recapitulating hardware knowledge gained over the past fifteen years of MSL's existence as much as possible. It should be easily adjustable and extendible such that the league maintains innovative on the hardware level. Teams focussing on a specific part of the robot (e.g. vision) should be able to add their own equipment or replace existing units with their own.

Starting point is a cost-driven redesign of the Tech United TURTLE robot. Our target is to make the robot available at a price of 5000 euros, which is a significant reduction in price. A feasible reduction though, because (i) a lot of cost-driven design expertise exists in industry and (ii) the aim of the project is to sell ten teams of robots, allowing for package deals with suppliers.

A ball handling and shooting mechanism have been redesigned already. By making slight changes we were able to reduce the number of custom made parts, using readily available products instead. The chassis of the original robot contained a base-plate which required 40 hours of milling. By redesigning the entire robot-chassis such that it can be made out of bolted sheet metal we expect to be able to produce a chassis fifteen times cheaper (Fig. 2). We are also experimenting with web cams that could possibly serve within a low-cost omnivision unit.

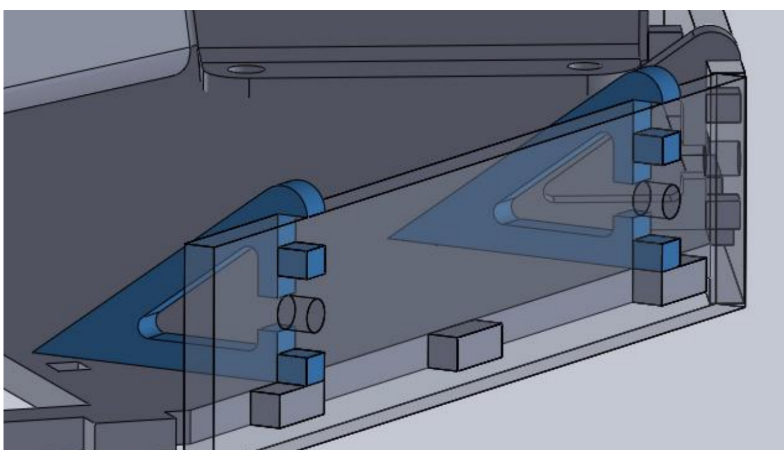

Fig. 2. Edge of base plate design, laser-cut sheet metal is bolted together.

Eindhoven University of Technology is joining forces with a group of high-tech companies. One of them will focus on hardware design and project managent[4], one will focuss on electronics design and low-level control[5] and a third company will focus on hardware realization and assembly.[6] Ten areas of functionality can be identified for a soccer playing robot: Move robot, handle ball, shoot ball, control robot, communication, structural integrity, protect robot, perceive environment, identification and manage power. Each partner in our consortium is responsible for a subset of these areas of functionality. Together we agreed to provide resources for a prototype that will be

[4] ACE, consultancy in engineering - http://www.ace.eu
[5] VEDS, electronic design solutions - http://www.veds.nl
[6] Frencken Group, global equipment service provider http://www.frenckengroup.com

build in the upcoming month. The prototype will be tested, evaluated and redesigned if necessary, such that a full team of robots can be presented at RoboCup Eindhoven this year.

4 Automatic referee

Contemporary middle-size league matches are presided over by a human referee. Automating her job is an interesting topic for research. For this year's scientific challenge we are aiming on demonstrating a real-time tool that is able to, at least, provide useful assistance to a human referee. As a first step we have implemented an autoref making decisions based only on positioning of ball and players (algorithm 1).

Algorithm 1 Auto referee

Initialize
while no human intervention **do**
  Filter top-cam image
  Object detection
  if ball crossed line **then**
    Refbox stop signal
    Broadcast decision autoref
    while positioning robots **do**
      Restart procedure
    end while
    Refbox start signal
  end if
  Update last-touched
end while

Building on previous work[7], we are able to easily obtain video frames from a top-camera placed above the field. To identify objects (Fig. 3), first the foreground and background are separated using the mean threshold method. Given is an input image I and a background model μ, containing averaged values over all pixels. In case $|I_k - \mu_{k-1}|$ is within a predefined threshold, the pixel is included in the background binary map F_b. If not, the pixel is treated as foreground F_f. Because in some cases background pixel values can change over time, e.g. due to lightning conditions, average values are updated on-line (Eq. 1).

$$\mu_k = [\alpha_h I_k + (1 - \alpha_h)\mu_{k-1}]F_b + [\alpha_l I_k + (1 - \alpha_l)\mu_{k-1}]F_f \quad (1)$$

Tunable parameters α_l and α_h are learning rates corresponding to the foreground and background respectively (where $0 < \alpha_l << \alpha_h << 1$). A sufficiently high α_l assures that initial mistakes in the background model are corrected.

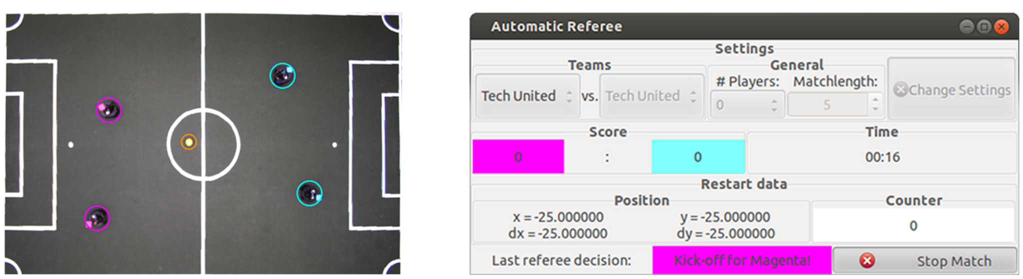

Fig. 3. Object-detection out of top-cam image (left), graphical user interface for autoref (right).

After objects are separated from the background, they are identified using a predefined color range for the ball and robot markers. Based on their position, autoref is able to observe events like corners, goals or the ball going out of bounds.[8] If such a referee event is detected, a stop signal is communicated to the robots, the referee decision and restart point are shown in a graphical user interface (Fig. 3) and a restart procedure is entered. Since the restart point is known by autoref, the restart procedure will wait until the ball is put at the restart spot. Hereafter, it will allow the robots some time to take their position.

[7] Tech United wiki, Greenfield Augmented Reality - http://goo.gl/oMC9v

[8] Video if first trial: http://www.youtube.com/watch?v=9bUeItsAFXI

5 TURTLE improvement

In parallel to our work described in the previous two sections, we have been improving our own robot soccer system. Our focus was on improving cooperative team play (subsection 5.1, 5.2 and 5.3) and on improving ball tracking in three dimensions (subsection 5.4).

5.1 Redesign of ball handling mechanism

In the past our ball handling mechanism was optimized for dribbling and designed to be as robust as possible in case of collisions. The system performs pretty well but some improvements were desired. For example the catching angle, i.e. the angle of the approaching ball with respect to the heading of the robot, was quite narrow. A small misalignment or misplacement of the robot while catching a ball would cause it to fail. Also improving the ability to capture a ball from a scrum situation was desired.

A new ball handling design has been developed and tested. Major changes are the relocated position of the wheels with respect to the ball (higher and further apart), and the pivot point of the levers is placed higher (Fig. 4).

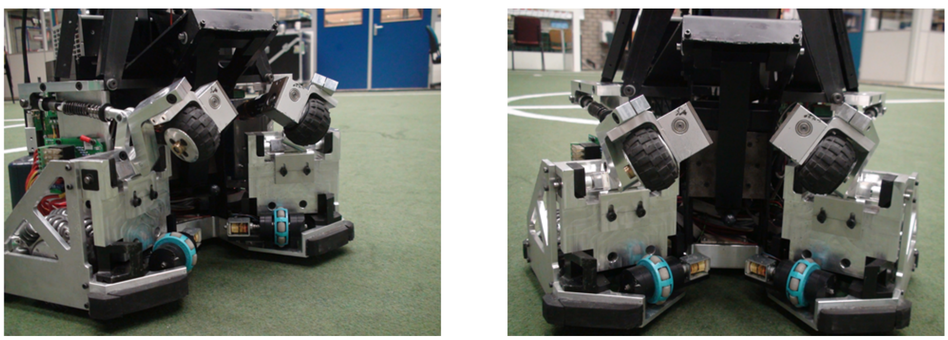

Fig. 4. Redesigned ball handling mechanism seen from different angles.

The new design outperformed the old one during a number of experiments. First the ability to catch balls arriving misaligned with respect to the robot heading was tested. The new mechanism is able to catch more balls successfully both when arriving with a misaligned angle or when arriving with a positional offset (Fig. 5).

Hereafter the ability to win scrums was compared to the old mechanism. Two situations were tested with approximately 25 trials each. In dual team mode, two robots autonomously chased the ball, the one with the redesigned ball handling mechanism was able to win 68% of all scrum situations. In an experiment involving only artificial scrums, created by pushing two robots together in demo mode with only the ball handling mechanism working, the redesigned system won 80% of all scrums. The wider configuration of the wheels in the new design is the reason of this improved scrum-performance, since the robot keeps in touch with the ball longer while rotating around its vertical axes.

5.2 Pass strategy

When obligatory passing through the middle line of the field was introduced in 2012, we implemented a two-robot strategy. The robot in possession of the ball would continuously choose between a shoot, dribble or pass action, while a second robot continuously tried to find an open space with clear line of sight to the robot with the ball. But many defending strategies currently locate more than one robot around the opponent with ball, creating opponent-free areas that could

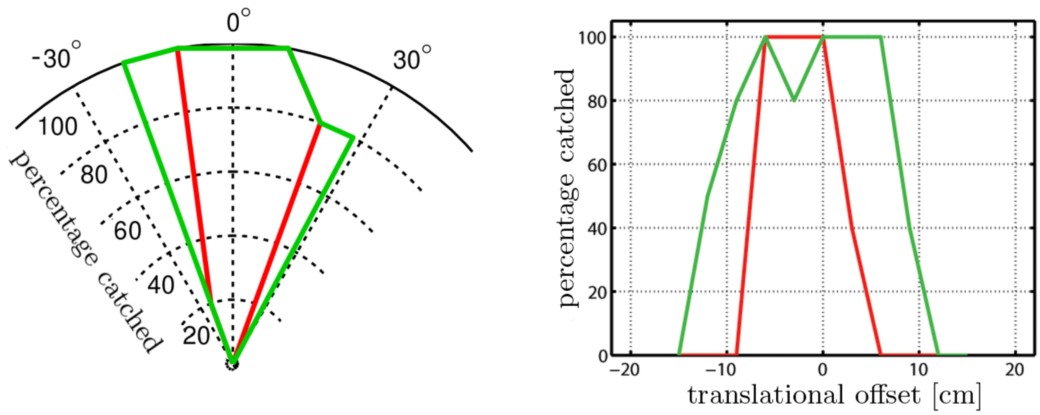

Fig. 5. Experimental comparison between old ball handling (red) and new ball handling mechanism (green). Each data point was created by rolling five balls towards the robot, providing guidance to the ball such that the same offset was reproduced.

easily be exploited. Hence, we are now implementing a three-robot strategy. Making use of passing not only in mandatory situations but also to take advantage of open space, putting ourselves in positions more suitable to score.

As a design case, consider a game situation with robot 1 in possession of the ball but being covered by several opponents (Fig. 6). In the current strategy robot 3 would try to locate itself such that is has clear sight to its teammate, robot 1. However the line between the ball and robot 3 can be easily be covered by the opponents. To increase the chances of scoring, in the new strategy robot 1 can actually pass first to robot 2. Simultaneously, robot 3 will move towards the open space which is a spot with free line to robot 2 and to the opponent goal. After robot 2 receives the ball it can pass to robot 3 who has a great chance to score.

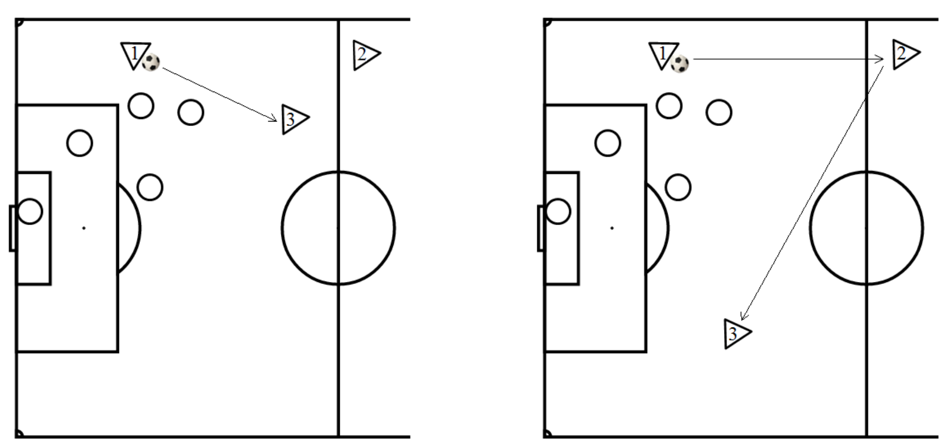

Fig. 6. Game situation to exploit in-game passing.

5.3 Ball push function

Two disadvantages of an active ball handling mechanism are: (i) Intercepting, aiming and shooting towards a certain target takes some time and (ii) intercepting the ball fails in case of inaccurate or late estimation of the position and velocity of the ball.

Therefore an alternative that is faster, while also more robust for measurement errors, has been developed. A push action will give the robots the ability to hit the ball using its housing. The angle of reflection will be controlled such that the ball bounces off in a desired direction (either towards the goal, a teammate or an open space on the field). In the upcoming paragraph we briefly describe the algorithm we use to do so.

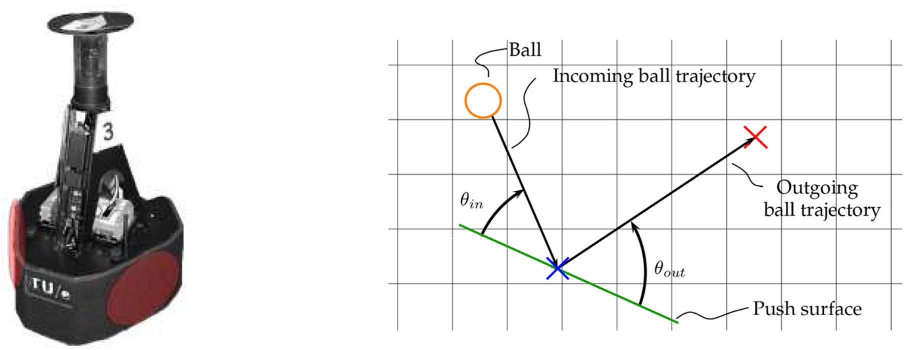

Fig. 7. Red indicates push surfaces (left) and schematic representation of a push action (right)

First, the trajectory of the ball has to be known, this is already being calculated and thus known in the current software. Second, a range of possible locations on the trajectory of the ball, which can be reached in time by the robot, will be calculated. Third, the optimal target, to reach a certain goal (as in scoring, passing, etc.), that can be achieved from any of these locations within this range is determined using potential fields. Next, we use a physics model based on the 'coefficient of restitution model' [2] to predict how the ball will bounce of the robot.

Now we are able to calculate the required position, velocity and rotation of the robot at the instance of interception. With this information we invoke a polynomial trajectory planner [3]. This hypothetical push action will be graded based on: A function of the variables involved and/or a learning algorithm that estimates the probability of success. If the grading is sufficiently high, the push action will be executed. While executing the calculation steps will be repeated to check if the action is still viable.

5.4 Kinect enabled ball detection

Ball detection on field-players is handled by the omnivision module. Since this module only provides a two dimensional image of the environment of the robot, possibilities of detecting balls in air are limited. An advantage of a depth-based camera (e.g. the kinect camera), is that it is equipped with two sensors: The first one captures a full-color image, the second camera provides a three-dimensional representation of the environment by looking at a laser-grid, emitted by the device itself. Combining both images provides enough information to distinguish a ball from its surroundings (Fig. 8)

Our algorithm[9] uses color segmentation to distinguish a ball from its surroundings, so pregame color calibration is required. After clustering using a k-means algorithm, a set of possible ball candidates is obtained that can be evaluated using information out of the depth image. We calculate the position of all ball candidates by averaging the position of the individual pixels in

[9] Tech United wiki, 3D Ball Recognition with Kinect http://goo.gl/DDTNZ

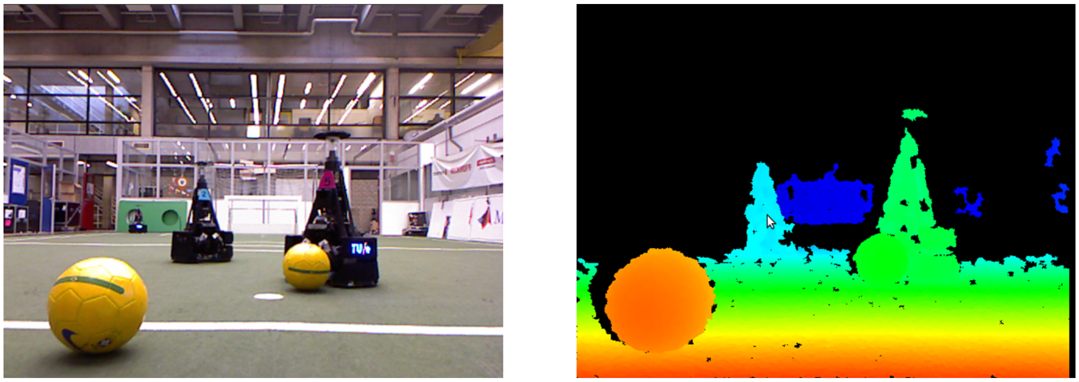

Fig. 8. Color image (left) and depth image (right) provided by kinect camera.

each cluster. Using this position, ball candidates outside the field can be discarded. Moreover, the size of the clusters can be compared to the expected size of a ball at this distance, which gives an indication for the reliability of the ball candidate.

At the moment, the kinect sensor is only implemented on our goalkeeper. Main drawbacks here are limitations in resolution and angle of view. Therefore only balls within approximately six meters can be recognized, which limits the available time for the goalkeeper to react when a shot is fired. We are planning to add kinect sensors to all robots, such that they are able to assist the goalkeeper while tracking balls in flight.

6 Conclusions

In our team description paper we have presented concrete steps towards the creation of a low-cost robot platform for the middle-size league. Such a robot facilitates knowledge sharing and could lower the hassle new teams might experience when starting from scratch. Also a blueprint for an autonomous referee system has been presented. Progress in hard- and software of our own TURTLE robot will lead to better passing and therefore more dynamic team-play. Which makes matches more exciting for spectators, but also opens doors towards more advanced levels of multi-agent coordination.

References

1. J.J.T.H. de Best, M.J.G. van de Molengraft, and M. Steinbuch. A novel ball handling mechanism for the RoboCup Middle Size League. *Mechatronics*, 21(2), 2011.
2. Salah Faik and Holly Witteman. Modeling of impact dynamics: A literature survey. *International ADAMS User Conference*, 2000.
3. Y. Guan, K. Yokoi, O. Stasse, and A. Kheddar. On robotic trajectory planning using polynomial interpolations. *Robotics and Biomimetics (ROBIO) IEEE International Conference*, 2005.
4. D. Jansen and H. Buttner. Real-time ethernet the EtherCAT solution. *Computing & Control Engineering Journal*, 15(1):16–21, Feb.-March 2004.
5. J.J.M. Lunenburg, M.J.G. van de Molengraft, and M. Steinbuch. ROP: developing a robotic open platform. http://goo.gl/Ljqls/, 2012. Presented at the 31st Benelux Meeting on Systems and Control.
6. J.J.M. Lunenburg, S. van den Dries, J. Elfring, R.J.M. Janssen, J.H. Sandee, and M.J.G. van de Molengraft. Tech United Eindhoven Team Description, 2012. RoboCup@Home.
7. K.J. Meessen, J.J.H. Paulides, and E. Lomonova. A football kicking high speed actuator for a mobile robotic application. *Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society*, pages 1659–1664, 2010.
8. Manuela Veloso and Peter Stone. RoboCup Robot Soccer History. http://goo.gl/10oSS, 2012.