# NuBot 2003 team description paper

Liu Lin[1], Xiucai Ji[2], Liu Fei[3], and Zhiqiang Zheng[4]

Department of Automatic Control
National University of Defense Technology, China
[1] liu_double@263.net
[2] jxc_nudt@hotmail.com
[3] liufei_only@263.net
[4] xyzheng@sohu.com

**Abstract.** This paper describes the NuBot 2003 RoboCup small-size league team, which is a new comer of this domain. And we are preparing this team for the 2003 competition. This paper gives a general view of the implementation of our current robot team. These aspects include both the hardware design of the robot vehicle and the AI software running on the host computer.

## 1 Introduction

The NuBot 2003 team is a new comer of the coming 2003 RoboCup small-size competition. We develop this team from our previous Mirosot small-size team. This team includes five field robots of which four serve as common players and one servers as the goalie.

In the current implementation, all the five robots are the same from the mechanical aspect of view. They are all with 2-wheeled architecture, and have special designed kicking and dribbling mechanics. This robot is demonstrated in figure 1.

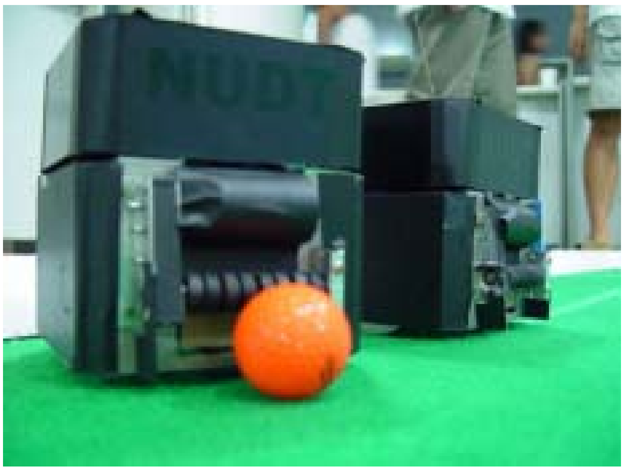

**Fig. 1.** Our current robot. The robot is a 2-wheeled architecture with both kicking and dribbling mechanics. The kicking and dribbling mechanics are separated modules and can be easily taken down or installed.

As this kind of 2-wheeled architecture is no longer used by most of the advanced teams, now we are only using them to test our host program and vision system. And we are now busy developing a new version of robot vehicle, which is omni-directional, with both kicking and dribbling mechanics. This new robot team will be available soon.

Each robot is equipped with a TI DSP TMS320F240, which is the first member of a new family of DSP controllers based on the TMS320C2xx generation of 16-bits fixed-point digital signal processors (DSPs). This new family is optimized for digital motor/motion control applications. The DSP controllers combine the enhanced TMS320 architectural design of the ‘C2xLP core CPU for low-cost, high-performance processing capabilities and several advanced peripheral optimized for motor/motion control applications. These peripheral include the event manager module, which provides general-purpose timers and compare registers to generate up to 12 PWM outputs, and a dual 10-bit analog-to-digital converter (ADC), which can perform two simultaneous conversions within $6.1\ \mu s$.

The communication between the robots and the host computer is via a communication module: Radiometrix tx2rx2 data link module which can be 418 or 433 MHz.

The host computer program processes the vision information grabbed by a high-speed machine vision frame grab card with a PCI interface and sends appropriate commands to the robot vehicles. The host computer program is called AI (Artificial Intelligence) program, which includes six modules: overall control module, vision module, strategy module, communication module, simulation module and virtual play ground module.

## 2 System Description

Currently we employ a global-vision-based system. The vision system for the project consists of an overhead camera that has been mounted and interfaced to a high-speed frame grabber in a PC. The camera is about three meters overhead from the play ground.

The host computer provides all the intelligence for all the robots. It processes the vision data from the camera, and computes the next move for each robot. Via a RF transmitter, the commands are directly transmitted to the robots. These robots are only executors of the host commands and their task is only to provide a solid servo loop and reliable communication.

The system architecture is demonstrated in figure 2.

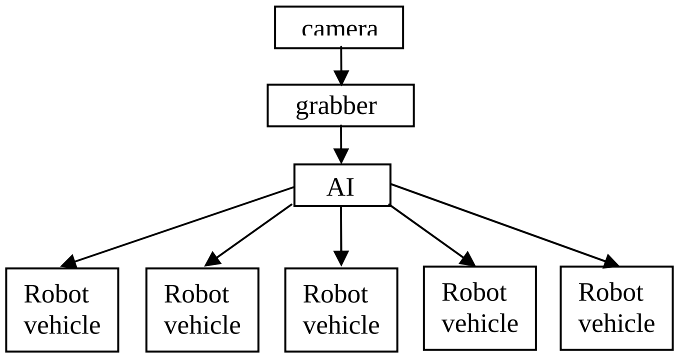


**Fig. 2.** Architecture of the whole system.. The camera continuously captures the playing information and feeds it to the grab card through an AV wire. The grab card in mounted in the host computer. AI program analyses the vision data and picks out the key information. With the key information, commands for each robot is generated and broadcasted to them via the RF module.

## 3 Robot Design

As our current robots team is with two-wheeled architecture, and we are not planning to use them to attend the coming 2003 RoboCup small-sized competition, here we won't describe the mechanical implementation of it in detail. We only want to give some words on our kicking mechanism implementation. And in order to attend the competition, we are now busy developing a new version, which is omni-directional, with both kicking and dribbling mechanics. We are sure that the new version of our robots team can soon be available and will play a better performance than our current one.

The robot's lower controller employs a TI DSP TMS320F240 and an Altera CPLD. The DSP controllers combine the enhanced TMS320 architectural design of the 'C2xLP core CPU for low-cost, high-performance processing capabilities and several advanced peripheral optimized for motor/motion control applications. This DSP controller has an event manager module, which provides general-purpose timers and compare registers to generate up to 12 PWM outputs.

In order to spend less CPU time of the DSP, we use an Altera CPLD to server as several 16-bits counters. These counters can calculate the counter pulses of the motors, and provide the DSP with these data if needed. In this way, the DSP can focus on the controlling of the motors, and thus provide a solid servo loop.

The CPLD can also provide the robot vehicle necessary logic such as the logic needed for the H-bridge. With the PWM pulse provided by the DSP and some necessary logic, the H-bridge can drive the motors as wanted.

Another key point is that our controller board is designed to be a module. That is, several controller boards can be mounted together to provide more driving capability.

Each controller board is equipped with a standard PC/104 bus connector. This bus maybe has no use currently, but in the future, if we want to develop a local vision robot team, we need only plus this controller board to the embedded PC/104 micro-computer, and it will work well.

A kicking mechanism is also implemented in the robot hardware. As we know, most of the F180 teams implement their kicking mechanism using the solenoid system, and this kind of design can successfully accomplish the kicking task. In order to kick the ball at a more fast speed, people use higher and higher voltage (some even use voltage of above 100 voltages). As we think such a higher voltage in such a small robot playing game in a ground is not so safe for the control circuit, we still stick to the spring-gear and motor implementation.

## 4 AI Design

As the spirit part of the whole system, the AI program works in the host computer of P4 2.0G. Our AI program is made up of six modules: overall control module, vision module, strategy module, communication module, simulation module and virtual play ground module. The software architecture is demonstrated in figure 3.

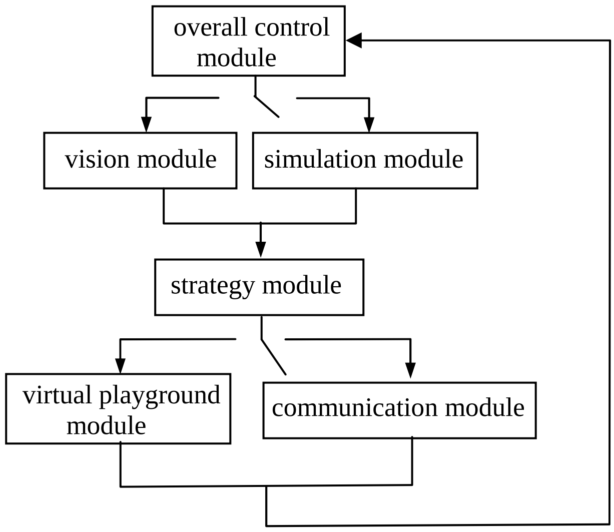

**Fig. 3.** AI software architecture. The AI software serves as the spirit of the whole system and includes six parts: overall control module is on duty of the management of all other modules; vision module processes the raw vision data and pick out the key information of the playground; strategy module parses the key information and generates commands for each robot; communication module sends the above commands to the robots; virtual playground module displays the playing information and can also record, replay the information; simulation module provides the system a simulation environment so that the strategies can be tested before implemented to the real robots.

The AI software is developed with visual 6.0 in a multi-thread manner. Each module is implemented as a separate thread. The overall control module serves as the main thread and starts before all other threads. Though this module does not do any computation in the processing period, it can be regarded as the manager of the whole program, and all the other threads are designed as worker threads. Also the overall control thread works as the man-machine interface in the configuration period.

The vision module is on duty of processing the vision information grabbed by the video frame grab card. The frame grab card is mounted in the PCI interface and can work parallel with other modules. Here “parallel” means that the grabbing processing almost spends no CPU time (only about 5% of CPU time). And the grabber continuously puts the frame data to the specific memory in a speed of about 30f/s. The task vision module should do is to configure the grab card so that it can work as expected. The mayor task of the vision module can be divided into three parts: obtaining the frame data from the specific memory, processing the data and picking up the key information of the current playground, feeding the key information to some specific memory so that the strategy module can use it.

The strategy module can be regarded as the head of the soccer robots. It processes the information provided by the vision module and generates commands which decide what each robot should do. Through a communication module these commands can te broadcast to the robots on the playground. One important issue is the delay problem. When the strategy module is processing some data gotten from the vision module, in fact, this is not the very current information. Due to different hardware and software implementations, the latency is also different. Sometimes the latency can be more than 100ms. This means that what the strategy module is processing is not the “now” data, but data some time ago. The latency comprises several parts: time spend in the camera figure capture period, time spend in the grab card processing (A/D) period, time spend in the vision information kicking up period, time spend in the commands generating period and time spend in the commands broadcast-receive period. Taken into all of the above latency factors separately is impossible and also unnecessary. So we just take some experiments and have an estimate from the experiment data. In the strategy module, we take into account of the estimated latency and have a prediction from the gotten data to compensate the delay problem.

The virtual playground module serves as a display thread. It continuously displays the robots and ball information gotten from the vision module. It can also record all the match period information so that we can “replay” the game and study the parts we are interested in.

The AI software can not only used in the real robot team competition, but also provide a simulation module with which a simulation competition can be carried out.

With this simulation module, some strategy can be tested before implemented in the real robot competition. Thus we can not only save much time, but also prolong the robot's life.

## 5 Summary

So far, we have developed a RoboCup small-sized team, and we are using it to do some research work. But the we have not completed our work. We are now busying developing a new version of omni-directional robot vehicle, and will use it to attend the coming competition.

## References


1. Liu Lin, Xia Min, Zhiqiang Zheng: Implement of a RoboCup small-size robot system . Proceedings of the world Congress on Intelligent Control and Automation (WCICA), Vol 1, IEEE(2002) 483-487
2. TMS320C240/F240 DSP Controllers data sheet, Texas Instrument incorporated (1998)
3. Low Power UHF Data Transceiver Module, Radiometrix LTD (2001)
4. Raffaello D'Andrea, Jin-Woo lee, Dennis Huang: Cornell University RoboCup: Big Red Bots final design document Team Brazil 1999