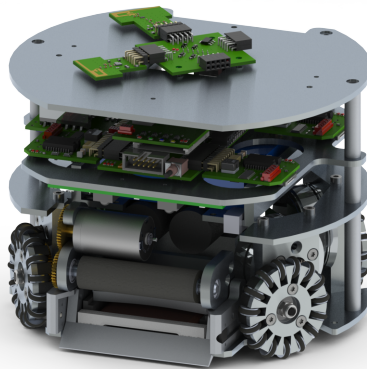


# ER-Force 2018 Open Source Award

Michael Niebisch, Andreas Wendler, Tobias Heineken, Alexander Danzer and  
Michael Eischer

Digital Sports, Pattern Recognition Lab, Department of Computer Science  
Friedrich-Alexander University Erlangen-Nürnberg (FAU), Erlangen, Germany  
Robotics Erlangen e.V., Martensstr. 3, 91058 Erlangen, Germany  
Homepage: <http://www.robotics-erlangen.de/>  
Contact Email: [info@robotics-erlangen.de](mailto:info@robotics-erlangen.de)

**Abstract.** This paper presents proceedings of ER-Force, the RoboCup  
Small Size League team from Erlangen located at Friedrich-Alexander-  
University Erlangen-Nürnberg, Germany.



## 1 Introduction

For multiple years we have open sourced our software framework and updated it continuously with our newest improvements.

## 2 Project

Our framework has been published under the GNU GPLv3 license and can be accessed from our GitHub page under following link: <https://github.com/robotics-erlangen/framework>

Ra is both a simulator and framework for the full usage during a RoboCup SSL competition game. It can be used to simulate whole games running two strategy's with built in RefBox and Autoref, which immensely helps the development of new features.

The same configuration can also receive vision packets from the SSL Vision, trigger the strategy and send RF packets to the robots.

## 3 Documentation

An extensive description of most features can be found in the README.md provided with the repository.

Features include a full graphical client for configuration and testing, a plotter and logplayer.

## 4 Innovation and Features

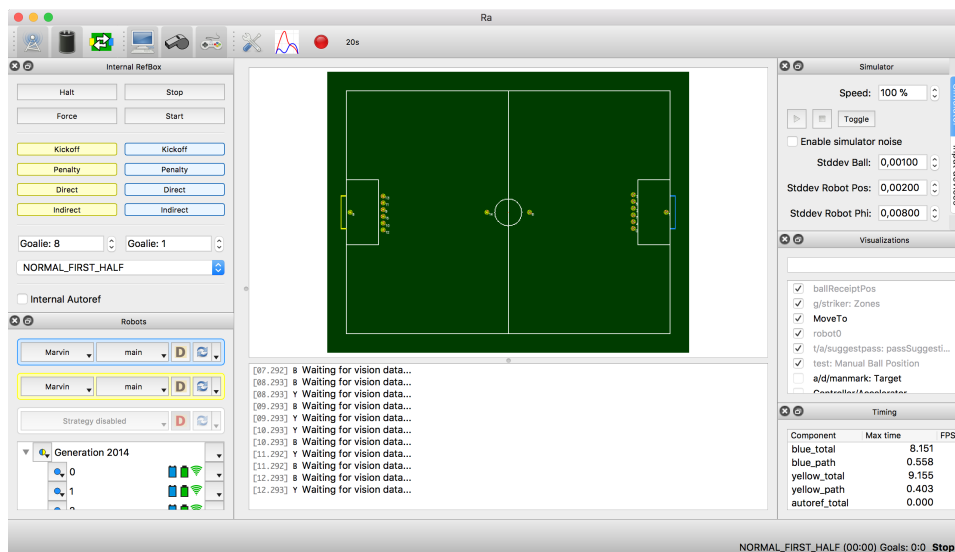
Ra and the other components in our framework constitute a complete system. Teams seeking for a fast access to the SSL League can use this very easily. With the provided framework the only thing missing for a full team software setup is a strategy, firmware for the robots and changes for the desired radio protocol.

Components included are:

- Simulator
- Tracking
- Pathfinding
- Strategy runner
- Visual overlay of debug information
- Debug tree
- Chip kick tracking
- Logplayer
- Robot control by input devices

Most valuable to us and any other team is the ability to debug the strategy. For that purpose we have built in multiple features supporting this.

- Logplayer: Analyze logfile frame by frame
- Backlog: Save the last 20 seconds of gameplay with full debug information
- Instant Replay: Analyze problems quickly in Ra without saving to logfiles
- Replay: Rerun a logfile in the logplayer with changed strategy code
- Plotter: Plot everything on the field of play including robots, ball, radio commands and strategy



**Fig. 1.** Ra with field of game and configurations