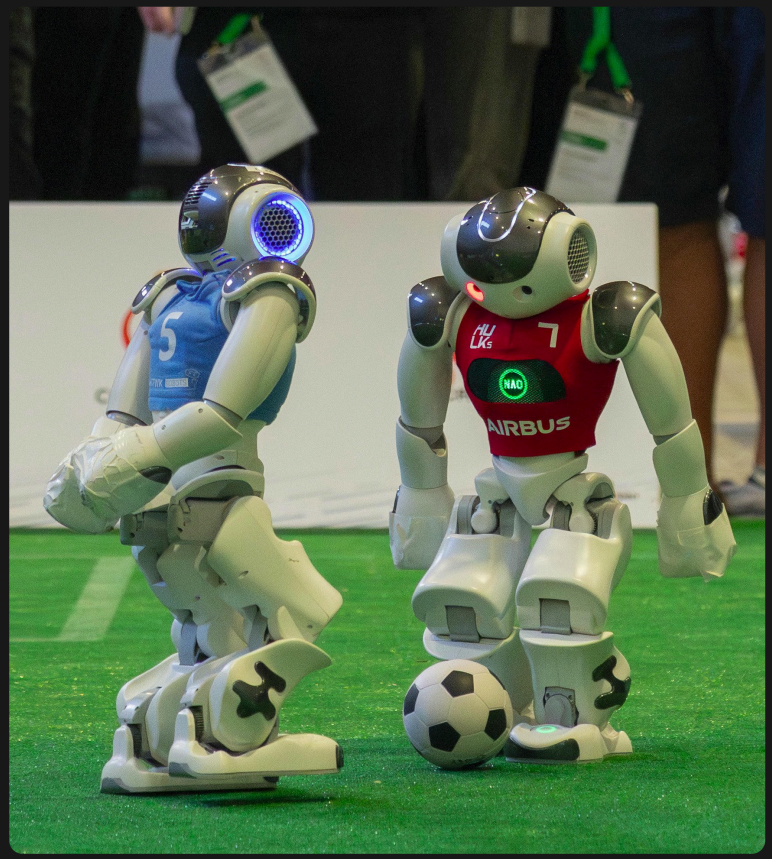
Robotics with Rust: A Case Study in RoboCup Soccer

Maximilian Schmidt, Ole Felber

What is RoboCup SPL?

- International competition
- autonomous soccer-playing humanoids
- Standard Platform League (SPL):
  ↳ all teams use identical NAO robots
- Focus: software engineering, teamwork, perception, planning, motion control

⟹ Real-time robotics under physical, strategic, and computational constraints



Who are the HULKs?

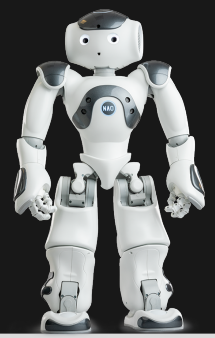
- Team from Hamburg (TUHH)
- Active since 2013
- Open-source contributions to software + research papers

First team to seriously **integrate Rust** in robot control

Where we use Rust

Robot Control

- Perception
- Sensor Fusion
- Behavior
- Motion Control



WebSockets

Tooling

- Live Visualization
- Replay Data
- Behavior Sim

Yocto Linux + Rust SDK

DEMO

Safety by Design

- ✅ Modern ecosystem
- ✅ Memory safety: no nulls, no data races, no UBs
- ✅ Borrow checker prevents most common bugs at compile time

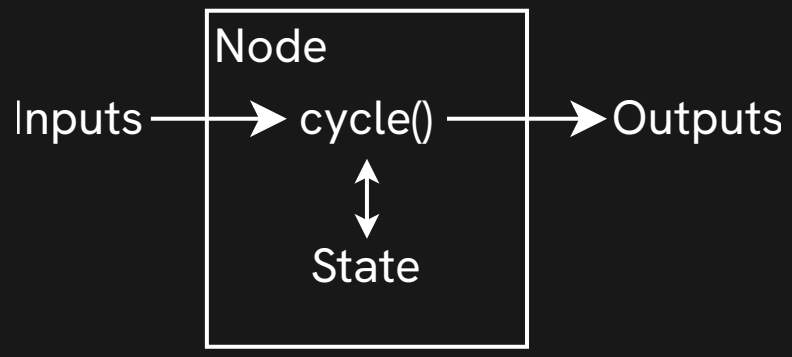
⇨ New contributors can write control logic without breaking internals

⇨ Reduces integration effort

⇨ Speeds up onboarding

Robot Software Architecture

- Custom framework
- Event-driven “node” model
- Features:
  - Time-sorted sensor events
  - Zero-copy message passing
  - Designed to be safe and nearly lock-free



⇨ *Advancing Humanoid Robotics with Rust - An Open Framework for Runtime Efficiency* (2024)[1]

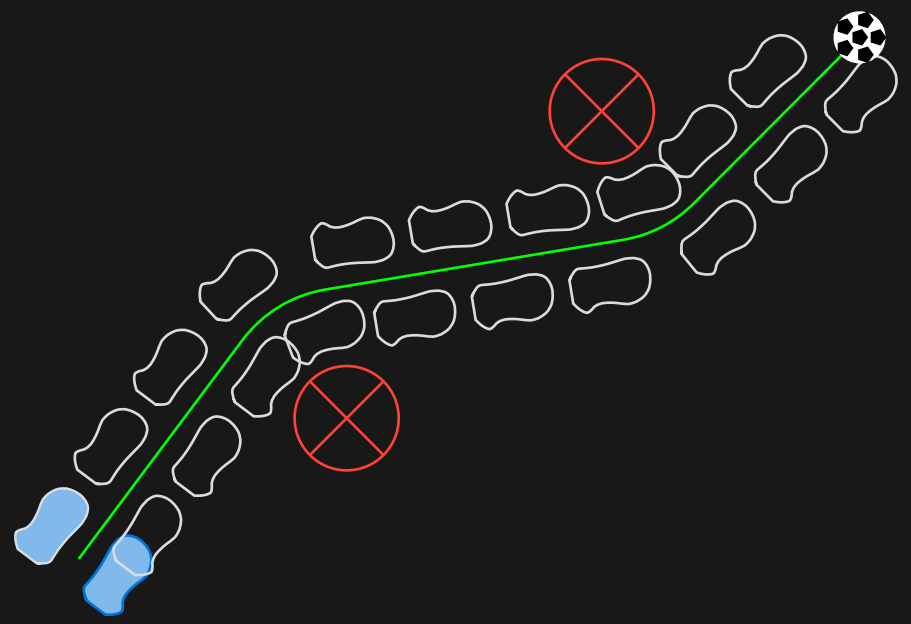
[1] DOI: 10.1007/978-3-031-85859-8_34

Rust-Powered Linear Algebra

```rust
fn ball_to_ground(
  ball: Point3<Camera>,
  ground_to_field: Isometry3<Camera, Ground>
) -> Point3<Ground> {
  camera_to_ground * ball
}
```

- `linear_algebra`: wrapper crate over `nalgebra`
- Transforms become explicit, type-safe operations
- Prevents mixing up incompatible frames at compile time
- Increases confidence and correctness in geometric computations

MPC-Based Step Planner



- Optimizes step sequences in real time (10ms budget)
- Will be presented in upcoming RoboCup 2025 paper

Still a Challenge: Neural Networks in Rust

- Inference ecosystem still maturing
- Currently rely on:
  - Hand-crafted JIT compiler[2] for models
  - Bindings to C++ frameworks (OpenVINO, TensorRT)
- Hope: candle, burn, tract

[2] BHuman/CompiledNN

Thank you!

hulks.de

HULKs/hulk

@hulks_tuhh