Mechatronics and Machine Vision 2003: Future Trends,
ed. J. Billingsley,

Research Studies Press Ltd, Baldock, UK,
ISBN 0863802907, 2003.

Autonomous Racing Car Competition for Mechatronics Engineering Education

Samuel N. Cubero, Jeffrey Layanto and Matthew Goode
Mechatronics Studio, School of Mechanical Engineering
Curtin University of Technology, Perth, Western Australia

Abstract
A robot competition is described which overcomes many inherent limitations of existing competitions. The proposed "CARbot" (Computer-aided Autonomous Racing robot) competition requires students to build and program their own sensor-guided electric-powered racing cars and race these vehicles against each other on a standard closed-loop flat track. The design and manufacture of the "official" racetrack is described briefly as well as the basic design and control specifications for a fully functional "CARbot". Students will be able to develop useful skills relating to low-level microcontroller programming, high-level PC programming and interfacing, basic PCB manufacture and design, serial communications and protocol programming, sensor interfacing, signal conditioning, hardware debugging and feedback control programming, while they work towards building an operational vehicle to race in an end-of-semester competition.

Keywords:
Autonomous Racing Car, "CARbot" (Computer-guided Autonomous Racing robot), Micromouse, Robot Soccer, Robot Sumo, Microcontroller programming, CCD optosensor array, Problem Based Learning (PBL)

1. Justification for a robotic racing car competition

"Why do we need another robot competition? Don't we already have Robot Wars™ [3, 5], Robot Soccer (RoboCup) [1], Micromouse and Robot Sumo [4] competitions?" Table 1 shows a qualitative comparison of the most popular robot competitions. It compares the typical costs involved in Australian dollars, the technologies learned (learning outcomes), the number of students required per team, the number of robots needed for a competition, the typical times required for completion, the degree of difficulty and complexity (amount of learning required) and the "excitement factor", which can be measured by the variety of events that can occur within each type of contest. The more things that the robots can do, or the more things that can happen to them, the more interesting they will be to watch.

Table 1. Comparison of popular robot competitions

Typical Attributes	Robot Wars	Robot Soccer	Micromouse maze contest	Robot Sumo	CARbot racing contest
$ Cost per robot (AU$)	>$800 to $4000	>$500 to $1000+	>$100 to $300+	>$60 to $200+	>$60 to $500+
Learning outcomes (practical skills gained by students)	Mechanical design for robustness, actuator mounting & control, welding & fabrication, remote or radio control interfacing	Micro-controller usage, 2D image analysis, PC interfacing, wireless networking, motor control, soccer skills	Micro-controller usage and software design, sensor interfacing, motor control, maze solving algorithms	Simple motor control, analog circuit design, switch sensor inter-facing	Micro-controller usage, 2D or 1D line scan image analysis, DC motor control, PC interfacing, steering algorithms
Students per robot	1 or 2	2	1 or 2	1	2
Minimum No. robots	2	2	1	2	2
Build time (months)	6	12	6	2	4
Difficulty out of 5 stars	✯	✯✯✯✯✯	✯✯✯	✯✯	✯✯✯✯
Excitement factor out of 5 stars	✯✯✯	✯✯✯✯	✯✯✯	✯✯	✯✯✯✯✯

Compared to the other types of contests, a Computer-aided Autonomous Racing robot contest offers a very low cost-to-benefit ratio and gives students a "hands-on" introduction to many areas of Mechatronics technology within only one teaching semester. The following sections describe how a "CARbot" racing competition could be implemented into a Project Based Learning (PBL) course.

2. Autonomous racing car competition rules

The objective of this competition is for students to build the fastest and smartest possible autonomous racing vehicle. The CARbot racing competition requires students to work in pairs to design, build and program their own PCB controller board. As an example, an onboard 8-bit microcontroller (Atmel™ AVR AT90S8535 [2]) can be used to analyse a CCD sensor and control two DC motors (in "skid steering" mode) so the car can steer itself on the racetrack and overtake other competing CARbots. Competitors are free to design their own controller.

2.1 Racetrack and vehicle design rules

The closed-loop racetrack for the "CARbot" competition is shown in Figure 1. It consists of 4 flat wooden panels (in this case, 4 doors placed side by side) painted with a black background and a white 45 cm wide track. For detailed dimensions and construction procedures, visit http://www.mech-eng.curtin.edu.au/carbot/. The white track is wide enough to accommodate three CARbots side by side, thus allowing ample room for overtaking manoeuvres. The top-view dimensions of any vehicle in the competition must not exceed 12.5 cm wide × 14 cm long (in the forward direction), which is exactly the same size as a standard hardcover CD case. There is no height restriction for mounting a CCD camera on a tall mast mounted on the vehicle in order to obtain a wider image of the racetrack. Students will have to discover by themselves the advantages and disadvantages of using a very tall mast. All car bodies, cameras, masts and wheels must be completely black in colour, except for a unique identifying colour or number at the front of the vehicle. All vehicles must be fully self-contained (no external tethers or power cables are allowed), electric motor driven and must operate without any external guidance.

Fig 1. Standard Racetrack for the "CARbot" competition

2.2 Autonomous Vehicle Mechanics

Each team of two students is given a "Radio Control" car kit, complete with 2 DC motors, wheels, gears and body (and a remote control). The car kit recommended for this competition is the Enertec™ "SUV Rally Racer" RC toy car, distributed by Gameco Ltd., however, any vehicle which satisfies the above length and width restrictions is acceptable for entry into the competition. The "SUV Rally Racer" toy cars (approximately AU$23 each) come with gears and DC motors. Students can remove the car's control board and replace it with their own controller PCB.

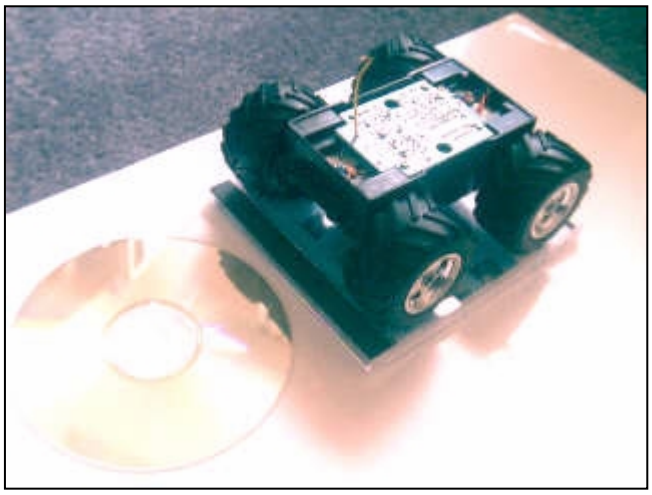

Fig 2. CARbot body and wheels (camera not shown)

2.3 Competition Guidelines

On the day of the competition, all fully functional CARbots will race individually against the clock and the fastest car will win "pole position" (the innermost lane position closest to the starting line). Each car must have a starting push-button or switch that starts the program. A PC will display a traffic light image, showing 4 circles: Red, Red, Red, Green, which will flash in succession at 1 second intervals). The "start" button must only be pressed after the "Green Light" or GO signal turns on. This is the only physical contact allowed between a competitor and his or her CARbot during the entire race. The lap times and final finishing time for each CARbot will be measured with a stopwatch by a non-competing observer who is not on the same team. Vehicles that fall off the racetrack or that can no longer move under their own power must be promptly removed and cannot rejoin the race. Any vehicles that are seen to drive in the wrong direction, against the normal direction of traffic, or any vehicle that takes a shortcut, driving entirely over the black divide at the centre of the track, will also be disqualified from the race and must be removed from the track immediately to avoid hindering other vehicles. The fastest CARbot that completes the maximum number of full laps will win the championship trophy! Standard 9V batteries can become flat in around 10 minutes so a limit of 20 laps for a complete race is reasonable to determine the winner.

2.4 Learning outcomes

In laboratory time, students using the AVR™ microcontroller [2] will learn how to:

♦ design a voltage regulated power supply, a crystal (clock) source, a SPI programmer to download programs, and an interface for a linear CCD optosensor.

♦ design, etch, drill, assemble and test their PCB (using Protel™ and other tools).

♦ download programs (assembly language or BASCOM™ [6]) into Flash ROM

♦ control two independent motor speeds with the AVR's on-chip PWM hardware.
♦ interface the AVR to a PC via serial communications (RS-232C COM1 port) and program a "master-slave" serial communications protocol for data sharing.
♦ create a "Graphical User Interface" (GUI) for plotting control variables vs. time
♦ write software to analyse image data for steering and motor speed control.

3. Vehicle control algorithms

The success of each team depends largely on how quickly and how well the suggested control strategies shown in Figure 3 can be implemented. There are four suggested "control layers" for making the "CARbot" completely self-steering and autonomous on the racetrack: image, navigation, steering and motor speed layers.

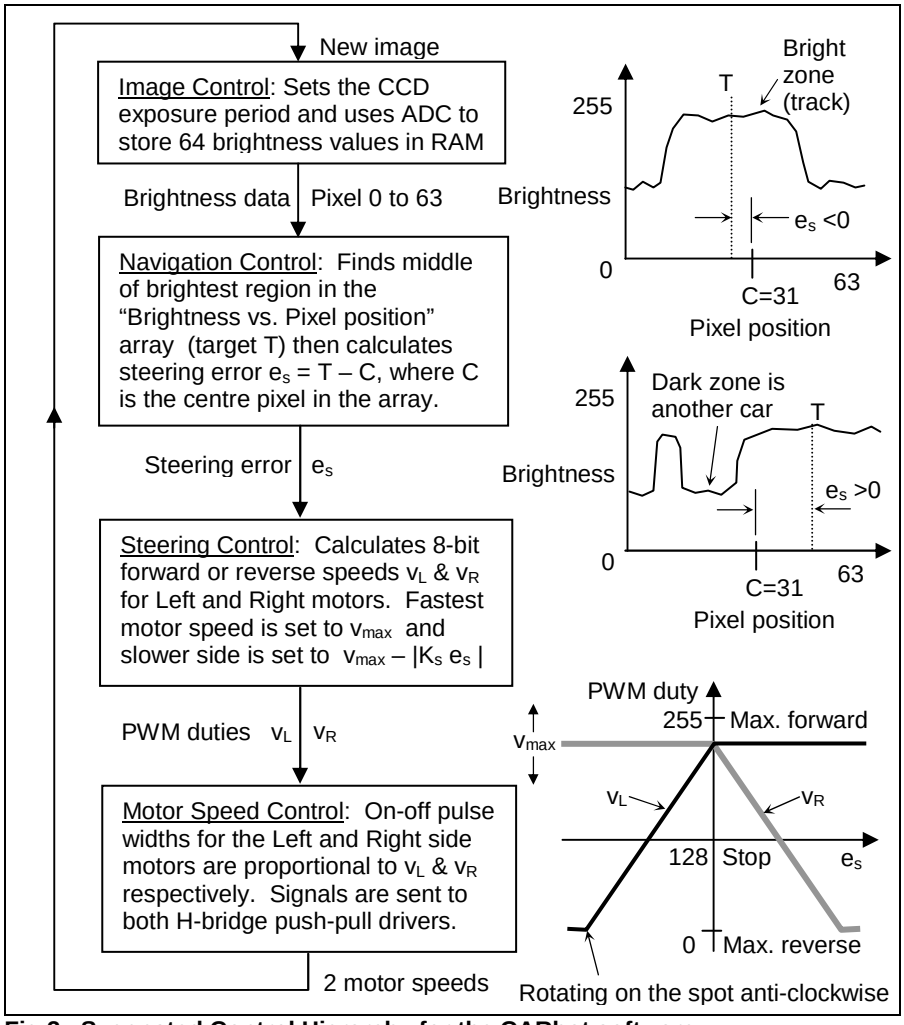

Fig 3. Suggested Control Hierarchy for the CARbot software

3.1 Image control layer

The "image" is basically an array of 8-bit (grayscale) numbers each ranging between 0 to 255. These numbers represent the "brightness" of light measured at known pixel positions on the CCD optosensor measured within the previous "exposure" period. This table of numbers is produced by the AVR's Analog-to-Digital Converter (ADC) which reads the analogue voltage outputs of the CCD line scan sensor (eg. Texas Instruments™ TSL214, available from RS™ Components).

The "image control" layer must send the correct SI (Start Integration) pulse to the CCD chip and wait a certain amount of time for light absorption ("integration time" or exposure period). At the end of the exposure period, it must repeatedly request the sensor to output the 64 analogue voltages in succession so that each voltage can be converted to an 8-bit number and stored in a table in the AVR's RAM. This type of CCD sensor, in effect, behaves like an analogue shift register when outputting brightness values. Each number is proportional to the "brightness" or light energy measured at that particular pixel position during the given exposure period. In order to monitor these 64 brightness values stored in RAM memory, these values must be sent to a PC for graphical plotting. The "integration time" for the CCD array may need manual adjusting to allow for sufficient light energy to be absorbed in each exposure period, otherwise, there may not be enough contrast in the image to distinguish between the dark and the bright regions of the image produced by the CCD sensor. The magnitudes of the brightness values detected are proportional to the exposure time and the size of the aperture (hole) behind the lens of the camera housing.

It is important that the sensor surface of the CCD chip is centred with and placed at a suitable distance (about 12 mm) behind a wide-angle lens so that the widest possible light image can be captured by all pixels. The "field of view" or width of the racetrack's "strip" image seen by the lens can be maximised by mounting the lens and sensor (or camera housing) as high above the racetrack as possible while pointing forwards at around 45 degrees downwards so that the triangular plane of vision of the camera intersects the racetrack surface at a line which is about 10-15cm in front of the CARbot's body. The length of this line is known as the "field of view". The middle of the CCD sensor array (or the camera's centre-line) should lie within in the vehicle's midplane, equidistant to the left and right side wheels. Also, the CCD sensor array should be oriented perpendicular to the straight or "forward" direction of travel for the car. (Fig. 4)

3.2 Navigation control layer

The input for this layer is the array of "brightness values vs. pixel position". This control layer searches for the largest "bright" zone in the image and marks its centre as the target position. The output of this layer is the steering error e_s , which is the pixel difference between the target pixel position and the centre pixel (No. 31) in the 64 pixel CCD array. e_s can be positive (turn right) or negative (turn left). Note that the search for the target pixel is useful for a very general "overtaking procedure". If a black car ahead appears in the field of view of the camera and partially obscures the full width of the white "road" image in the brightness array, the navigation control layer can still search for the centre of the "largest" bright

zone (which has the largest concentration of high brightness values). If the width of this widest bright zone is too narrow, it means that the road ahead is blocked by two or three cars side by side (or that the car is too close to a car in front), and hence, the car should not attempt to overtake or increase its present speed. If there are more than one "bright zones" or plateaus of wide brightness zones in the image, the AVR can perform a scan on the pixel widths between neighbouring pairs of steep positive and steep negative slopes and choose the pair of slopes that are furthest apart as bounding the largest bright zone. Hence, the middle pixel position between these two slopes is the target pixel T, as seen in Figure 3.

3.3 Steering control layer

The input for this layer is the steering error e_S which could be a positive (need to turn right) or negative (need to turn left) value. This is multiplied by a steering gain value K_s so that the wheels on the slower side (left or right side) must reduce their speed by an amount $K_s\, e_s$ relative to the speed of the faster wheels on the other side. The maximum speed of the vehicle is governed by the speed of the fastest wheels, set at v_{max}. For straight or forward driving, $e_s = 0$, hence, the wheels on both left and right sides of the vehicle can run at the same speed, v_{max} , which is an 8-bit number representing a duty value for the PWM outputs which control the forward or backward speeds of both motors.

One DC motor controls the two left side wheels (which are mechanically coupled together by gears such that they rotate at the same speed), and the other DC motor controls the right side wheels (also travelling at the same speed). The vehicle can be steered via "skid steering", which is similar to tank track control. For example, if the wheels on the left side travel slower than the wheels on the right side, the vehicle will tend to steer left, with the sharpness of the turn dictated by $K_s\, e_s$, where e_s is negative since the target pixel is left of the centre pixel in the image. If the left and right side wheels rotate in opposite directions at the same speed, the vehicle can rotate on the spot without changing position.

Since there are two motors with their own individual speed setting, the outputs for the steering control layer are the two duty values, v_L and v_R , for the left and right side motor speeds respectively. The PWM duty values range between 0 to 255, where 255 represents the maximum forward speed for a motor, 128 represents a complete stop (brake) and 0 gives maximum reverse speed. These values correspond to mean voltages on the AVR's PWM output pins which are connected to an external "dual H-bridge" push-pull driver chip (SGS-Thomson™ L293B) and two inverters. Note that v_{max} can be adjusted to the maximum desired forward or reverse speed at any time via software and that during forward or reverse motion, one motor is always travelling at speed v_{max} while the other drops below this value by an amount proportional to the steering error e_s. For steering in the "reverse direction", the target velocity (PWM duty) graph for determining v_L and v_R , shown at the bottom of Figure 3, may be flipped upside-down (flipped vertically) so that v_{max} is well below 128 and "forward" may be treated as "reverse". In this case, the slower motor must be run at a speed of $v_{max} + | K_s\, e_s |$. This can be implemented as another driving mode, just like rotation on the spot in the anti-clockwise (left) or the clockwise (right) direction.

3.4 Motor speed control layer

The inputs for the motor speed control layer are the two speed values, v_L and v_R , from the steering control layer. These values are stored in the AVR's Timer1 compare register for controlling the duties or pulse widths of the two "on-off" PWM output signals which set the two independent motor speeds.

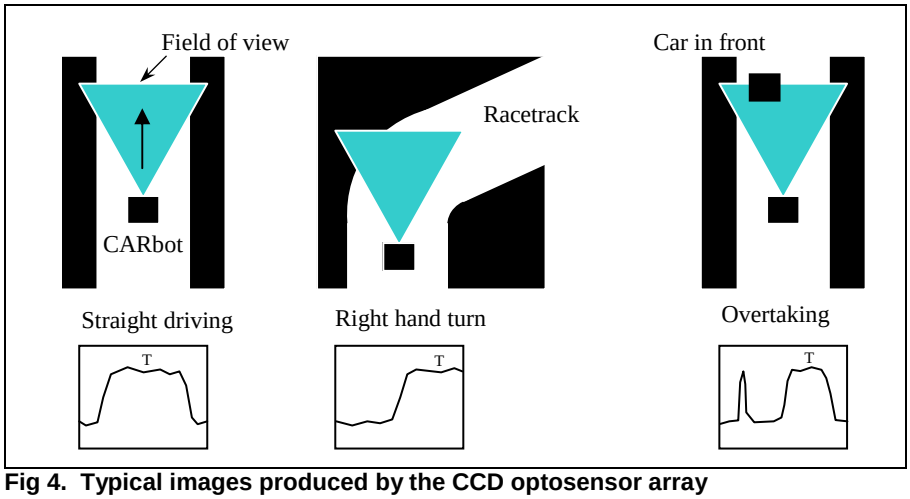

Fig 4. Typical images produced by the CCD optosensor array

4. Conclusions

The CARbot (Computer-aided Autonomous Racing robot) competition described in this paper can be implemented easily using low cost components (<$60) and free software development tools. Currently, one CARbot has been built and can successfully drive itself around the track autonomously. Future work will focus on improving the car's power to weight ratio (energy efficiency) and using 2D vision hardware and analysis. It is hoped that this competition will increase in popularity in the years ahead and eventually become an exciting international event.

References

1. Wyeth G., Brown B., "Robust Adaptive Vision for Robot Soccer", Proc. 6th International Conference on Mechatronics and Machine Vision in Practice, 2000, Research Studies Press Ltd., ISBN 0-86380-261-3
2. Atmel™ AVR websites www.atmel.com & www.avrfreaks.com
3. Miles P., Carroll T., "Build Your Own Combat Robot", McGraw-Hill/Osborne Publishing 2002, ISBN 0-07-219464-2.
4. Miles P., "Robot Sumo: The Official Guide", McGraw-Hill/Osborne Publishing 2002, ISBN 0-07-222617-X.
5. Hannold C., "Combat Robots Complete", McGraw-Hill/Osborne Publishing 2003, ISBN 0-07-140888-6
6. BASCOM-AVR™ compiler by MSC Electronics: www.mcselec.com