The Design of an Intelligent Soccer-Playing Robot

Dan Xiong · Junhao Xiao · Huimin Lu · Qinghua Yu · Zhiwen Zeng · Kaihong Huang · Shuai Cheng · Xiaodong Yi · Zhiqiang Zheng

Abstract

Purpose - In the RoboCup Middle Size League (MSL), two teams of five autonomous robots play on an 18 x 12 meter field. Equipped with sensors and onboard computers, each robot should be able to perceive the environment, make decisions, and control itself to play the soccer game autonomously. The purpose of this paper is to design intelligent robots operating in such dynamic environments like the RoboCup MSL.

Design/methodology/approach - We present the design of our soccer robots, participating in RoboCup MSL. The mechanical platform, electrical architecture, and software framework are discussed separately. The mechanical platform is designed modularly, so easy maintainability is achieved; the electronic architecture is built on industrial standards using PC-based control technique, which results in high robustness and reliability during the intensive and fierce MSL games; the software is developed upon the open source Robot Operating System (ROS), thus inheriting the advantages of ROS as modularity, portability, and expansibility.

Findings - Based on this paper and our open source hardware and software, our MSL robots can be re-developed easily to participate in RoboCup MSL. Our robots can also be employed in other research and education fields, especially for multi-robot systems and distributed artificial intelligence. Furthermore, the main designing ideas proposed in the paper, i.e. using a modular mechanical structure,
an industrial electronic system and ROS-based software, provide a common solution for designing general intelligent robots.

**Originality/value** - The methodology of the intelligent robot design for highly competitive and dynamic RoboCup MSL environments is proposed.

**Keywords** Soccer robots, Mechanical structure, Electronic architecture, Software, ROS

**Paper type** Research paper

---

### 1 INTRODUCTION

RoboCup provides a standard test bed for the dissemination and validation of innovative theories, algorithms, and agent architectures, which has promoted artificial intelligence and robotics for almost two decades (Kitano et al, 1997). The final goal of RoboCup is that a team of fully autonomous humanoid soccer robots will beat the human World Cup champion team by 2050.

At present, RoboCup is made up of RoboCup Soccer (Simulation League, Small Size League, Middle Size League, Standard Platform League), RoboCup Rescue, RobotCup@Home, RoboCup Junior, etc. Different robots are designed for these competitions. The NAO robot developed by the French company Aldebaran-Robotics has been deployed in the RoboCup standard league (Gouaillier et al, 2009). Researchers can build their software based on the Robot Operating System (ROS)\(^1\) for NAO robots. Tracked robots are good at locomotion, which are developed to search and rescue in large-scale disasters for RoboCup Rescue (Kadous et al, 2006). For RobotCup@Home, human-robot interaction and speech understanding are more important. Chen et al (2013) developed the robot KeJia, which was based on a two-wheels driving chassis. And the software architecture of KeJia was based on ROS.

For the RoboCup Middle Size League (MSL), all the robots can be designed freely as long as they stay below a max size and a max weight. And they are distributed and fully autonomous, which means all the robot sensors are on-board, and robots should be able to process the sensor information and realize decision, motion planning and control by themselves. Wireless communication can be used to help cooperation and coordination with teammates. As shown in Fig. 1, the MSL game is highly competitive and dynamic, for example, on an 18 x 12 meter field, velocities of up to five meters per second are reached. Lots of research subjects are included in MSL, such as mechanical design, electric system design, visual perception, real-time reasoning, motion control and multi-robot cooperation.

Founded in 2004, our NuBot MSL team has been participating RoboCup actively since 2006 in Bremen, Germany. In this paper, we will share the whole design of our forth generation NuBot robots, especially, mechanical platform, electronic

---

\(^1\)http://www.ros.org.
The Design of an Intelligent Soccer-Playing Robot

system and software based on ROS, which are made open source\(^2\). Therefore, others who want to participate the RoboCup MSL can avoid some repetitive low-level work and develop their soccer robots quickly based on our experience. With less time and energy spent on the design of basic robot hardware and software, more attention can be paid to multi-robot cooperation, coordination and other high level researches. Our robots can also be re-developed for universities and laboratories for research and education purposes. Meanwhile, the NuBot robots can be developed further to become a standard MSL platform. The ideas of developing our robots, using a modular mechanical platform, an industrial electrical system and ROS-based software, provide a valuable reference to design general intelligent robots.

The rest of this paper is organized as follows. The related work is introduced in Section 2. The mechanical platform is introduced in Section 3. In Section 4, we describe the industrial electrical system. Then the software based on ROS is presented in Section 5. Finally, the paper is summarised in Section 6, which also states the future work.

2 Related work

To design intelligent robots for highly competitive and dynamic environments like the RoboCup MSL, lots of hardware designs and software algorithms have been proposed. For example, in order to create a RoboCup MSL standard platform, the Tribots team from the University of Freiburg proposed the DFG Project: Design of a standard platform (Hardware and Software) for the RoboCup MSL league as an open-source project. In this section, it is impossible to give a comprehensive introduction about all the related work. Instead, we just try to give a brief introduction

\(^2\)The ROS-based software has been committed to GitHub (https://github.com/nubot-nudt), and the materials of the mechanical platform and the electronic system are being collected and revised, which will be uploaded to the website (http://www.nubot.com.cn) in the near future.
from which readers can acquire more details about the achievements in MSL, and the efforts which have been done to cut down the difficulty in developing MSL robots. Our previous works about RoboCup MSL robots also will be introduced in this section.

A special issue on “Advances in intelligent robot design for the Robocup Middle Size League” was published by Mechatronics (Van De Molengraft and Zweigle, 2011). In this issue, the state of the art research about mechatronics and embedded robot design, vision and world modelling algorithms, and team coordination and strategy were presented. Some surveys about team strategies, vision systems, and visual perception algorithms in robot soccer can be found in Nadarajah and Sundaraj (2013a); Nadarajah and Sundaraj (2013b); Li et al (2013).

Taking into account that the design of soccer robots from the very beginning is time consuming and nontrivial, several efforts have been done to reduce the difficulty in implementing MSL robot systems in the robotics/RoboCup community. In hardware, Robotic Open Platform (ROP) launched by Eindhoven University of Technology (TU/e) provides a place to discuss and share robot hardware designs (Lunenburg et al, 2014), and The TU/e team also shared their hardware and software on ROP. Together with its industry partners, the TU/e team develops the low-cost TURTLE-5k platform, based on their TURTLE robots for RoboCup MSL. They employ the Value Engineering method to seek out some functions with the most cost and decide where they should reduce costs during developing their TURTLE-5k robots. Leng and Cao (2009) focused their attention on the holonomic wheeled platform, and analysed the anisotropy of the holonomic mobile robots. The results were employed in motion planning for MSL robots. In Azevedo et al (2014), the electronic architecture of their MSL robots including a set of microcontrollers interconnected through a CAN network is implemented to develop their low-level sensing/actuation system.

In software, during the past few years, ROS has become popular in robot software programming. By using ROS, robot software components can be well and easily organized, and the high modularity and re-usability of the codes can be achieved. Several teams have developed their software systems based on ROS and made them open source.

Our NuBot MSL team has been built for about 10 years, and some research results have been obtained. In order to deal with increasingly fierce soccer competitions, we already designed several kinds of robot platforms in the past. We mainly focused our attention on the motion ability especially velocity and acceleration. However, we ignored anti-impact safety apparatus, and the ball handling mechanism usually was damaged (Yu et al, 2010). We also designed our omnidirectional vision system for MSL robots (Lu et al, 2009, 2011), and proposed a camera parameters auto-adjusting technique for our vision system (Lu et al, 2010). For distributed multi-robot cooperation, we proposed a cooperation framework to meet MSL competition requirements based on market and capability classification (Lin and Zheng, 2005a,b). We dedicatedly designed a second controller DSP (Digital Signal Processor) for our control system in the past. However, the designed controller is not very stable, and sometimes can malfunction (Yu et al, 2010).
Although many researchers have made great efforts to develop MSL robots, some problems still have not been solved. Some challenges in the RoboCup MSL are shown as follows.

1. The robot platform should have good performance in critical aspects such as top speed and top accelerations, and be able to handle impacts. It should be easy to assemble and maintain.
2. It is necessary to improve the stability of the electrical system, and the extension of sensors should be better supported.
3. The robotic software should be robust and real-time for RoboCup MSL competitions. Especially, the robustness of the robot vision system should be improved to make it work reliably in indoor and outdoor environments with highly dynamic lighting conditions. The software framework should be universal and reusable as much as possible.

In this paper a modular robot platform, a PC-based control technique and a ROS-based software framework are used to solve these problems while designing new RoboCup MSL robots.

3 Mechanical design

This section describes the mechanical design of our soccer robots. When designing the robot platform, there are several criteria to be considered. Firstly, it should comply with the rules and regulations of RoboCup MSL, namely its size, weight and safety concerns. Secondly, it should have excellent performance in critical aspects such as top speed and top accelerations. Lastly, since malfunctions or failures are unavoidable during the intensive and fierce MSL games, mechanical parts of the robot should embrace high modularity such that they are easy to assemble and maintain. It is not trivial to fulfill these criteria, therefore we make our mechanical design, which satisfies above criteria and has been tested in real games, open source to help others develop soccer robots quickly and easily.

In industrial design, the modularity refers to an engineering technique that builds larger systems by combining smaller subsystems. Our whole MSL robot is subdivided into several modules (subsystems) according to different functions, and these modules can be independently created and then used in different RoboCup MSL robots with some small changes. Currently the regular robot and the goalie robot are heterogenous due to their different tasks. For a regular robot, it should be able to do the same things as a human soccer player, such as moving, dribbling, passing and shooting. Therefore, the mechanical platform is subdivided into five main modules: the base frame, the ball handling mechanism, the electromagnet shooting system, the omnidirectional vision system and the front vision system, as illustrated in Fig. 2(a). For the goalie robot, the ball handling mechanism, the electromagnet shooting device and the front vision system are removed, instead two RGB-D cameras are integrated as shown in Fig. 2(b).

\(^5\)We have shown these tasks in our qualification video for RoboCup MSL 2015, which can be found in http://www.nubot.com.cn/2015videoen.htm.
3.1 Base frame

The holonomic wheeled platform, which is capable of carrying out rotation and translation simultaneously and independently, has been employed by most MSL teams (Aangenent et al, 2009; Neves et al, 2010). In our omnidirectional wheeled platform, we use custom-designed omnidirectional wheels, which are illustrated in Fig. 3(a). Four such omnidirectional wheels are uniformly arranged on the base as shown in Fig. 3(b). Despite the added costs of extra weight and extra power consumption, the 4-wheel-configuration platform can generate more traction force than a normal 3-wheel-configuration one, resulting in a boost in average speed and average acceleration respectively. Their motion control methods are similar and can be mainly divided into two categories: kinematic model based control and dynamics model based control (Ashmore and Barnes, 2002).
3.2 Ball handling mechanism

The ball handling mechanism enables the robot to catch and dribble a ball during the game. As illustrated in Fig. 4, there are two symmetrical assemblies, and each contains a wheel, a DC motor, a set of transmission bevel-gear, a linear displacement transducer and a support mechanism. The wheels are driven by the DC motor and are always pressed to the ball, therefore they can generate various friction forces to the ball, and make it rotate in desired directions and speeds together with the soccer robot. During dribbling, the robot will constantly adjust the speeds of the wheels to maintain a proper distance between the ball and the robot using a closed-loop control system. This control system takes the actual ball distance as the feedback signal, which is measured indirectly by the linear displacement transducers attached to the supporting mechanism. As the ball moves closer to the robot, the supporting mechanism will raise, then compress the transducer, otherwise the support mechanism will fall and stretch the transducer. The information obtained from two transducers can be used to calculate the actual ball distance based on a given detailed geometry model. This system effectively solves the ball handling control problem.

Fig. 4 The ball handing mechanism of the NuBot.

3.3 Electromagnet shooting system

The shooting system enables the robot to score goals and can be subdivided into three categories: spring mechanisms, pneumatic systems and solenoids (Zandsteeg and van de Molengraft, 2005). When using spring mechanisms, the shooting power is quite hard to control. The pneumatic systems usually need a large gas tank to generate high pressure to realize strong shooting, and the number of shots generally depends on the size of the gas tank. If choosing the solenoid, the shooting system can be powerful and lightweight, and it is also quite easy to control the shooting power. Our shooting system is basically a custom-designed electromagnet
with a high impulsive force. As depicted in Fig. 5, it consists of a solenoid, an electromagnet core, a shooting rod, a capacitor, and two linear actuators with potentiometer. The shooting rod can be adjusted in height to allow for different shooting modes, namely flat shots for passing and lob shots for scoring. Two modes are realized using two linear actuators to move the hinge of the shooting rod to different positions. Initially, the electromagnet core is rearward located within the solenoid and the capacitor is charged. When the shooting action is activated, the rod will be adjusted according to the currently selected mode. Then the control circuit board will switch on the solenoid by discharging the capacitor, thus produce a strong electromagnetic force to push forward the rod. The rod then strikes the ball and delivers momentum to it. After the shooting is finished, the core will be pulled back to its initial position by an elastic stripe and the capacitor will be recharged again and wait for the next shooting action. Therefore, this system is simple yet capable of various shooting angles.

![Fig. 5 The electromagnet shooting system of the NuBot soccer robot.](image)

3.4 Omnidirectional vision system

The omnidirectional vision system is composed of a convex mirror and a camera pointing upward towards the mirror (Kasaei et al, 2010; Lu et al, 2011). The panoramic mirror plays the most important impact on the imaging quality, especially on the distortion of the panoramic image. We designed the panoramic mirror using the 3dmax software which can be used to simulate imaging results of the omnidirectional vision system. Our panoramic mirror is made up of hyperbolic mirror, horizontally isometric mirror and vertically isometric mirror from the inner to the outer (Lu et al, 2011). The designed profile of mirror and manufactured mirror are shown in Fig. 6. The typical images captured by the omnidirectional vision system in a MSL standard field are showed in Fig. 10. The system not only makes the imaging resolution of the objects near the robot on the field constant and the imaging distortion of the objects far from the robot small in the vertical...
direction, but also enables the robot to acquire a very clear image of the scene which is very close to it, such as the robot itself.

Fig. 6 The designed profile of mirror and the manufactured mirror.

3.5 Front vision system and RGB-D camera

The front vision system and the RGB-D camera are auxiliary sensors for the regular robots and the goalie robot, respectively. The front vision system is a low-cost USB camera, which is tilted down upon the ground. With it, the robot can recognize and localize the ball with high accuracy when the ball is close to the robot. The position of the ball is estimated based on the pinhole camera projection model. It is of great significance for accurate ball catching and dribbling. In the current MSL games, most of the goals are achieved by lob shooting, so accurate estimation of the shooting touchdown-point of the ball is fundamental for the goalie robot to defend these shoots. Although the object’s 3D information can be acquired using the omnidirectional vision system and the front vision system together equipped in our regular robots, the accuracy cannot be high, because the imaging resolution of the omnidirectional vision is quite low to achieve large field of view. The RGB-D camera like Kinect can output color and depth stream simultaneously at the frame rate of 30 fps, and its maximal sensing range can be up to 10m, which makes the RGB-D camera be the ideal sensor to obtain the 3D ball information for the goalie robot. Thus our goalie robot is equipped with two RGB-D cameras, as demonstrated in Fig. 2(b), to recognize and localize the ball, estimate its moving trace and predict the touchdown-point in 3D space.

4 Industrial electrical system

The NuBot soccer robot is a platform that contains different sensors, controllers and actuators. The sensors include an omnidirectional vision system, four motor encoders, two linear displacement sensors, a front vision system for the regular robot and two RGB-D cameras for the goalie robot. The controllers consist of
a Beckhoff industrial PC, six ELMO motor drives and a shooting module. The actuators contain a shooting system, a ball handing mechanism and a base frame. The PC-based control technology is becoming popular in industrial automation system because it can provide industrial level stability. In this section, we use this kind of technology to build the industrial electrical system to support our robot sensors, controllers and actuators.

As the MSL game becomes more and more competitive and fierce, the requirements on the robustness and reliability of the electronic system are also increasing. The electrical system for last generation NuBot robots consists of an industrial PC, a second controller, the I/O modules, the sensors and the motion controllers. The industrial PC is the core module, and the secondary controller is dedicated hardware for the automation tasks. The form of communication established between them is serial communication based on the RS232 protocol. Some disadvantages exist in this kind of electronic system using a secondary controller. Firstly, the secondary controller is hard to maintain. And the dedicated hardware has not been adequately tested according to some industrial standards, so the stability of the electrical system also decreases. Secondly, the expansion of sensor and actuator modules is restricted by only a limited number of interface components. Finally, the bandwidth is limited by serial transmission.

In recent years, the risk of the fierce collision between robots increases in highly dynamic MSL competitions. To improve the robustness of our robot control system, we design our current electrical system using PC-based control technology as shown in Fig 7. Due to steadily growing processing power, PC can work as an ideal platform for automation. It enables automation tasks to be performed through software without dedicated hardware (Harris and Beckhoff Automation, 2004). All control system and visualization tasks can be carried out by a powerful central CPU and decentralized I/Os, thus the electrical system is highly scalable. For example, the limitation on the number of I/O modules, sensor modules and actuator modules is only dependent on the CPU processing power. In addition, the system employs the Ethernet-based fieldbus system EtherCAT and the TwinCAT system to realize high speed communication between industrial PC and the connected modules. Furthermore, the electrical system also realizes the effective utilization of high-performance multi-core processors in industrial PCs.

By using the PC-based control technology mentioned above, the schematic diagram of the NuBot electrical system is shown in Fig. 8. All vision and control algorithms are processed on the industrial PC. The industrial PC communicates with the EtherCAT system via Ethernet. The Elmo Motion Control (SOL-WHI 20/60) is the intelligent miniature digital servo drive for the 150W DC brushless motor. The CANopen modular EL751 embedded in the EtherCAT is used to realize communication between the industrial PC and the Elmo Motion Controls. Our shooting module, also named as kicker driver, is mainly composed of a relay and an IGBT FGA25N120ANTD. The PC can send control signals to the kicker driver for shooting or passing via the EtherCAT.

The industrial electrical system was tested through 2014 Brasil and 2015 Hefei international RoboCup competitions, 2014 China RoboCup competition and there was no fault happening during these three events. So the system can meet the demands of the RoboCup MSL competition and provide a good solution for the design of an intelligent robot.
The Design of an Intelligent Soccer-Playing Robot

5 Software based on ROS

The recent achievements in robotics make autonomous mobile robots play an increasingly important role in daily life. However, it is difficult to develop a generic software for different robots. For example, debugging usually is necessary and difficult to employ others robotic software. To make robotic software develop high-efficiency, some robotic software development platforms come into being, e.g. Microsoft Robotics Developer Studio (MRDS) (Jackson, 2007), LabVIEW Robotics (Johnson, 1997), and ROS. These platforms allow to directly use the software written by others with minimal debugging. MRDS is a Windows-based software platform with C# as its main programming language. LabVIEW Robotics is developed by National Instruments (NI), and supports Windows, Linux, Mac and Unix. However, MRDS and LabVIEW Robotics are not open source. ROS, launched by
Willow Garage company, provides a set of software libraries and tools for building robot applications across multiple computing platforms. ROS has many advantages: ease of use, high-efficiency, cross-platform, supporting multiple programming languages, distributed computing, code reusability, and being completely open source (BSD) and free for others to use. Furthermore, our software is developed on Ubuntu, and it is also open source. Therefore, we also use ROS to build our NuBot software. The operating system is Ubuntu 12.04, and the version of ROS is groovy.

As shown in Fig. 9, the software framework is divided into 5 main parts: the Prosilica Camera node and the OmniVision node; the UVC Camera node, the FrontVision node and the Kinect node; the NuBot Control node; the NuBot HWControl node; the RTDB and the WorldModel node. Two Kinect nodes replace the FrontVision node and the UVC Camera node for the goalie. These nodes will be described in the following sub-sections.

![Fig. 9](image)

**Fig. 9** The software framework based on ROS.

5.1 The OmniVision node

The perception is the basis to realize the autonomous ability such as motion planning, control decision and cooperation for mobile robots. Omnidirectional vision is one of the most important sensors for RoboCup MSL soccer robots. The image is captured and published by the Prosilica Camera node\(^6\). It takes about 30ms to perform these computation, so the OmniVision node can be run in real-time.

5.1.1 Color segmentation and White line-points detection

The color lookup table is calibrated off-line. Because of its simplicity and low computational requirements, it is used to realize color segmentation. A typical

\(^6\)http://wiki.ros.org/prosilica_camera.
panoramic image captured by our omnidirectional vision system is shown in Fig. 10(a) in a RoboCup MSL standard field. The results of color segmentation for Fig. 10(a) are demonstrated in Fig. 10(b). We can conclude that this method can be used to distinguish ball, green field, black obstacles and white line-points in the color-coded environment. To detect white line-points in the panoramic image, we search for significant color variations along some scan lines because of the different color values between the white lines and the green field. As shown in Fig. 10(b), these scan lines are radially arranged around the image center, and the red points represent the obtained white line-points.

![Fig. 10](image1.png)  
(a) The image captured by our omnidirectional vision system. (b) The result of color segmentation for image (a). In (b), the white lines, which are radially arranged around the image center, are some scan lines. The red points represent the obtained white line points, and the purple areas represent some candidate obstacles.

5.1.2 Self-localization

The localization for an autonomous mobile robot under highly dynamic structured environments is still a challenge. Matching optimization localization algorithm, which can be employed to find the locally best match between the detected white line points and the field lines, is used to realize localization tracking and global localization for our soccer robots quickly and accurately (von Hundelshausen et al., 2003; Lauer et al., 2006; Xiong et al., 2012). For global localization, every robot needs to localize itself without any prior information about its position and orientation. We obtain the robot’s orientation by Motion Trackers instrument (MTi). Then we acquire 315 samples as the robot’s candidate positions which are located uniformly in the field. Finally, the match optimization localization algorithm is employed to optimise these candidate samples and find the robot’s real position (Xiong et al., 2012). For localization tracking, we usually suppose the initial pose is known. The odometry information is updated to obtain its coarse pose according to four motor encoders. Then we employ the match optimization localization algorithm to optimise the coarse pose. Finally, we obtain the robot’s
real pose with Kalman Filter to fuse the information from the odometry and the optimized pose. The robot’s self-localization results are demonstrated in Fig. 11. During the experiment, the robot was pushed by people to follow some straight tracks on the field shown as the black lines in Fig. 11. The red traces depict the robot’s self-localization results. The mean position error and orientation error of the robot self-localization are smaller than 6 cm and 0.045 rad respectively. We can conclude that the robot is able to achieve good localization results.

Fig. 11 The robot’s self-localization results.

5.1.3 Obstacle and Ball detection

Owing to relative easiness of detecting the obstacles and the ball based on the result of color segmentation, we mainly introduce multi-target tracking for obstacles and the estimation of ball velocity. It remains as a popular research theme for soccer robots under highly dynamic environments. For multi-target tracking, we firstly utilize a scan-line approach to determine the positions of these obstacles, which is similar as the white line points detection method mentioned above. The “Current” statistics model is used to describe the current probability density of the target maneuvering, which can adjust the model parameters according to the target maneuvering and is appropriate for characterizing the high-speed targets (Zhou and Kumar, 1984). Considering the physical relation between the acceleration estimation and the mean value of the state noise for the obstacles in the field, we utilize a “Current” statistics model to establish the state models for tracking moving obstacles. Based on this model, an adaptive Kalman filter with the maximum acceleration constraint is employed to realize single target tracking (STT). Finally, the Joint Possibilities Data Association algorithm (JPDA) is employed to associate data with targets in highly dynamic multi-target environments (Fortmann et al, 1983; Chang et al, 1986). We combine STT and JPDA to realize...
robust and accurate multiple obstacles tracking. The ball velocity is important information for the goalie’s shoot defense, ball passing and intercepting in multi-robot cooperation. After acquiring a series of ball positions and timestamps, the ball velocity can be estimated through a linear least square method after assuming that the ball velocity is constant during a short period of time like hundreds of milliseconds (Lauer et al, 2005).

5.2 The FrontVision node and the Kinect node

The FrontVision node processes the perspective image captured and published by the UVC Camera node\(^7\), and provides the more accurate ball position information for the regular robot. There are several premises needed to be considered. Firstly, the ball should be located on the ground. Secondly, the pinhole camera model is adopted to calibrate camera interior and exterior parameters off-line. Lastly, the height of the camera to the ground and the horizontal view angle of the camera are known. The node detects the ball using a color segmentation algorithm and region growing algorithm similar to the OmniVision node. Then we can estimate the position of the ball on the ground according to the pinhole camera model.

The 3D information of the ball is of great significance for the goalie robot to intercept the lob ball (Lu et al, 2014). However, the front vision system and the omnidirectional vision system cannot obtain depth information directly. Therefore, we make use of two RGB-D cameras to recognize and localize the ball and estimate its moving trace in 3D space. ROS provides the RGB-D camera driver and integrates the Point Cloud Library (PCL). The color segmentation algorithm that is the same as in the OmniVision node is employed to obtain some candidate ball regions. Then the random sample consensus algorithm (RANSAC) (Schnabel et al, 2007) is used to fit the spherical model using the 3D information of these candidate ball regions. With the proposed method, only small amounts of candidate ball regions need to be fitted. Lastly, to intercept the ball for the goalie, the 3D trajectory of the ball regarded as the parabola is estimated and the touchdown-point in 3D space is also predicted (Lu et al, 2014).

About the real-time performance, it takes about 30-40ms to process a frame of perspective image and RGB-D data in the FrontVision node and the Kinect node respectively, so these two nodes meet the real-time requirement of highly dynamic MSL games.

5.3 The NuBot Control node

On top level of the controllers, the NuBot soccer robots typically adopt a three-layer hierarchical structure. To be specific, the NuBot control node basically contains strategy, path planning and trajectory tracking.

The design of the soccer robots is aiming to fulfill all the tasks completely autonomously and cooperatively. Therefore multi-robot cooperation plays a central role in MSL (Kasaei et al, 2011). To allocate the roles of the robots and initiate the cooperation, a group intelligence scheme is proposed to imitate the captain or the decision-maker in the competition (Wang et al, 2010).

\(^7\)http://wiki.ros.org/uvc\_camera.
In this direction, we employ a hybrid distributed role allocation method including role performance evaluation, assignment and dynamic application. The soccer robot can select the following candidate roles: attacker, defender and other roles based on the performance evaluation. In the end, we utilize dynamic application to avoid the repeated role assignments for the inconsistent information between different robots.

We employ non-hierarchical cooperation and hierarchical cooperation to deal with different situations as shown in Fig. 12. For the non-hierarchical cooperation in Fig. 12(a), each robot is equal. Besides, the non-hierarchical cooperation is performed by sharing the common information, which is individually maintained by each robot. Taking the defensive action as an example, each robot chooses its defensive action mainly based on its location and the information exchanged between others. Different from the non-hierarchical cooperation, the robots are not equal in hierarchical cooperation in Fig. 12(b). The robots realize the cooperation through direct communication. The cooperation is initiated by one robot called initiator. Particularly, the initiator is responsible for selecting and informing the other robots for tactic cooperation. The cooperation relationship vanishes naturally while the specific tactic is finished. For example, we realize the free kick cooperation through hierarchical cooperation.

While the roles are determined, each robot is motivated to perform the corresponding tasks individually and autonomously, such as moving, defending, passing, catching and dribbling. Path planning and obstacle avoidance is still quite a challenge under highly dynamic competition environments. To deal with it, an online path planning method based on the subtargets method (Bruijnen et al., 2007) and B-spline curve is proposed (Cheng et al., 2014). Benefiting from the proposed method, we can obtain a smooth path and realize real-time obstacle avoidance with a high speed. The method can be summarized as follows.
1. generating some via-points employing the subtargets method iteratively.
2. obtaining a smooth path by using B-spline curve method between via-points;
3. optimizing the planning path via some actual constraints such as the maximal size of an obstacle and the robot velocity and so on.

In fact, this method is simple yet effective. Besides, we also notice that, for the original subtargets method, the local minima problem cannot be avoided. As depicted in Fig. 13(a), while the destination is blocked by some obstacles, the robot oscillates back and forth and cannot find a path to the destination. Our method can identify this situation accurately, deal with it by exchanging the destination and the robot’s position, and obtain a smooth path to the destination, as shown Fig. 13(b).

Fig. 13 The paths generated by using the original subtargets method (a) and our proposed method (b). The red circles represent some obstacles between the robot and the target point. The blue circles represent some via-points in the planned dynamic path.

To track the planning path/trajectory with high speed and obtain a quick dynamic response with low tracking errors, Model Predictive Control (MPC) is utilized, since MPC can easily take into account the constraints and utilize the future information to optimize current output (Zeng et al, 2013). To begin with, we obtain linear full dynamic error model based on the kinematics model of the soccer robot. Then, MPC is employed to design the control law to satisfy both the kinematics constraints and dynamics constraints. Meantime, in order to reduce the computational time for the on-line application, Laguerre Networks is used to design the MPC controller (Wang, 2009). As illustrated in Fig. 14, our robots can track the path/trajectory with a quick dynamic response and low tracking errors by our proposed MPC control law, so the motion ability and the obstacle avoiding ability can be improved.

5.4 The NuBot HWControl node

On bottom level of the controllers, the NuBot HWControl node performs four main tasks: controlling the four motors of the base frame, obtaining odometry information, controlling the ball handling system and the shooting system. The ROS EtherCAT library for our robots is developed to exchange information between the
industrial PC and some actuators and sensors (e.g. AD module, I/O module, Elmo, motors, linear displacement sensors.). The speed control commands calculated in the NuBot Control node are sent to four Elmo motor controllers of the base frame at 33Hz for realizing robot motion control. Meanwhile, the motor encoder data are used to calculate odometry information, which are published to the OmniVision node. For the third task, high control accuracy and high-stability performance are achieved by feedback plus feedforward PD control for the active ball handing system. The relative distance between the robot and the ball measured with two linear displacement sensors is regarded as feedback signal, and the robot velocity is used as the feedforward signal. For the last task, the shooting system firstly needs to be calibrated off-line. During competitions, the node adjusts the hinge of the shooting rod to different heights according to the received commands: flat-shooting or lob-shooting from the NuBot Control node. Then it receives the shooting commands, selects the shooting strength according to the calibration results and kicks the ball out.

5.5 The WorldModel node

The real-time database tool (RTDB) developed by the CAMBADA team (Almeida et al, 2004) is employed to realize robot-to-robot communication. The information of the ball, the obstacles and the robot itself provided by the OmniVision node, the Kinect node and the FrontVision node is combined with the data communicated from teammates to acquire a unified world representation in the WorldModel node. The information from its own sensors and other robots is of great significance for single-robot motion and multi-robot cooperation. For example every robot fuses all obtained ball information, and only the robot with the shortest distance to ball should catch it and others should move to appropriate positions; each robot achieves accurate positions of the obstacles and obtains the positions of its teammates by communication, thus it can realize accurate teammate and opponent identification, which is important for our robots to perform man-to-man defence.
5.6 Improving real-time performance using RT-preempt patched kernel

Our software based on ROS can be developed more efficiently and reused in different MSL robots without changing codes or with a few changes. Though it is possible to integrate ROS with real-time code. But ROS is not a real-time framework. So we develop the RT-preempt patched kernel\(^8\) to replace the vanilla Linux kernel in Ubuntu, which provides real-time capabilities on the OS layer to real-time demanding ROS nodes. The RT-preempt patched kernel can help ROS realize process scheduling, and provide more computing resources to those ROS nodes with higher priority. For that reason, the kernel is used to improve the real-time performance for our robots. To validate the performance of the RT-preempt patched kernel, we run the Kinect node which involves massive calculation, and increase the priority of the Kinect node. The processing time for the Kinect node was evaluated in each frame when using the kernel or not. As shown in Fig. 15, the processing time with the RT-preempt patched kernel is more smooth, and all frames of image data can be processed within 40ms, hence demonstrating that the RT-preempt patched kernel can be used to optimize ROS for real-time applications. The spike of the blue line may be caused by suddenly increased computing resource or no object being recognized from the RGB-D data.

6 Conclusion

In summary, we presented the whole design of our soccer robot with the modular mechanical platform, industrial electrical system and ROS-based software

framework in this paper. The proposed designing methods support iterative and incremental development for soccer robots at relatively low costs. In addition, we employed the RT-preempt patched kernel to optimize ROS for improving the real-time performance. We expect this work to be of value in the robotics community. On one hand, the researchers can refer to our method to design their MSL soccer robots or general intelligent robots. On the other hand, the NuBot can also be developed further to become a candidate standard platform for RoboCup MSL. Our work can be beneficial to promote the research in artificial intelligence and robotics. Further, we will focus our attention to multi-robot cooperation based on our soccer robots. Still, some questions remain open regarding how to avoid potential conflicts due to miscommunication. And the future deeper research is necessary in order to develop our robot’s cooperation ability by involving more robots and more complex cooperative behaviors.

Acknowledgement

Our works are supported by graduate school of National University of Defense Technology and National Science Foundation of China (No.61403409).

References

The Design of an Intelligent Soccer-Playing Robot 21


Van De Molengraft M, Zweigle O (2011) Special issue on advances in intelligent robot design for the robocup middle size league. Mechatronics 21(2):365


