Abstract. This paper describes the research improvements in the mechanical, electrical and software design of the robots of team Tech United. The main improvements are an automated calibration procedure, active ball handling, a new solenoid kicker design and an active keeper.

Keywords: Robotics, RoboCup, Autonomous Systems, Image Processing, Motion Control

1 Introduction

Tech United is the RoboCup team of the Eindhoven University of Technology, participating in the Middle Size league since 2005. Team Tech United consists of MSc, BSc, PhD students and staff members from the Eindhoven University of Technology.

This team description paper is based on the status of Tech United in January 2008 as a part of the Qualification package for the RoboCup World Championships 2008 in Suzhou, China. This paper describes the most significant advancements achieved in the past year.

First, a brief introduction of the Tech United robot platform is presented. Next, the main improvements compared to [5] are described, namely: (i) automated calibration, (ii) realtime localization and object detection, (iii) motion path generation and control, and (iv) developed mechatronic aids.

2 Robot Platform

Development of the Tech United robot, the so-called TURTLE (Tech United RoboCup Team: Limited Edition), was started in 2005. A picture of one of the third generation robots is shown in Fig. 1. The main improvement in the mechanics is the placement of several parts at a lower position in the robot, which lowers the center of gravity and enables a higher acceleration. For data acquisition and motion control, the robot is equipped with EtherCAT devices [6, 8] which are connected to the host computer via ethernet. Power is supplied by two Makita 24 V, 3.3 Ah batteries. The Maxon motors are driven by Elmec Violin 25/60 amplifiers. Furthermore, a capacitor of 350 V with a capacity of 4.7 mF is installed for the solenoid shooting mechanism. Each robot is equipped with a notebook running a preemptive Linux kernel. The robot software is implemented in the Matlab/Simulink environment and is built from Simulink models automatically via the RTW toolbox. In this way, a modular software framework is obtained.

3 Integration of Essential Tasks

In Fig. 2, a schematic overview of the robot’s hardware and software components and their interconnections is shown.

Ongoing optimization of the design leads to the renewal of several software and hardware components each year. In the next sections, innovations in a number of components are presented: (i) automated calibration, (ii) realtime localization and object detection, (iii) motion path generation and control, and (iv) innovative mechatronic aids.
4 Automated Calibration

As shown in Fig. 2, several components require offline calibration to work properly. Calibration is crucial for vision and localization and have to be redone when environmental conditions change, such as: different soccer fields, changes in lighting conditions and hardware tolerances of the robot.

The calibration procedure should be robust against environmental variations, it should be fast and simple in that no calibration expert is required to operate it. The present algorithm relies on easily recognizable features in the image.

4.1 Procedure

The automated calibration process consists of the following steps:

1. Automatic shutter time adjustment for the camera.
2. Capturing of a photo with the new shutter settings.
3. Automatic creation of a mask to eliminate useless parts of the image.
4. Calibration of the mirror and the compass, which is required for the mapping from robot localization.
5. Automatic color segmentation of the ball and the obstacles.
6. Processing of other images to make both the compass and the color calibration robust against variations across the field.
7. Synchronization of the calibration data to the computer on the robot.

The total automatic calibration process for a single robot requires about 1 minute which is considered fast in comparison to approximately 15 minutes for manual calibration by an expert user. Most parts of [1] are integrated in the automated calibration process. The main improvements are presented below.

4.2 Automatic Shutter Time Adjustment

The overall light intensity between soccer fields generally varies heavily, but it is approximately constant for one specific soccer field. A proper exposure of an image is essential for object detection, hence, calibration of the shutter time is required for each different soccer field.

The main color of the soccer field, i.e. green, is assumed to be dominant in the image. The average value of the green color near the robot is determined and used in a feedback loop to control it towards a desired set point of the intensity of green as shown in Fig. 3. The output of the controller is the shutter time. On average, the shutter time value has settled after two seconds.

![Fig. 3. Control scheme of the automatic shutter time adjustment.](image)

4.3 Automatic Mask Creation

A mask is required to remove unnecessary information in the omni vision image such as the robot itself, the bars that position the mirror and the image data outside the mirror. The most important part in creating this mask automatically is to detect both the center and the front of the robot. A white circular marker is placed on top of the camera as shown in Fig. 4. Furthermore, on this white circular marker a black line is placed to indicate the front of the robot.

To detect the white circular marker, edge detection using the Canny method [4] is carried out. The most round convex hull with a proper size and a dark center is considered to be the center of the robot. In practice, this has shown to work robustly. To find the exact center and front of the robot a local optimization is performed where the center of the robot \( (p_x, p_y) \) and the radius of the white circular marker \( p_r \) are determined. After that, the darkest peak on the radius \( p_r \) is detected and saved as the front of the robot. The three bars, shown in Fig. 4, are detected by collecting the intensity of the image of points on a spiral and cross correlating this with a pulse train containing three pulses per circumference. From this, the offset is captured and the bars can be masked.

4.4 Line Points Detection

For automatic mapping of the image to the field as described in [1] as much as possible line points have to be detected and the erroneous line points have to be reduced to a minimum. The first step is to detect the borders of the field. These borders of the soccer field are determined by looking for
colors close enough to green by searching in radial direction from the outside towards the center. The result for a typical calibration photo is shown in Fig. 5.

The next step is to detect white line points. Due to the large contrast between green and the white lines, second derivative based edge detection is suitable [4]. This is done by scanning in 100 radial directions. Due to noise in the image a smoothing operation is necessary. Both filter operations are combined into a single filter operation.

4.5 Compass

In order to determine the orientation of the field, an electronic compass is implemented. Soccer fields can contain materials below the surface that disturb the earth magnetic field. By placing the electronic compass on top of the robot, these disturbances are typically in the order of ± 30° when moving across the field following from measurements at a number of soccer fields. For robot localization such disturbances are too large, however, for only determining the side where to score, such disturbances are no problem.

4.6 Object Detection

Our color segmentation approach is based on creating tight bounds in a 3D colorspace (e.g. Y-Cb-Cr) by using 3D convex hulls. The calibration data is saved in a 3D lookup table. The ball is detected using shape recognition similar as described in Section 4.3. Obstacles are detected by looking for colors in the image that are darker than the green color. For the selection of the ball colors and the obstacle colors, a 3D convex hull is computed. For the points of the 3D lookup table which lie in this 3D convex hull, a color label is assigned. The result is a fast color segmentation using a 3D lookup table which is robust against light variations over the field. In Fig. 6, an example is shown of such a 3D lookup table. Due to the trend in RoboCup to be less dependent on color segmentation, only orange and black still have to be calibrated.

At the start of the automated calibration, a number of images can be selected for which object detection will be carried out to make the object calibration more robust across the field. Also, the compass calibration is adjusted when these images are processed.

5 Motion Path Generation and Control

5.1 Path Generation

Encoders are mostly considered to be too inaccurate for self localization due to drift caused by e.g. wheel slippage [7] and numerical integration. The cause of wheel slippage can be divided into two

Fig. 4. An example of an omni vision image.
Fig. 5. Omni vision image, white line: estimated border of the soccer field, black stars: detected line points.
parts: (i) slow drift due to finite stiffness of the wheel contact surface with the ground, (ii) fast drift by applying a too high torque regarding the grip and the wheel load. Fast drift can be largely prevented by limiting the motor torques satisfying the robot’s limitations. Only if the robot is colliding with or pushing against other objects, fast drift can occur.

In [2], a computationally cheap algorithm is presented which generates a motion path complying with the robot’s physical limitations such as velocity, acceleration and jerk limitations in all directions.

5.2 Decoupled Motion Control

The radial displacement of the motor axles is measured by three encoders $E$ with a resolution of $n_c = 2000$ counts per revolution. They are connected to the omniwheels via a gearbox. The motor coordinate frame based on the omniwheel displacement $q_i$ is defined as $m_x = (q_1, q_2, q_3)^T$. For direct motion control of the robot, a coordinate frame with respect to the orientation of the robot is more convenient. Therefore, the local coordinate frame $\ell_x = (x_\ell, y_\ell, \phi_\ell)^T$ as depicted in Fig. 7 is introduced. The transformation matrix relating the displacements applied to the local frame $\ell_x$ to the corresponding displacements in the motor frame $m_x$ is given by $T^{mf}$ while the transformation matrix relating torques applied in the motor frame to corresponding forces in the local frame is denoted by $T^{f\ell}$. These transformations allow for motion control design in a decoupled manner, i.e., instead of designing a MIMO controller in the frame $m_x$, it now suffices to design 3 SISO controllers for the decoupled frame $\ell_x$, see Fig. 8. The local frame $\ell_x$ is well suited in case the robot
is controlled by a manual remote control device such as a joystick. However, during autonomous
game play, it is more convenient to be able to prescribe trajectories with respect to an absolute
frame which is related to the layout of the field, \( f_x = (x_f, y_f, \varphi_f)^T \). The robot’s absolute pose on
the field cannot be determined from odometry and is obtained via vision \( v_x = (x_v, y_v, \varphi_v)^T \). A
simple solution would be to reset the pose based on odometry to the actual pose obtained from
vision every time a vision update is available \( t_x = f_x = v_x \). However, since motion control is done
in a collocated manner, this would result in frequent discontinuities in the control loop signals,
which is undesirable. Therefore, a much better solution from a motion control point of view is to
map the pose in the local frame \( t_x \) to the absolute frame \( f_x \), determine the robots’ targets in the
absolute frame \( f_x \), and transform these targets back to the robots local frame \( t_x \) which is then used
as the set-point for the decoupled control loop from Figure 8. As long as no vision-update becomes
available, the local and absolute coordinate frames coincide, i.e. \( t_x = f_x \). When a vision-update is
available, the absolute frame is reset to coincide with the actual pose obtained from vision, and
from this moment on ideally \( t_x = v_x \). However, due to drift, the field frame \( f_x \) moves with respect
to the actual position \( v_x \) and when the difference exceeds a certain off-set, again a vision update
takes place.

6 Innovative Mechatronic Aids

Three innovative mechatronic aids have been developed:

1. The active ball handling mechanism improves the ability to receive a ball and keep the ball in
   front of the robot while driving. A new feature is to drive backwards while keeping the rolling
   ball in front of the robot.
2. A solenoid kicker device enables variable shooting power. Furthermore, a mechanical switch
   that switches between shooting straight or in the air enables a variety of shots.
3. The keeper is equipped with an extra ball stopping mechanism that can move to the left or
   right when a ball is approaching the goal.

In the next sections, these mechatronic aids are presented in more detail.

6.1 Active Ball Handling Mechanism

The ball handling mechanism of the TURTLE platform is shown in Figures 9 and 10.

![Fig. 9. Top view ball handling.](image)

![Fig. 10. Side view ball handling.](image)

The main part of the ball mechanism consists of two levers. At the end of each lever a wheel is
mounted. These wheels are actuated by DC motors and the velocities of these wheels are measured
by tachos. The levers can rotate around fixed points of the robot and the angles of the levers can be
measured with potentiometers. The control architecture that is used is a hierarchical one. On the
low level, it consist of two velocity control loops to track a motor velocity reference and account
for disturbances acting on motor level. On the high level, the control architecture contains two
position control loops, one for each lever, to control the angles of the two levers. All controllers
are SISO based controllers. This can be done only if the levers are placed under an angle of approximately $90^\circ$ with respect to each other. In such a way we effectively create a decoupled system valid within the frequency range of interest. A preferred distance from the ball to the front of the robot can be defined, which results in preferred angles of the two levers. If the levers are bending forward an position error is introduced which is controlled towards zero by adjusting the velocities of the wheels. If the levers are bending backward the wheels will adjusted such that the ball is more or less pushed away from the front of the robot. Since, the robot also has to move, additional effort is necessary in order to maintain the levers at the preferred angle during these movements. This issue can be tackled by using feedforward. The input for the feedforward is the velocity (translational as well as rotational) of the robot itself. The active ball handling mechanism is superior to the commonly used passive ones for multiple reasons.

- It introduces the opportunity to drive backwards while still possessing the ball.
- The position of the ball with respect to the robot can be adjusted. This property can be exploited for example to produce special kicks, e.g. under an angle.
- Dribbling with the ball becomes much simpler since it is not necessary to constantly rotate around the vertical axis of the ball.

During all moves, the ball will keep rolling in a natural way.

### 6.2 Solenoid kicker

The kicking device is based on an electromechanical solenoid, in which a metallic plunger is moved via a magnetic field generated by a current flowing through a coil [3]. The plunger of the solenoid is pushed against a metallic leg to kick the ball, as shown in Fig. 11. The leg consists of two parts, an inner and an outer leg. The inner leg produces a straight, flat shot. By turning the plunger over $90^\circ$ the rectangular shaped top will push both the inner and the outer leg, the latter having a foot-like shape, thus producing a lob shot. The solenoid is powered by a 350 V, 4.7 mF capacitor, which is charged using the main batteries.

The shot timing and type are determined by the game situation extracted from vision software. When a shot is to be performed, first the switching solenoid (see Fig. 11) is actuated to get the desired shot type. After a specified delay, the capacitor is discharged over the coil through an insulated gate bipolar transistor (IGBT) to move the plunger against the leg and shoot the ball.

By applying pulse width modulation (PWM) on the control input of the IGBT, the time the capacitor discharges over the coil can be adjusted. This makes it possible to vary the end speed of the solenoid $v_{\text{end}} \in [0.5, 9.0] \text{ m/s}$. The adjustable shot can for example be employed to perform a pass between two robots dependent on their mutual distance. Another application is to vary the speed of a lob shot on goal based on the distance of the robot to the goal.
6.3 Active keeper frame

All robots are limited to a maximum static size of $50 \times 50 \times 80$ cm. For the keeper it is allowed to instantaneously increase its size temporarily in one of the three directions during a period of maximally 1 second [9]. When properly used, this instantaneous increase of size can stop a ball from entering the goal. Therefore, an active module is added to the keeper robot, as shown in Fig. 12. The keeper module consists of a static frame, which covers the maximum allowable static area. A movable frame is attached behind the static frame. The movable frame can be translated by a motor to the left or right side over a distance of 10 cm. The direction of movement and the timing are determined in software using a prediction of the ball position and velocity obtained from vision. The keeper module is only activated when the ball moves in a specified area on either side of the keeper robot and if the ball has a velocity component in the direction of the goal. The timing of the extension is crucial since the extension of the dimension is allowed for only one second. The actuation moment is determined by estimating the time it takes for the ball to reach the keeper.

The active keeper module has proven to be able to effectively stop a number of shots at goal during several games in the Robocup World Championships 2007.

7 Conclusions

Compared to the situation described in the previous team description paper, Tech United has advanced significantly. The automated calibration significantly reduces the calibration time from approximately 15 minutes to 1 minute while improving robustness. The improved ball-handling mechanism facilitates dribbling and interception skills, while the active keeper frame will stop shots at the goal. All new developments together should yield an improved game performance, at least matching and hopefully improving last year’s results in the Middle Size league.

References