

# R-ZWEI KICKERS

## Team Description Paper 2023

Adrian Müller<sup>1</sup> and Andreas Hobelsberger<sup>1</sup>

<sup>1</sup> University of Applied Sciences Kaiserslautern, Amerikastr. 1, 66482 Zweibrücken, Germany  
rzwei.kickers@hs-kl.de  
www.hs-kl.de

**Abstract.** The R-ZWEI KICKERS, founded in 2020, established in less than three years a stable team, contributing actively to the field and having a positive impact on education and research at UAS Kaiserslautern. The foundation of our development is the B-Human 2021 code release, allowing us to quickly catch up with the league and make meaningful contributions. Our work focuses on behavior and communication optimization, as well as new team strategies and shooting techniques. We intend to use our approaches to at least participate in the RoboCup 2023 dynamic ball handling technical challenge, and the 2023 German Open Replacement Event.

**Keywords:** hybrid AI, audio communication, reinforcement learning, teach-in

## 1 Team Info

Name: R-ZWEI KICKERS  
Leaders: Prof. Adrian Müller (scientific lead)  
Andreas Hobelsberger (team captain)  
Contact: [rzwei.kickers@hs-kl.de](mailto:rzwei.kickers@hs-kl.de)  
Website: <https://smart-machines.hs-kl.de/projekte/hs-kl-robocup/>  
Country: Germany  
Affiliations: UAS Kaiserslautern, abat+ GmbH

## 2 Code Usage

The code we use to play soccer is largely based on the B-Human 2021 Code Release [1]. The behavior control architecture used by our team is directly from the B-Human code release. The actual behavior, implemented as cards, is custom-built by us. Furthermore, we have developed two novel approaches for defining behavior by extending this architecture. (see 3.2. Teach-In and hybrid AI)

Communication Protocols have also been overhauled by us, specifically the frequency with which messages are broadcast (see 3.2 Network data minimization).

## 3 Own Contributions

### 3.1. Algorithms, Architecture, Network

Most of our contributions described here are demonstrated in the video presentation.

**Strategy and dynamic role assignment.** R-ZWEI KICKERS are using a role-based behavior, based upon three player roles (GOALIE, DEFENSE, OFFENSE), three vertical offsets (left, middle, right) and four strategies (NORMAL, OFFENSIVE, DEFENSIVE, SPARSE). For example, while following the defensive strategy, three robots are assigned a DEFENSE role (left, middle, right), with only one offensive player, and vice versa in OFFENSIVE mode. In sparse mode (only a few players are active), long distance shots are the preferred behavior. If a goal is scored, or a bot is penalized, the roles are recomputed instantly, and the bots change their mode and default positions.

**Gameplay.** The robot closest to the ball is called a „striker“. It is responsible for handling the ball in most of the standard situations (e.g. kick-in). For the normal gameplay we developed ~10 prioritized behavior sets (according to the B-Human framework they are called “cards” - e.g. ChaseBallCard, ClearOwnHalfCard, GoalieDefaultDecard, etc.) Important cards are discussed below. A card becomes active according to the robot’s role, the game state and specific, relevant information (e.g. constraints to the x,y-position of the bot).

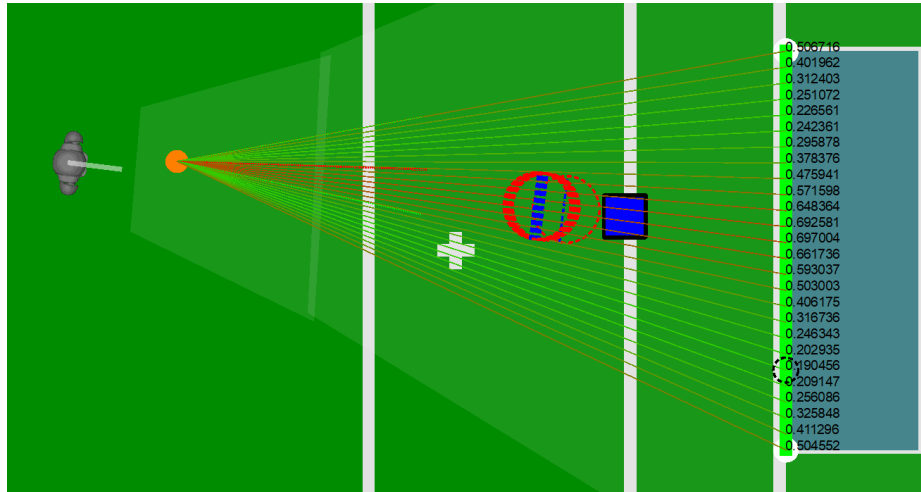
**Passing.** A robot in the back line that possesses the ball can pass it to a robot further down field. We compute and maintain the line-up of the bots on the x-axis (so-called “supporter index”) so we can identify passing and receiving robots. The passing robot signals its intent to pass over wireless communication. In reaction, the receiving partner stands still and awaits the ball.

**Shot predictor and Goal Shot.** To ensure an accurate shot on a target area, such as the goal, we have developed the Shot Predictor module. This module calculates the probability of a shot reaching its target for a range of potential shots. A shot is considered to have failed if it misses its target or hits an opponent robot.

To do this calculation, the standard deviation of shot angle and range are considered, which were determined experimentally. Using the robot detection provided in the B-Human 2021 code release, and the assumption that opponents will be moving towards the ball, the Shot Predictor can extrapolate the approximate location of opponents at the time when the ball would be kicked. A further factor in estimating time to kick is the distance of the striker to the ball.

Finally, the predictor stores the direction and power required for the optimal shot on a specific target. This data is locked before the shot is executed to avoid noise in the detection suddenly canceling the behavior.

Using this approach, we have seen an increased number of goals, occasionally “threading the needle” through multiple robots.



**Fig. 1.** A representation of considered shots by the Shot Predictor module. Shots which are likely to succeed are shown as green lines. Shots that are likely to fail are drawn in red. Red circles represent the predicted position of the opponent robot at kick time.

**Network data minimization.** Our reduction strategy on network data consumption is twofold: limited bandwidth for periodic information, combined with ad hoc broadcasts due to certain conditions.

The team strategy solely relies on input information from the game controller (penalties, score, game time, message budget). Thus, each bot re-computes the recent strategy in parallel, without consuming bandwidth. However, resulting changes in its role etc. are broadcasted immediately.

Certain conditions and changes like whistle detection, or possessing the ball (“becoming striker”) are sent with high priority. We will use our audio communication (see 3.2. Audio Communication) to further enhance the communication here. We leave it to the developer of behavior rules (cards) whether a change in behavior of a bot (e.g. entering another card) is broadcasted with high priority (e.g. the intent to pass).

Position information changes (own position, ball position) are sent periodically (round-robin) with low frequency. The base frequency is adjusted dynamically according to the remaining budget.

### 3.2. Experimental features

**Teach-In and hybrid AI.** Rule-based behavior control systems [2] make it necessary to define priorities, fine tune the thresholds in the entry and exit conditions, and finally to maintain the ensemble manually. To achieve scalability and robustness, we

follow a hybrid approach, combining normal, i.e. “rule-based cards” with so-called “teach-in cards”.

A teach-in card is recorded by a domain expert using a PS4 controller and a modified keylogger. This soccer instructor intervenes in a specific game situation, teaching a dedicated behavior (“playback sequence”) for one manually selected robot. The starting point of this sequence is called a “word-model”, its features contain the maximum of available information like position of robots, ball, game time, score, etc. The set of all world-model data is subject to an offline data mining analysis, which identifies the minimum subset of features which give a maximum of dissimilarity, the so-called “trigger points”.

During game play, the minimized world-models are matched to the current game situation, and the card with the best matching trigger point gets executed. Teach-in cards are mixed in between rule-based cards in the gameplay stack. This hybrid behavior computation results in an intelligent behavior of our robots by following a three-phases execution strategy:

1. Very specific rule-based cards (eg. “offense is close to goal → kick!”)
2. Teach-In card, the best card is selected dynamically by a breadth search
3. “Default” rule-based cards (eg. “return to home position”)

**Tactics optimization by Reinforcement Learning.** Currently, all tactical decisions made by the robots are based on rules written by a programmer, or stem from teach-in data. These rules, while expressive, are often the point of failure in the robot’s playing. In our experience, odd behavior of a robot during a match can typically be attributed to a bad logical rule or latency in communication between robots.

Our approach to solving this problem is to outsource tactical decisions to a reinforcement learning system. Using B-Human’s SimRobot as a base, we are able to simulate a randomized gameplay scenario in 2D and play through robot behavior at a faster than real-time speed. An external python script, containing the training environment, communicates with the simulation and reads the world state from each robot’s perspective. A recurrent neural network is then applied to predict the next best action (card) for the robot. The predicted action is injected in the simulated robot and executed. This system evaluates the changes in ball position and goals over the span of 30 seconds to determine a score for reinforcement learning. Then the game is reset, robot and ball positions are randomized, and the next training cycle begins [3].

*Further Development Plans.* The reinforcement learning system is currently only able to control simulated robots via a socket based communication link between the python script to SimRobot. We are currently working to convert the system’s pyTorch model to a TensorFlow lite model and run it on a live robot. Once a live system can exhibit simple behavior similar to that trained in the simulation, we will move to adding more complex actions to the set available for reinforcement learning.

**Audio Communication.** Due to further limitations on the communication between robots over the wireless interface, we found it beneficial to develop a means of communication over audio. The primary concept is to encode a few bits in a superposition of sine waves, then using a Fourier transform in the other robots to monitor for broadcasts. Each robot in the system is given a unique base frequency, bits are encoded by using frequencies offset from the base. The frequencies which one robot is allowed to broadcast do not overlap with the frequencies of other robots, effectively giving each robot its own “voice”. Each message that can be broadcast by a robot is then associated with a meaning, such as the intention to play a pass. The messages encoded by the protocol are primarily messages deemed important by an analysis done by B-Human [4].

The sounds generated by this protocol do not sound like speech and still transfer binary data. The objective of this feature is to prove that audio communication is a viable option for communication between robots.

We can estimate the maximum broadcast speed of the system, although exact values will depend on how well the protocol can be optimized. Assuming that each message sent will require 3 back-to-back positive results from the Fast Fourier Transform (FFT) and adding one additional sample on each size of the signal as a buffer, the message broadcast should last for 5 FFT cycles. The FFT operates at 10Hz, so the maximum broadcast frequency would be 2Hz, or two messages per second per robot. This is a 10x increase of the 0.2 Hz average wireless broadcast frequency allowed by SPL Rules. Using the audio protocol, we can create semaphore like behavior between robots without wasting wireless messages.

In the application video, we include a clip of robot 1 broadcasting the message “1”. In response, all other robots reply with their own “1” message. This is our proof of concept that binary data can be transferred using this protocol.

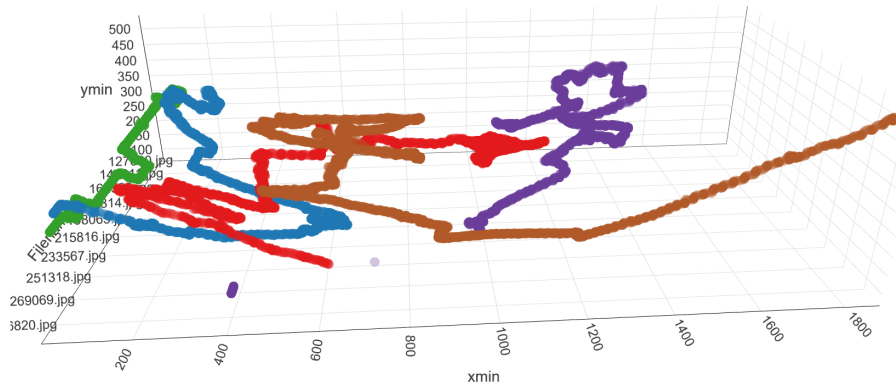
*Further Development Plans.* The audio communication system is still experimental, and thus is prone to false positives, and has a hard time detecting multiple signals from the same robot in quick succession. Following are our development plans to address these issues before the 2023 World Championship:

- A higher sample rate and input size to the Fourier Transform allows for more fine-grained signal detection, reducing the possibility for false positives.
- Using a more robust peak detection algorithm should further reduce false positives, as the current system simply checks for a threshold to be surpassed.
- Once false positives are reduced, the signal durations can be shortened to reduce noise on the field and allow for more signals to be sent.

## 4 Past History

So far, the R-ZWEI KICKERS have (partially remotely) participated in

- The German vRoHOW Workshop 2020
- The 2021 German workshop on image annotation
- The German Open Robotic Workshop RoDEO 2021 , playing 5vs5 remote, and conducted two local challenges on our own premises <https://rohow.de/2021/>
- The German RoboCup replacement in May 2021 in several 5vs5 remote games <https://www.robocup.de/en/2021/05/08/562/>
- The RoboCup 2021 – RoboCup Standard Platform League, in two local challenges (<https://spl.robocup.org/rc2021/>)
- GORE 2022, April, Hamburg where we entered the second round in the SPL ( [www.robocup.de/gore-2022/](http://www.robocup.de/gore-2022/) )
- The RoboCup 2022, remotely participating in the Video Challenge



**Fig. 2.** 3D scatter plot of our video tracking analysis, showing the routes of five robots over game time. RoboCup 2022, Video Challenge

## 5 Impact

The R-ZWEI KICKERS team is part of the Smart Machines working group [5] at UAS Kaiserslautern. We take an agile approach to development. Students from all faculties are invited to join [6], and we regularly integrate project work and theses of students into our code base and future plans. We showcase our work for groups of pupils, freshmen, and visitors [7].



For higher semester students, we offer courses on AI and Robotics [8]. In Summer term '23, we will integrate two foreign students from Presidency University, India, into our team.

All team members are encouraged to join SPL workshops, challenges, and

tournaments. Our industrial partner abat+ partially sponsors our participation in these events. The major impact of our joint work is the successful identification of technical questions (e.g. vision, simulation, HCI, ...). We assign students to projects within SPL, ending with a transfer phase to the industrial view on how to apply the solutions in practice.

## 6 Previous Work

In 2020 - 2022, our work focussed on Q-learning, object recognition, path planning, image classification and data augmentation. The authors like to thank Connor Lismore, David Kostka, Felix Mayer, Thomas Jäger and many more, unnamed students for their valuable contributions.

## 7 References

1. B-Human Code Release 2021  
<https://github.com/bhuman/BHumanCodeRelease/tree/coderelease2021>, last accessed 2021/02/15
2. Russell, Stuart J.; Norvig, Peter: Künstliche Intelligenz. Ein moderner Ansatz. (2012)
3. Lismore, C.: Research and Implementation of Unsupervised Learning for Singular Agents in a Team-Based Environment, Master Thesis, 65–66 (2022), UAS Kaiserslautern
4. Thomas Röfer, et al.: Team Report and Code Release 2022, 17  
<https://github.com/bhuman/BHumanCodeRelease/blob/coderelease2022/CodeRelease2022.pdf>, last accessed 2021/02/15
5. Team Page on <https://smart-machines.hs-kl.de/projekte/hs-kl-robocup>, last accessed 2021/02/15
6. Our public Wiki and git repository: [https://zwoogle3.informatik.hs-kl.de/rzwei\\_kickers](https://zwoogle3.informatik.hs-kl.de/rzwei_kickers), last accessed 2021/02/15
7. Our past and present public events  
<https://smart-machines.hs-kl.de/veranstaltungen/liste/?eventDisplay=past>, last accessed 2021/02/15
8. Müller, Adrian “Robotics”, Lecture and module description, 2022,  
[https://campusboard.hs-kl.de/portalapps/sv/Modul\\_17412.pdf?modulid=17412&stgid=460](https://campusboard.hs-kl.de/portalapps/sv/Modul_17412.pdf?modulid=17412&stgid=460), last accessed 2021/02/15