Tech United Eindhoven Team Description 2020

Peter van Lith, Wouter Houtman, Ainse Kokkelmans, Thijs Ghering, Simon Geerts, Wouter Kuijpers, Koen Meessen, Yanick Douven, Ferry Schoenmakers, Dennis Bruijnen, Wouter Aangenent, Jorrit Olthuis, Stefan Kempers, Mathijs Schouten, Ruben Beumer, Johan Kon, Alberto Nunez Rodas, Franca Somers, Joep Selten, Robert de Bruijne, Matthias Briegel, Harrie van de Loo and René van de Molengraft

Eindhoven University of Technology,
De Rondom 70, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
techunited@tue.nl,
Home page: www.techunited.nl

Abstract. The Tech United Eindhoven Middle-Size league (MSL) team achieved a first place in RoboCup 2019 for the fifth time. To ensure a sixth first place in RoboCup 2020, the team has made a considerable amount of developments. For the sake of qualification for RoboCup 2020, this paper describes the most notable of those developments.

Keywords: RoboCup Soccer · Middle-Size League · multi-robot · machine learning

1 Introduction

Tech United Eindhoven represents Eindhoven University of Technology (TU/e) in the RoboCup competition. The team started participating in the Middle-Size League (MSL) fourteen years ago and has achieved a spot in the final for twelve years in a row, while achieving the first place five times: in 2012, 2014, 2016, 2018 and 2019. At the moment of writing, the MSL team consists mainly of PhD and MSc students, staff members and alumni of the TU/e.
This paper describes the major scientific improvements of the team over the past year. First, in Section 2, an introduction to our fifth generation soccer robot is given. The latest developments in the control of the eight-wheeled robot are presented in Section 3. Section 4 describes our efforts to use an artificial neural network to describe opponent data and learn from game situations. Section 5 continues with our implementation to improve on shooting accuracy using machine learning. Section 6 presents the lessons learned from the implementation of the Skills, Tactics and Plays architecture in the Middle Size League. Section 7 gives our concluding remarks.

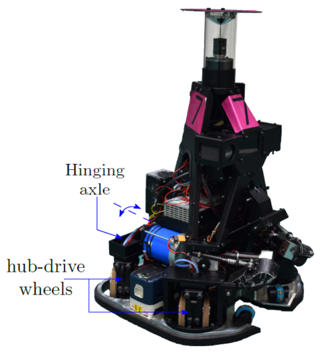
2 Robot Platform

Our robots have been named TURTLEs (acronym for Tech United RoboCup Team: Limited Edition). Currently, the fifth generation hardware TURTLE is used, while the sixth generation is being developed. The fifth generation TURTLE has evolved over the years into the robust platform we use today by means of small modifications. The reader is referred to the second section of the description paper of 2014 [1] for the schematic representation covering the main outline of the robot design. A detailed list of hardware specifications, along with CAD files of the base, upper-body, ball handling mechanism and has been published on the ROP Wiki[1].

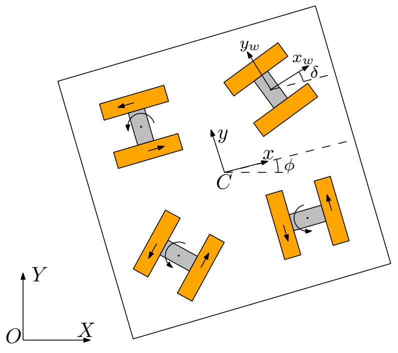
3 Eight-Wheeled Platform: Torque Distribution

Last year, the mechanical design of the eight-wheeled platform, shown in Figure 1(a), and its corresponding low-level control software were presented [2,3]. This platform consists of four wheel sets, each having two hub-drive wheels, thus being five times over-constrained. To avoid an over-constrained design, each wheel set has an internal degree of freedom. The last constraint is addressed by a hinging axle at the back of the robot. Each wheel set can rotate around its suspension by actuating the corresponding wheels in the opposite direction. This is graphically shown in Figure 1(b). The main advantages compared to the three-wheeled platform are the possibility to apply the torque delivered by the motors in the desired direction in order to achieve a higher acceleration in this direction. The reasoning is based on the following: as a forward acceleration causes a reduction in the normal force on the front wheels, this limits the ability to apply torque from the motors on to the field for the three-wheeled platform due to slip-phenomena. Since the normal force is proportionally increased on the back wheels, in contrast to the three-wheeled platform, the eight-wheeled platform can maintain the ability to apply torque from the motors on the field. As the torque-distribution from the platform controllers to the wheels was still under research last year, this section elaborates on the implemented solution.
In order to manipulate the x, y, ϕ-position of the center of mass C of the robot, a translational controller and a rotational controller have been designed [3]. As there are more actuators than the number of degrees of freedom, the desired output of these controllers has to be distributed over the actuators. This problem is also known as the control allocation problem or torque vectoring. For this platform, this problem translates to determine the contribution of each actuator, the eight wheels, to the control-effort desired by the platform-controllers given the given the orientation δ_i of each wheel set $i = 1, 2, 3, 4$. As the orientation of the wheel-sets is determined using the kinematics of the system, the control-allocator considers a single input to each wheel set. As a result, there are four degrees of freedom for the control-allocator and each actuation input to a wheel set is equally divided over its corresponding wheels.
Optimization-based approaches turn out to be powerful for control allocation problems [4]. A quadratic cost function is defined, constrained by the actuation limits of the wheels. The first objective in this optimization function is to minimize the difference between the desired control efforts and the control efforts

[1] http://www.roboticopenplatform.org/wiki/TURTLE



(a) Mechanical Realisation



(b) Wheel Configuration

Fig. 1. The eight-wheeled platform with four suspended wheel combinations which are able to rotate around their center hinge.

applied on the platform. On the contrary, as a second objective, the actuation-input to each wheel set is minimized. As slip is the root cause for not transferring the desired force on to the ground, in this optimization function, each wheel-input is weighted by the inverse of the normal force expected for each wheel set. This normal force is estimated based on a force and momentum balance of the system given the desired acceleration. In combination with selecting a tyre with more grip, this weighting at least doubled the achieved performance in terms of the acceleration achieved.

4 Understanding Team Behavior in MSL Competitions

Analyzing Team Formations
The goal of analyzing team formations is to learn a descriptive model of the behavior of the field players of a team. We want to use these models in a simulation environment, to find strong- and weak points in our strategy and that of our competitors. The source data is formed by log-files, recorded by our TURTLE robots during the competitions. These log-files have recorded the locations of all robots and the ball at a frequency of about 10 Hz. For our robots, roles and actions have been logged as well.

We take our inspiration from Futsal coaching and analysis techniques, as described in the FIFA and UEFA coaching manuals. In Futsal, team strategies are described visually in Set-Pieces (see Figure 2). Our objective is to learn behavior patterns on a symbolic level, expressed in the form of set-piece scenarios for a single agent. Afterwards, these scenarios form input to a neural network that will first learn the behavior of all robots of our team and later that of our opponents.

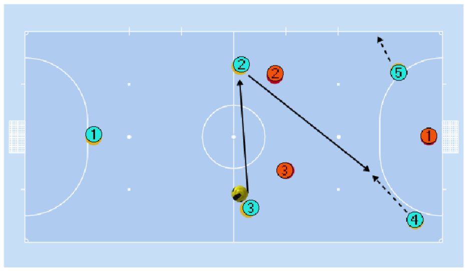


Fig. 2. Example of the set-piece in which the team attacks in a 1-2-2 formation. Set-Pieces are represented graphically in the Futsal coaching manuals, depicting the steps taken by each player for a given game situation.

Agent Role Identity
The analysis starts with selecting game episodes and determine the team formation. Episodes start and end with game turnovers (changes in ball possession). Within an episode, agents are given a unique identifier that is fixed for the duration of that episode. Using this identifier we can refer to a player by it's role in the current formation and facilitates generalizing game situations. We encode an agent's game situation in a code like AT-211 (Attack on Their half, 2 opponents in front and 1 opponent in the back). With that code we can then search for all similar game situations, and compare the actions that agents take, when in that position.

These scenarios form the ground truth and are input to a neural network that will learn to associate game situations with set-pieces.

Natural Language Processing (NLP) as a Basis
We describe the team formations during and episode in a six-position code, like AOW202 (Attack on Our half in a Wide formation with 2 attackers, 0 midfielders and 2 defenders).

In addition, we encode how this situation is handled. Such a description can be compared to a short story, summarized like: "Our team attacks in a tight 202 formation on our half. The opponent is defending in a 211 formation. Our agent in position 1 is moving towards the ball and intends to capture the ball."

Our goal is to use these game situation descriptions as input to a neural network, that will learn to associate a sequence of agent movements to a set of roles, intentions and actions. This work is still in an early stage. Tools have been built to perform the analysis. The information is visualized in a generated set-piece with the movements of all players during an episode and is used to explain the behavior of a single agent as shown in Figure 3.

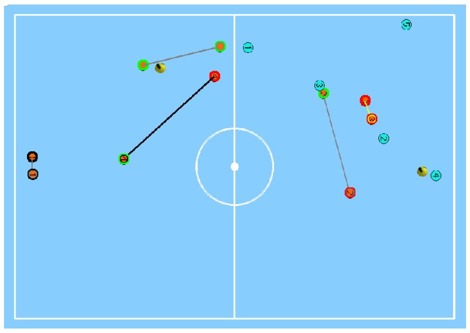


Fig. 3. Example of a generated set-piece. We are playing in a DTS202 formation. Agent 3 is facing 2 robots with 2 in the back (Agent formation AT-212). Agents 2 and 4 have switched from Attacker to Defender.

5 Improving the Performance of the TURTLE's Shot Using Machine Learning

When shooting the TURTLE's aim for a certain target $(y, z)_{target}$, not all shots hit the target, the variation in the error is discussed in [3]. In this section, a machine learning algorithm is proposed to reduce the bias of the shots.

The TURTLEs are equipped with a lever to be able to kick a ball. The height of that lever (L) can be altered and the amount of electrical energy used by the solenoid (K). Although the height of the lever will mostly affect the angle at which the ball leaves the ball handling (α_{real}) and the solenoid force will mostly affect the exit velocity (v_{real}), these are coupled. The mapping from K and L to α_{real} and v_{real} is denoted by *Ball Interaction* in Figure 4.

To be able to accurately reach the target, hence $(y, z)_{real}$ is equal to $(y, z)_{target}$, the *Inverse Interaction Mapping* has to be the inverse of the *Ball Interaction*, see Figure 4. A second-order polynomial was used to approximate the *Ball Interaction*, which can be used in the *Inverse Interaction Mapping*-block. The *Ball Interaction*-mapping might change through time and may differ per robot, the *Inverse Interaction Mapping* will therefore also have to change. Using the TURTLE's Kinect V2 camera as feedback, the control loop of the online calibration as described in Figure 4 updates the parameters of the *Inverse Interaction Mapping* (a). The Kinect V2 camera detects the ball at approximately 30 Hz during flight, a polynomial is fit through the trajectory to estimate the exit velocity of the ball (v_{est}) and the exit angle of the ball (α_{real})

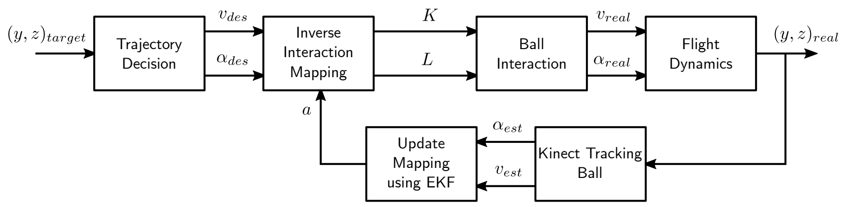


Fig. 4. A block scheme of the closed loop system of the online calibration.

The first results of the implemented algorithm are promising. In an experiment the TURTLE started with a wrong set of parameters and learned the correct mapping, while using random values for K and L. We find that the parameters of the map (a) have converged after 20 shots. Assuming the converged map to be the ground truth, thus the best fit through the *Ball Interaction*, the real exit angle of shot n in the experiment $(\alpha_{real,n})$ can be compared to the exit angle for the converged map $\alpha_{gt,n}$. The absolute value of the error in the latter comparison is plotted in Figure 5, showing that the map converges. A similar curve has been obtained for the exit velocity of the ball, this indicates that the bias in the error between the shot and the shot target can be removed.

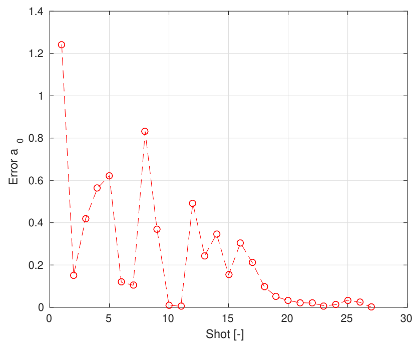


Fig. 5. Absolute difference between $\alpha_{real,n}$ and the value from the last iteration of the experiment, $\alpha_{gt,n}$, versus shot n.

6 Skills, Tactics and Plays

In [5] a pioneering collaboration between two robot soccer teams from different RoboCup leagues, the Small Size League (SSL) and the MSL was presented. One example implementation of an offensive team-level strategy from the centrally-controlled SSL of RoboCup was ported to the distributed MSL. In that example,

the Skills, Tactics, and Plays (STP) team-planning architecture from CMDragons from Carnegie Mellon University in Pittsburgh (United States) was successfully used in the Tech United team for one specific strategy. During the past three years this STP framework has been implemented in the Tech United software and since recently fully replaces the strategy framework that had been in use since 2011.
The STP architecture enables simple specification of team plans as Plays, which are then executed individually by each robot through Tactics and Skills. These elements can be shortly described as follows:
Skills are base behaviors that can be parameterized to achieve various goals. In the domain of robot soccer, these skills include navigating safely and quickly to a specific target location, shooting the ball with a specified velocity, and intercepting a moving ball, amongst others.
Tactics are goal-oriented behaviors that each robot performs to carry out a team plan. These behaviors may be composed of skills, organized to work cohesively towards achieving a goal.
Plays are team plans specified as a list of roles that the team of robots must fulfill. Each role, within the context of STP, is defined as a sequence of tactics to be completed sequentially.
In STP, team strategy is encoded into a playbook, made up of a set of plays, see [5] for more details. The goal of replacing the existing strategy architecture by the STP framework is twofold. Firstly, the original strategy software did not have a modular setup and it became more and more difficult to make changes. Secondly, and more importantly, the STP framework allows for the relatively easy and high-level generation of different strategies in the form of Plays. All already existing skills and tactics have been refactored into a structured code library. This enables us to quickly define a play via (at most) five roles by listing the set of tactics that that role should subsequently perform. Synchronization between roles is automatically taken care of in the STP framework. This not only allows relatively junior team members to be able to quickly contribute to the Tech United team by generating their own plays but also to change strategy during tournaments relatively fast. A high-level picture of the STP framework implementation is depicted in Figure 6.

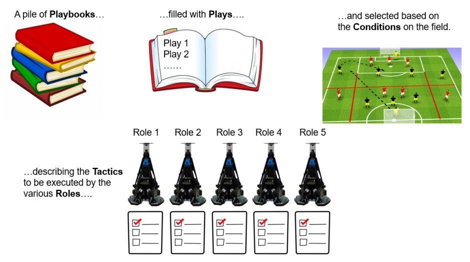

A pile of **Playbooks**…
…filled with **Plays**….
Play 1
Play 2
…and selected based on the **Conditions** on the field.
Role 1
Role 2
Role 3
Role 4
Role 5
…describing the **Tactics** to be executed by the various **Roles**….

Fig. 6. High level picture of the STP framework as implemented in the Tech United strategy software.

7 Conclusions

This paper has described the major scientific improvements of the Tech United soccer robots over the past year. Progress has been made on the control aspect of the eight-wheeled platform, bringing the switch to a full team of sixth-generation TURTLEs closer. To improve our strategy, a start has been made to analyze team formations to learn a descriptive model of opponent team behavior. For a more accurate and precise shot, a machine learning algorithm has been implemented. Our new Skills, Tactics and Plays (STP) framework implementation is finished by this year enabling faster adaption of our strategies during the next tournament. Altogether, we hope that our presented progress contributes to an even higher level of dynamic and scientifically challenging soccer competitions during RoboCup 2020 in Bordeaux, and brings us a step closer to our goal in 2050!

References

1. Lopez Martinez, C., Schoenmakers, F., Naus, G., Meessen, K., Douven, Y., Van De Loo, H., Bruijnen, D., Aangenent, W., Groenen, J., Van Ninhuijs, B., Briegel, M., Hoogendijk, R., Van Brakel, P., Van Den Berg, R., Hendriks, O., Arts, R., Botden, F., Houtman, W., Van 't Klooster, M., Van Der Velden, J., Beeren, C., De Koning, L., Klooster, O., Soetens, R., Van De Molengraft, R.: Tech United Eindhoven Team Description 2014. (2014)
2. Douven, Y., Houtman, W., Schoenmakers, F., Meessen, K., van de Loo, H., Bruijnen, D., Aangenent, W., Olthuis, J., de Groot, C., Farahani, M.D., van Lith, P., Scheers, P., Sommer, R., van Ninhuijs, B., van Brakel, P., Senden, J., van 't Klooster, M., Kuijpers, W., van de Molengraft, R.: Tech united eindhoven middle size league winner 2018. In Holz, D., Genter, K., Saad, M., von Stryk, O., eds.: RoboCup 2018: Robot World Cup XXII, Cham, Springer International Publishing (2019) 413–424
3. Houtman, W., Kengen, C.M., van Lith, P.H.E.M., ten Berge, R.H.J., Kon, J.J., Meessen, K.J., Haverlag, M.A., Douven, Y.G.M., Schoenmakers, F.B.F., Bruijnen, D.J.H., Aangenent, W.H.T.M., Olthuis, J.J., Dolatabadi, M., Kempers, S.T., Schouten, M.C.W., Beumer, R.M., Kuijpers, W.J.P., Kokkelmans, A.A., van de Loo, H.C.T., van de Molengraft, M.J.G.: Tech united eindhoven middle-size league winner 2019. In Chalup, S., Niemueller, T., Suthakorn, J., Williams, M.A., eds.: RoboCup 2019: Robot World Cup XXIII, Cham, Springer International Publishing (2019) 517–528
4. Johansen, T.A., Fossen, T.I.: Control allocation - a survey. Automatica **49**(5) (2013) 1087 – 1103
5. de Koning, L., Mendoza, J.P., Veloso, M., Van De Molengraft, R.: Skills, Tactics and Plays for Distributed multi-robot control in Adversarial environments. In: RoboCup Symposium 2017. (2017)