# RAC Robotic Soccer

# RACmotion Project


Dep. of Electrical and Computer Engineering,
Faculty of Science and Technology, University of Coimbra

**João Rodrigues, Sérgio Brandão, Rui Rocha, Jorge Lobo, Jorge Dias**

{joaor, brandao}@alumni.deec.uc.pt,
{jlobo, rprocha, jorge}@isr.uc.pt



FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA E DA TECNOLOGIA Portugal


## Introduction

The main goals of the RAC team (Robótica Académica de Coimbra) are to have a competitive RoboCup soccer team, interest students in robotics research, and build a team of high performance robots suitable for educational and research applications beyond the Robocup competition.

In order to achieve these objectives, several tasks were assigned to final project students. RACmotion was focused on robot assembly and motion control.

## RACmotion Objectives

- Omnidirectional drive control design and modeling;
- Project and implementation of a robot motion simulator using MATLAB® and SIMULINK®;
- Development of a calibration procedure;
- Prototype development and assembly;
- Deployment and configuration of a robot's embedded Linux™ operating system;
- Development of a communication system and communication protocol;
- Development of sensor and motor control software;
- Test and optimization of prototype mechanics and software in real world conditions.

## Robot Hardware Main Features

- Omnidirectional drive and kicker;
- NiMH batteries and PC104 power supply;
- DC motors and encoders;
- FPGA control card and motor driver daughter board;
- PC104 embedded computer;
- IEEE 802.11b wireless USB dongle;

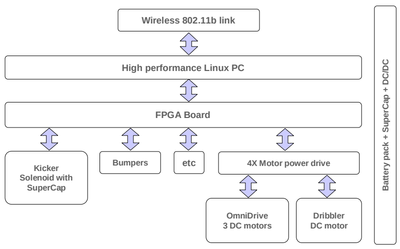



*Fig.1 – Hardware architecture for RAC SSL robot.*

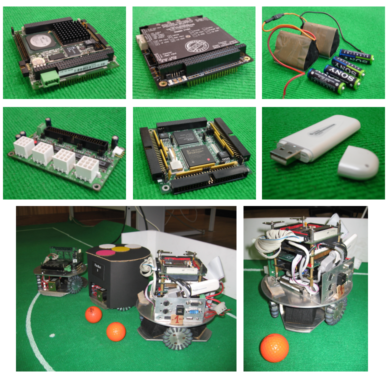
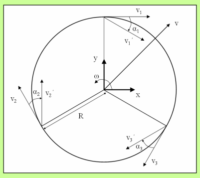


*Fig.2 – Hardware components and preassembled robots.*

## Omnidirectional Robot Model

The first step to develop a robot controller was to derive the inverse kinematics equations. In our model we consider several parameters, including skew wheel angles. This parameter was introduced to account for construction limitations that introduce miss-alignments of wheels and influence the real robot trajectory.



$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{r}\begin{bmatrix} v_1' \\ v_2' \\ v_3' \end{bmatrix} \Leftrightarrow \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{r} M \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

$$M = \begin{bmatrix} \cos(\alpha_1) & -\sin(\alpha_1) & -\cos(\alpha_1)R \\ -\cos(\frac{\pi}{3}+\alpha_2) & \sin(\frac{\pi}{3}+\alpha_2) & -\cos(\alpha_3)R \\ -\cos(\frac{\pi}{3}-\alpha_3) & -\sin(\frac{\pi}{3}-\alpha_3) & -\cos(\alpha_3)R \end{bmatrix}$$

*Fig.3 – Inverse kinematics equations.*

## Motion Simulator

Given the inverse kinematics equations, we simulated the nonlinear system using SIMULINK®. Simulation is divided in two parts: the motion controller and the motion simulator.

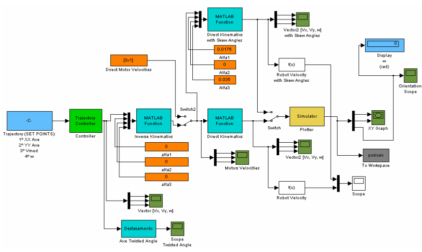


*Fig.4 – Simulator block diagram.*

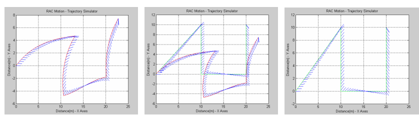


*Fig.5 – Robot trajectories can be visualized and the influence of motion parameters can be studied.*

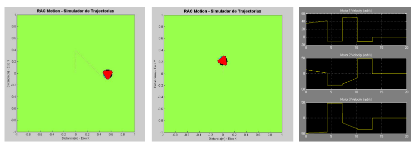


*Fig.6 – Robot's animation and motor's velocities during trajectory following.*

## Calibration

A calibration procedure was devised to estimate unknown wheel skew angles. Given the non linearity of inverse kinematics equations, we can't obtain a direct expression for $\alpha_1$, $\alpha_2$, and $\alpha_3$ (skew wheels angles). We developed an iterative calibration method. When the robot movement is parallel to a wheel axis, i.e. the wheel is not under traction, we assume that its effect on the trajectory is minimal and can be neglected. Under this assumption, we can decouple the behavior of two wheels from the third one.

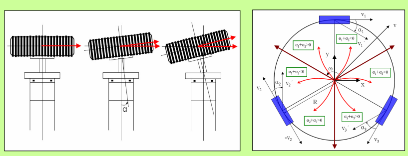


*Fig.7 – Skew wheels angles and their effect in robot trajectory.*

## Robot Construction and Assembly

The Robot's construction went trough several stages. First, it was built a prototype to test the hardware and to develop control software. More four robots were constructed based on prototype design.

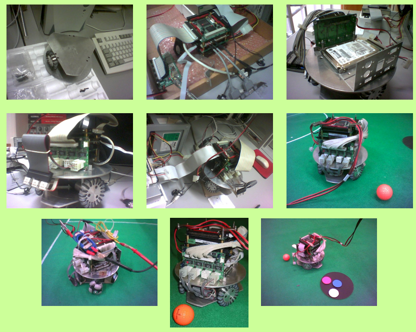


*Fig.8- Several stages of robot construction and assembly.*

## Control Software Development

Robot application software was developed for the Linux™ embedded operating system. The robot controller program has two major tasks: communications handling and motion control.

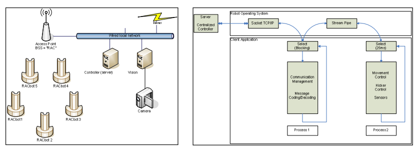


*Fig.9 - Robot controller diagram*

## Tests and Results

The robot's hardware and software were tested in the playing field. These tests helped us drawing conclusions about the design options taken. Further tests will allow adjustments and upgrades to the system.

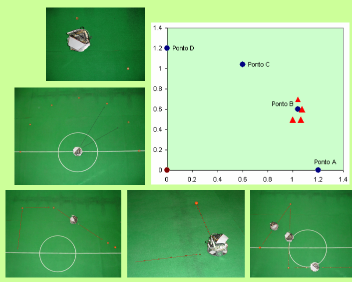



*Fig.10 – Motion control tests in the playing field.*

## Conclusions

We presented the modelling, control and simulation of the omnidirectional drive, and a calibration procedure. We presented the kinematics equations, plus preliminary control simulation results using a simple open loop controller. We also presented a calibration method to estimate the wheel skew angles. These robots will enable the setup of a competitive RoboCup team. The aim of the design options taken during this project was to provide a set of high performance robots suitable for educational and research applications beyond the RoboCup game.