Robótica 2005 – Actas do Encontro Científico
Coimbra, 29 de Abril de 2005

RAC ROBOTIC SOCCER SMALL-SIZE TEAM: OMNIDIRECTIONAL DRIVE MODELLING AND ROBOT CONSTRUCTION

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

* *Dep. of Electrical and Computer Engineering,*
Faculty of Sciences and Technology, University of Coimbra,
3030-290 Coimbra, PORTUGAL
{joaor, brandao}@alumni.deec.uc.pt,
{jlobo, rprocha, jorge}@isr.uc.pt

Abstract: The main goal of the RAC project is to develop a robotic soccer team for the RoboCup small-size league competitions. This paper is focused on two main issues: omnidirectional drive control design and modelling, and robot design and construction. The robot hardware implementation and design options are presented. Based on the robot kinematic model, a motion controller is designed using trajectory control techniques. A simple calibration procedure was devised to deal with construction deviations from the ideal model.

Keywords: Robocup, robot construction, omnidirectional drive, omniwheel, calibration

1. INTRODUCTION

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

In this paper we present the robot design, the kinematic model of the developed omnidirectional robot, control and calibration strategies used, and provide some details about the robot hardware and construction.

Initial work was done in surveying the existing teams to start with our design (Cornell, 2004)(RoboRoos, 2004)(CMUDragons, 2004). The following sections will describe the intended hardware design and first working prototype.

Fig. 1. *RacBot* - first working prototype.

2. ROBOT DESIGN

The Robocup F180 small size league requires fast slave robots with little autonomy. However we intend to eventually build smart autonomous robots. Figure 1 shows a first working prototype of our robots, the *RacBots*.

The design option of having a team of high performance robots suitable for educational and research applications other than the Robocup game was motivated by the big investment required. We chose to have small robots with full onboard processing power for embedded systems programming and VHDL hardware design by the students. The increased performance also met the requirements for research in autonomous and cooperative robots without moving into the more costly and space consuming middle sized league.

The robot design takes this into account. A basic implementation, the *RacSlaveBot*, will enable minimum operational requirements and rely on the central computer to close the control loop. An upgraded enhanced version, the *RacSmartBot*, will have onboard vision, inertial sensors, and processing power. Both implementations will share the same platform and drive system.

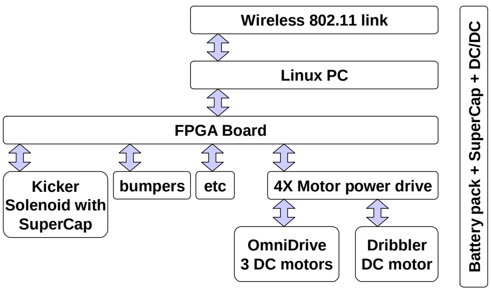

Fig. 2. *RacSlaveBots* - basic robot implementation.

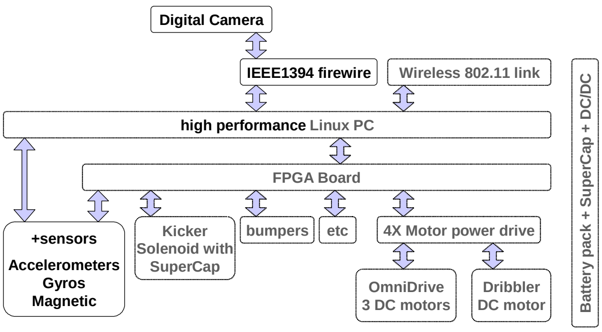

Fig. 3. *RacSmartBots* - enhanced robot implementation.

We realized that a key factor in the game play of a competitive team was to have a fast and precise motion control. We opted for an omnidirectional drive with three symmetric wheels 120° apart. This requires a precise control and the use of omniwheels, that provide traction in the rotation plane of the wheel and side roll along the axis.

Our omnidirectional mobile robot can move in any direction, while simultaneously controlling rotational speed, thus said holonomical. Compared with a more common car-like (non-holonomical) mobile robot, our omnidirectional robot has clear manoeuvrability advantages.

The aim is to have small compact and fast robots with omnidirectional drive and kicker. The robot onboard control is PC-based running Linux, using PC104+ modules. These allow stackable cards to be added or swapped for smarter robot enhanced versions. An FPGA PC104+ card is added to deal with motor control and all hardware I/O. The main actuators are the DC motors with encoders and the kicker. Wireless communication is done using IEEE 802.11 standard. The onboard power is supplied by a NiMH battery pack and a DC/DC power supply for the PC104+ boards.

3. OMNIDIRECTIONAL ROBOT MODEL

As a first step to develop a robot controller, the robot motion equations need to be derived. 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 the wheels and influence the real robot trajectory. The design and modelling of the omnidirectional drive was based on the works of (Schroder, 2002), (Carter, 2001), and (Loh, 2003).

We consider two frames of reference in our model. The body frame, fixed on the moving robot, with the origin at the chassis center, as show in fig. 4; and the world frame fixed on the playing field.

The following symbols are used in the modelling:

w_1, w_2, w_3 – motor speeds;
$\alpha_1, \alpha_2, \alpha_3$ – skew wheel angles;
v_1, v_2, v_3 – tangential wheel velocities;
$wheel_1$ – front wheel;
$wheel_2$ – rear left wheel;
$wheel_3$ – rear right wheel;
r – wheel radius;
r_r – robot body radius;
x, y – robot position;
$\vec{\boldsymbol{v}} = (v_x, v_y)$ – robot linear velocity
ω – robot angular velocity;

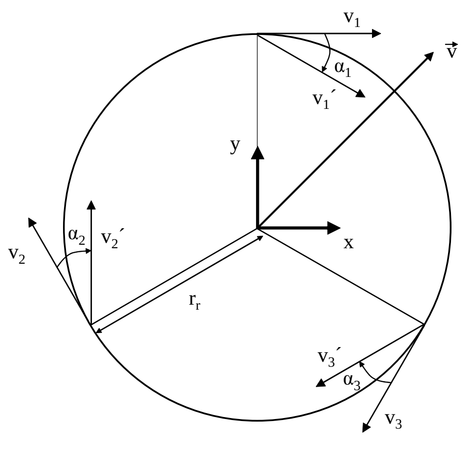

Fig. 4. Illustration of robot kinematics for tangential and twisted assembly of wheels.

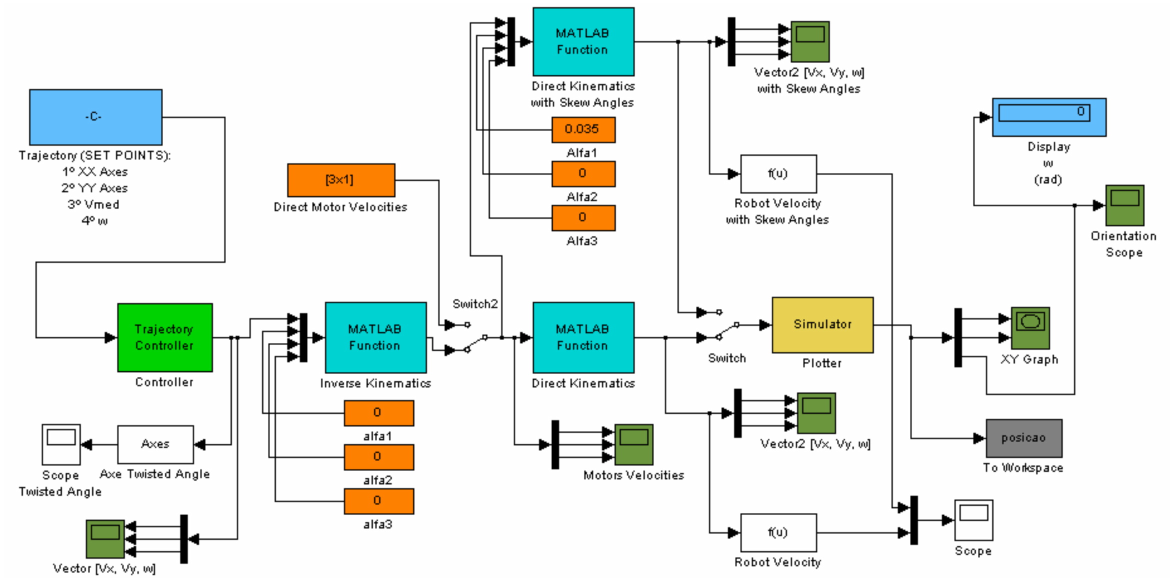

Fig. 5. Controller model and motion simulator.

The input variables for the inverse kinematics formulas are translational velocity of the robot center $\overrightarrow{\boldsymbol{v}} = (v_x, v_y)$ and robot angular velocity ω. The tangential velocities (velocities of wheel mounting points at the base plate) v_1, v_2 and v_3 are given by

$$
\begin{aligned}
v_1 &= v_x - \omega r_r \\
v_2 &= -\frac{1}{2}v_x + \frac{\sqrt{3}}{2}v_y - \omega r_r \\
v_3 &= -\frac{1}{2}v_x - \frac{\sqrt{3}}{2}v_y - \omega r_r
\end{aligned} \tag{1}
$$

If the motors are not mounted radially, and the wheels have a skew angle α, the tangential velocities will be given by

$$
\begin{aligned}
v_1^{'} &= \cos(\alpha_1)v_x - \sin(\alpha_1)v_y - \cos(\alpha_1)\omega r_r \\
v_2^{'} &= -\cos(\beta)v_x + \sin(\beta)v_y - \cos(\alpha_2)\omega r_r \\
v_3^{'} &= -\cos(\gamma)v_x - \sin(\gamma)v_y - \cos(\alpha_3)\omega r_r
\end{aligned} \tag{2}
$$

Where $\beta = (\frac{\pi}{3} + \alpha_2)$ and $\gamma = (\frac{\pi}{3} - \alpha_3)$. The rotational wheel speeds w_1, w_2 and w_3 are given by:

$$
w_1 = \frac{v_1^{'}}{r},\ w_2 = \frac{v_2^{'}}{r},\ w_3 = \frac{v_3^{'}}{r} \tag{3}
$$

Where r is the wheel radius.

Rearranging the terms, we have the following inverse and direct kinematics expressions:

$$
\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{r}\begin{bmatrix} v_1^{'} \\ v_2^{'} \\ v_3^{'} \end{bmatrix} \Leftrightarrow \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{r}M\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{4}
$$

$$
\begin{aligned}
& rM^{-1}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = r\frac{1}{r}M^{-1}M\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \Leftrightarrow \\
\Leftrightarrow & \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = rM^{-1}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}
\end{aligned} \tag{5}
$$

Where M is given by:

$$
M = \begin{bmatrix} \cos(\alpha_1) & -\sin(\alpha_1) & -\cos(\alpha_1)r_r \\ -\cos(\beta) & +\sin(\beta) & -\cos(\alpha_2)r_r \\ -\cos(\gamma) & -\sin(\gamma) & -\cos(\alpha_3)r_r \end{bmatrix} \tag{6}
$$

4. TRAJECTORY CONTROL SIMULATION

Trajectories are generated at a higher lever of game control, and consist of a group of set points to which the robot must comply. Each set point is composed by four parameters: (x, y, v, ω), where (x, y) is the position to be reached and $v = \|\overrightarrow{\boldsymbol{v}}\|$ is the velocity used to reach the point and ω the angular velocity.

In the Robocup small sized league, the overlooking global camera provides feedback for all robot and ball positions, orientations and velocities. But to have a fast control we rely on odometry when controlling a trajectory, and only use the visual feedback to provide corrections at a slower rate. A proper calibration of the robot geometry has to be performed so that we can rely on the odometry.

We assume a closed loop speed control at each axis, since this is implemented in the FPGA that drives the H-bridges using the feedback from the motor axis encoders.

The controller used in our simulation will be tested as an open loop controller, i.e. without the visual feedback. Given equations (4) and (5), we simulated the nonlinear system using Simulink, the graphical MATLAB workspace.

Simulation is divided in two parts: the motion controller and the motion simulator.

The motion controller first step is to convert trajectory set points into velocity commands based on the current robot position. In the second step we transform the robot velocity into wheel velocities by implementing the inverse kinematics function (4).

The motion simulator implements the direct kinematics function (5) and plots the trajectory by integrating the resulting velocity. The simulator can use skew wheel angles that are unknown to the controller, enabling the test of our calibration procedure. Knowledge of the skew angles is important since construction limitations can introduce miss-alignments of the wheels, or even intentional skew to accommodate longer motors, that influence the real robot trajectory.

Figure 6 shows the robot trajectory following four set points, considering ideal robot geometry, i.e. with zero skew angles.

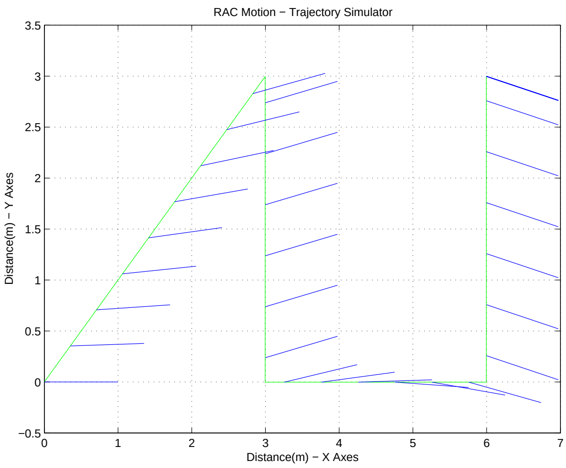

Fig. 6. Robot trajectory following set points $(3, 3, 10, 05)$, $(3, 0, 1, 0)$, $(6, 0, 1, -0.15)$, $(6, 3, 1, 0)$ considering ideal robot geometry.

The simulation logs full data about motor velocities and robot orientation along time, and the orientation is show in the plot at regular intervals.

When unknown skew angles are introduced we can observe their influence on the performed trajectory. After calibration, and using the estimated skew angles in the model, we are able to perform the correct trajectory, as shown in fig. 7.

The omnidirectional robot can simultaneously translate and rotate. The robot has three degrees of freedom. In simulation, we show that the controller can effectively decouple and stabilize the robot movement, but unknown skew angles

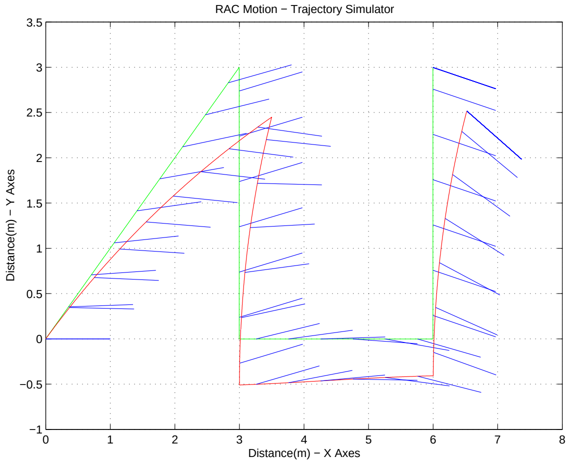

Fig. 7. Robot trajectory with unknown skew angles (red), and correct robot trajectory with calibrated skew angles (green).

can significantly influence the robot trajectory, degrading the controller performance.

5. ROBOT GEOMETRY CALIBRATION

Knowledge of the robot geometry is crucial to have a reliable control based on the odometry. Construction limitations can introduce miss-alignments of the wheels, or even intentional skew to accommodate longer motors, that influence the real robot trajectory. These can be modeled as skew angles at the wheel, provided that the wheels maintain a 120º symmetric layout, as shown in fig. 8.

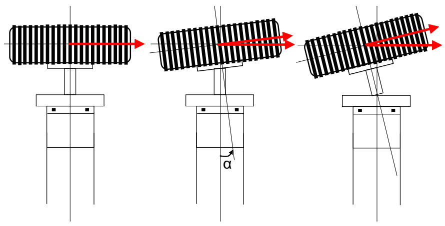

Fig. 8. Wheel skew angle.

A calibration procedure was devised to estimate these unknown wheel skew angles. Given the non linearity of equations (4) and (5) we can't obtain a direct expression for α_1, α_2 and α_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.

The calibration procedure is performed in three steps, one for each wheel. At each step linear motion is attempted along one wheel axis, and the skew parameters of the remaining wheels are adjusted until the robot performs a straight

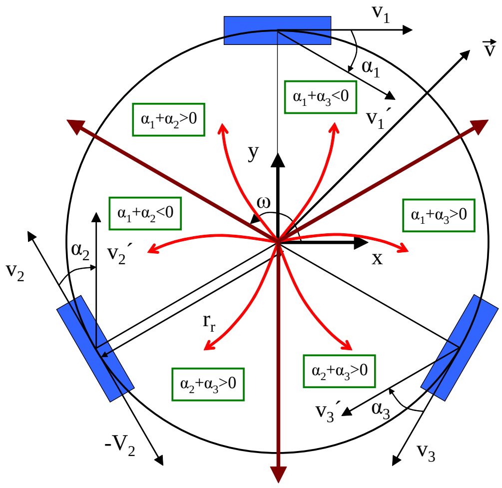

Fig. 9. Skew wheels angle relation to different trajectories.

line movement. At the last step, all controller skew angle parameters are adjusted in magnitude, maintaining the identified ratios, to remove any common factor introduced in the process. In the end, all three motors will be calibrated relative to each other.

6. ROBOT CONSTRUCTION

To have versatile and powerful mobile robots we chose to equip them with a full onboard computer. The CPU PC104+ board chosen was the M570B from Seco (Seco, 2004). This is a low power board that integrates a high performance VIA Eden CPU with graphic Controller, ethernet, audio, RS232, USB, LPT, IDE, and ISA PC104 and PCI PC104+ bus connectors.

The simplest and smaller solution we found for the wireless communication, was to use a USB wireless 802.11b dongle. This can be easily replaced later if needed.

An FPGA PC104+ card, 4i65 from Mesa (Mesa, 2004), was used to deal with motor control and all hardware I/O. The main actuators are the DC motors with encoders and the kicker. The DC motors are driven by an H-bridge motor control board, 7i30 (Mesa, 2004). The kicker power drive is custom build but all logic is done by the FPGA.

For our current research purposes the FPGA is programmed at startup via the PCI bus. A working *slave* robot can be implemented with the FPGA alone, i.e. no PC, if an appropriate radio link and an EEPROM to program the FPGA are used.

To implement an omnidirectional drive, a symmetric 3 wheel triangular mount was used, as shown in figure 10.

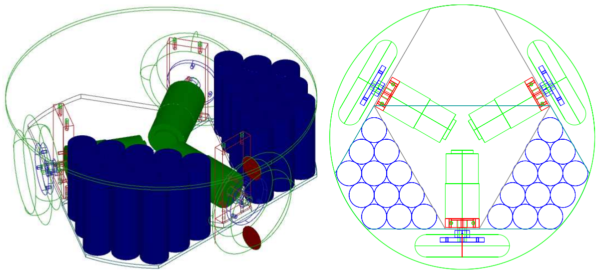

Fig. 10. *RacBot* chassis and motor drive layout.

To meet the necessary torque and size requirements, Faulhaber 2224 minimotors where used with 23/1 planetary gearheads with 3.71:1 ratio (Faulhaber, 2004).

To implement the above drive omni wheels are required, i.e., wheels that can provide traction perpendicular to their axis, and free roll for motion along the axis. Initially a plastic conveyor belt roller was tested, as shown in figure 11.

These where readily available, but had poor grip on the carpet, where very sensitive to floor level and axis alignment, and took up too much space.

Motivated by other team designs (5DPO, 2004) we attempted to build the wheels using standard parts and machined acrylic. Metal washers and a curtain metal ring were used to build a successful but limited wheel prototype (fig.11). The wheel was heavy, and side rolling had some friction.

Fig. 11. Plastic omni wheel used in conveyor belts and omni wheel built from standard parts and machined acrylic.

To build a better omnidirectional wheel we had to design and out-source the parts. The perimeter wheels are built from a nylon rod machined with an inlet to fit an o-ring rubber tyre, to which a small a metal axis is fitted. These are fitted between two aluminium disks with radial cuts for the wheels. The final result is shown in figure 12 The design follows the approach used by many other teams (Cornell, 2004)(CMUDragons, 2004)

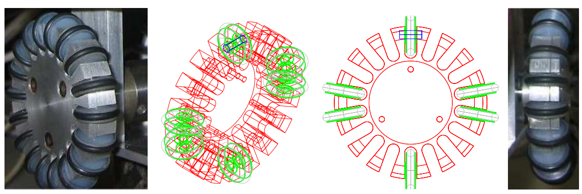

Fig. 12. *RacBot* omni wheel design and implementation.

The kicker was implemented using a standard solenoid, with some modifications to fit the kick head. A disposable camera flash circuit was adapted, using its blocking oscillator with step-up transformer to charge the high voltage capacitor for impulse kick discharge. The FPGA trigger is isolated from the high voltage by an optotriac that controls the switching power triac.

To provide suitable power to all the electronics a DC-DC module is used. The motors are driven directly from their separate battery pack.

7. CONCLUSIONS

We presented the design and construction of the RAC team robots, the modeling of the omnidirectional drive, control simulation and a calibration procedure. We have presented the kinematics equations, plus preliminary control simulation results using a simple open loop controller. We also present a calibration method to estimate the wheel skew angles. These robots will enable the setup of a competitive Robocup team, and the design options taken aim at having a set of high performance robots suitable for educational and research applications beyond the Robocup game.

ACKNOWLEDGEMENTS

This work was funded by FCT — Fundação para a Ciência e a Tecnologia, with project grant POSI/ROBO/43890/2002. This work was possible thanks to the contribution from all RAC Robocup team members, and in particular João Dias for the robot and omniwheel technical design, and Carlos Ferreira for the motor drive setup (João Dias, 2005).

REFERENCES

Brian Carter, Matt Good, Mike Dorohoff, Jae Lew, Robert L. Williams II, Paolo Gallina (2001). Mechanical Design and Modeling of an Omni-directional RoboCup Player.

Joachim Schroder, Peter Steinhaus, Tilo Gockel, Rudiger Dillmann Institute for Computer Design and Fault Tolerance, University of Karlsruhe. Design of a Holonomous Platform for a Humanoid Robot using MCA2.

W.K. Loh, K.H. Low, Y.P. Leow. Proceedings of the 2003 IEEE International Conference on Robotics and Automation Taipei, Taiwan, September 14-19, 2003.Mechatronics Design and Kinematic Modelling of a Singularityless Omni-Directional Wheeled Mobile Robot.

5DPO (2004). 5dpo robocup f-180 league team. *http:// paginas.fe.up.pt/ robosoc/.*

CMUDragons (2004). Carnegie mellon robot soccer. *http:// www-2.cs.cmu.edu/ robosoccer/ small.*

Cornell (2004). Cornell robocup team. *http:// robocup.mae.cornell.edu.*

Faulhaber (2004). Faulhaber 2224 minimotors. *http:// www.micromo.com/ library/ docs/ products/ 2224_SR.PDF.*

João Dias, Carlos Ferreira, Jorge Lobo (2005). Racbots - building small robots for robocup. *technical report.*

Mesa (2004). Mesa pc-104 motion control cards. *http:// www.mesanet.com/ motioncardinfo.html.*

RoboRoos (2004). UQ's RoboRoo Team Homepage. *http:// www.itee.uq.edu.au/ dball/ roboroos.*

Seco (2004). Seco pc104+ m570b. *http:// www.seco.it/ schede_prod/ M570-HTE01A.htm.*