Team description RFC Uppsala – Aros

L. Asplund[1]
L. Almgren[1], D. Andersson[2], J. Andrén–Dinerf[1], C. Benson[1], N. Berg[1], H. Bergkvist[2], H. Bovin[2], M. Brunk[1], T. Forslund[2], A. Gudmundsson[1], C. Hörz[2], F. Karhu–Wedin[2], J. Lönnberg[1], J. Nielsen[2], H. Olsson[1], T. Petterson[1], M. Pettersson[1], M. Röjdeby[1], C. Sjöö[1], M. Sträng[2], H. Strömblad[2], M. Stålberg[2], P. Sundin[1], P. Sävström, M. Wall, and K. Wijk[1]
aros-vt03@student.docs.uu.se

[1] Department of Information Technology, Uppsala University, Sweden
[2] Department of Natural Science and Engineering, Mälardalens University, Sweden

Abstract. The soccer playing robot AROS presents a new design philosophy in robot design with focus on modularity and the capability of adding and subtracting modules without the need of wiring, solved by implementing an optical CAN–bus constituted through a vertical mounted titanium tube functioning as the robot's communication backbone as well as housing electric power wiring for the "plug–and–play" sensor modules.

1 Introduction

Yet a new robot design has emerged from the RFC Uppsala Team in collaboration with students from Mälardalens University in Västerås, Sweden, giving a total of 27 students working with constructing the new robot AROS. The new design idea is to make a complete module–based system with "plug–and–play" functionality. The base system is 450 mm in diameter equipped with a omnidirectional driving mechanism consisting of three motors oriented in 120 degree arrangement, each equipped with a "Swedish" wheel with 24 washers on the rim. The base system also contains batteries and one of the latest FPGAs from Xilinx which contains an internal PowerPC CPU, used as the main computer. A vertical titanium pipe is mounted on the base system which constitutes the physical layer of an optical CAN–bus, forming the intra–communication backbone of the robot. The various sensor subsystems are mounted on circular discs and mounted on the pipe. Each disc is equipped with an optical CAN transceiver, a stepper motor for independent 360° rotation capability and an optical index switch, apart from the actual sensing device. The motor controller on each disc can lock onto the robot direction or lock to the field geostationary, using the output from a disc equipped with a gyro. The Aros 2003 project falls into the following subprojects

- The vision subsystem ChipVision.
- The inter–robot communication subsystem ArosComm.
- The strategy software subproject.
- The hardware base system.

2 The ChipVision project

A brief presentation will be given here since a more thorough description of this project can be found in [1]. The aim for the ChipVision project is to design and build a real–time image analysis module using a one million gate Field Programmable Gate Array (FPGA) from Xilinx. In [6] a similar hardware setup is described. On the board there is also 16 MB RAM and 16 MB Flash EPROM. The camera is an Omnivision OV7620 digital colour camera with the capability to output images with 640×480 pixel resolution at 30 Hz. The camera is controllable using an I^2C bus with changeable settings such as brightness and colour balance. The camera is directly coupled with the FPGA which is fed with images in the RGB colour space at a rate of 30 Hz. The FPGA also contains controllers for the camera, the CAN–transceiver and the stepper motor. At an initial calibration stage, the

FPGA analyses the colour balance in the camera output and tries to adjust the settings according to the present light conditions. Having a correct colour balance is crucial for the image analysis, since objects are recognised using thresholding in different colour levels. At playing mode, the FPGA has a 5–staged pipeline for the image analysis algorithms. The steps are

1. Conversion of RGB colour space to Hue, Saturation, Intensity (HSI) colour space (Saturation not used).
2. Thresholding in the H channel and extraction of black and white pixels.
3. Relaxation in an $n \times n$ environment, where $3 < n < 11$.
4. Labelling of objects using colour codes and calculation of object descriptors (area, perimeter etc.).
5. Classification into interesting objects, i.e. ball, goal, opponent robot etc.

The lowest pixel value corresponding to an object is sent to the distance and angle calculation algorithm which is based on principles of projective geometry, where each point on the soccer field corresponds to exactly one point in the camera's image plane. With the knowledge of the camera's properties, a very precise estimation of the distance and the angle relative to the gyro direction is done. The same algorithms are used to perform an estimation of the robot's position on the soccer field with a resolution down to 10 cm, by using the corner posts as landmarks. An algorithm for finding lines in the soccer field based on the Hough transform ([2], [3]) will also help doing position estimation. The algorithms are prototyped in MATLAB which allows for manual or even automatic translation into VHDL. In [4] an automated translation tool named MATCH is presented and [5] presents a floating point representation using optimal FPGA resources.

3 The ArosComm project

The ArosComm group have chosen to develop three technologies for their inter-robot communication, one not excluding the other. These are based on infrared communication, IEEE 802.11 wireless LAN and Bluetooth radio technology. Bluetooth is not allowed for the competitions in Paderborn, Germany in April 2003, but it is not yet made clear if it is allowed in the RoboCup World Championships held in Padua, Italy in July 2003.

Infrared communication

Inter–robot communication by the means of infrared light is accomplished by using standard IrDA components such as a HSDL1001 transceiver and a HSDL7000 circuit controlled by an Atmel AVR8535 microcontroller. This microprocessor is in turn connected to an CAN–transceiver to communicate through the robot's optical CAN–bus. The customised hardware enables infrared communication at distances up to 10 meters at a data rate of 125 Kbits/second. The data transmission is made more reliable by the development of a higher level protocol using checksums, acknowledgements and retransmissions. Due to the line–of–sight requirement when using infrared communication, the robot disc carrying this technology, needs to be aligned towards each other. Customised hardware and software is developed for the purpose of establishing this alignment.

Optical CAN–bus

Transmission over the CAN-bus is by means of a dominant and a recessive bit. The Optical CAN–bus is based on light in the central tube as the dominant bit. Such a system is free from interference, and there are no problems with moving contacts, which otherwise make rotating discs problematic. Most of the CAN–controllers are implemented in the FPGAs, and some by the use of standard circuits. The plan is to use FPGAs for all controllers, and to modify the CAN–protocol to suit the needs for robotics.

4 The Strategy project

The Aros robot's Sense–Plan–Act functionality depends on a variety of sensors, such as the camera disc and the gyro, to collect information about the surroundings. The robots will also mediate information between each other using the inter–robot communication functionality. All information is processed in the sensor–fusion part of the robot where it establishes its own world view. Based on this knowledge it will make decisions about what to do and what it wants other robots to do. For example, the robot holding the ball can tell another robot to place itself in position to wait for a passing of the ball. There is no central decision–making computer, but the robots agree on the strategy between themselves, decisions like choosing a position on the field, forward or defence, depending on what the other robots are doing.

Each robot's decision–making algorithm will use the principle of reinforcement learning and case–based reasoning. Each case consist of many different parameters, such as

1. Team mates and their position.
2. Opponents and their position.
3. Ball position.
4. Who has the ball.
5. If we are attacking.
6. Case weight.
7. The appropriate action.

The weight shows how efficient the case has proven earlier. The reinforcement learning methods are used to update the case weight. This way, the robots will learn what strategies pay the best dividends and give them higher priority than the strategies that are less effective.

To select a case, the robot tries to match the world view to a case within the case library. If the robot fails to do so, it creates a new case. When several cases match the world view it is most likely that the robot will select the one with the highest weight.

The real–time system is written in Ada and follows the Ravenscar profile to become as robust and error proof as possible.

Positioning

There will be two independent dead reckoning systems in the robot. Each of the three motors (Maxon) are equipped with 500 pulse digital encoder, and a dedicated FPGA will handle both the speed regulation of the motors using pulse width modulation of an H–bridge. The output from the encoders will simultaneously be used to track all movements of the robot (rotation and translation). Apart from this, one of the discs will act as a gyro. These gyro–discs will have two two–axis–accelerometers. The accelerometers will be mounted diagonally on the disc each with a radial and a tangential component. The stepper motor will be used to keep the integrated value of the difference of the tangential sensors at zero. By doing this the disc will act as a gyro and always be aligned with the field. By measuring the angle between this disc and the robot a value of the robots orientation will be achieved. Furthermore by a double integration of the sum of the radial and the tangential components an estimation of the robots position is also obtained.

Computer system

A robot is definitely a real–time system, and it is therefore amazing to see so many implementations relying on operating systems that are not real–time systems, such as Linux or Windows NT. The computer system in the Aros robot is based on the latest Virtex Pro FPGA from Xilinx. In a donation the Aros team has received FPGAs containing 1 million gates and a PowerPC. The PCB used to hold this device there is also 64 MByte of RAM and 32 MByte of Flash memory. The reconfigurable hardware will be used to hold a dedicated operating system. The kernel supports all necessary functionality of the Ravenscar profile of Ada. Apart from the run-time system the FPGA will hold CAN and USB functionality.

References

1. J. Andrén–Dinerf, L. Asplund, C. Benson, H. Bergkvist, M. Brunk, M. Pettersson, C. Sjöö, P. Sundin, M. Wall and K. Wijk.: ChipVision – A Vision System for Robots Based on Reconfigurable Hardware, `http://www.robocup.it.uu.se/pdf/chipvision.pdf`
2. S. Ben Yacoub and J. M. Jolion.: Hierarchical Line Extraction, IEEE Proceedings – Vision, Image and Signal Processing, vol. 142, no. 1, pages 7–14, Feb. 1995
3. J. Princen, J. Illingworth and J. Kittler.: A Hierarchical Approach to Line Extraction Based on the Hough Transform, Computer Vision, Graphics and Image Processing, vol. 52, pages 57–77, 1990.
4. P. Banerjee, N. Shenoy, A. Choudhary, S. Hauck, C. Bachmann, M. Haldar, P. Joisha, A. Jones, A. Kanhare, A. Nayak, S. Periyacheri, M. Walkden and D. Zaretsky.: A MATLAB Compiler For Distributed, Heterogeneous, Reconfigurable Computing Systems, Electrical and Computer Engineering, Northwestern University, Illinois, USA, Technical Report.
5. A. Nayak, M. Haldar, A. Choudhary and P. Banerjee.: Precision and Error Analysis of MATLAB Applications During Automated Hardware Synthesis for FPGAs, IEEE, 2001.
6. W. Luk, P. Andreou, A. Derbyshire, D. Dupont–De Dinechin, J. Rice, N. Shiraz and D. Siganos.: A Reconfigurable Engine for Real–Time Video Processing.