NuBot Mechanical and Electrical Description and Software Flow Chart 2018

Wei Dai, Weijia Yao, Sha Luo, Junchong Ma, Runze Wang, Shaozun Hong, Zhiqian Zhou, Xiao Li, Bingxin Han, Junhao Xiao, Huimin Lu, Zhiqiang Zheng

College of Mechatronics and Automation,
National University of Defense Technology, China, 410073
http://nubot.trustie.net
lhmnew@nudt.edu.cn, junhao.xiao@ieee.org

Abstract. The software structure, mechanical and electrical system of the robot from NuBot team is described in this material. The size of the current robot is 50cm × 50cm, and the weight is about 35 kg.

1 Mechanical

After RoboCup 2017 in Japan, we developed a new generation of robot platform as shown in Fig. 1. As the upgrade of the last generation, this generation of robot has a more reasonable layout and more stable performance. We re-designed the external frame of it to better adapt to the competition environment with huge impact (size: 50cm*50cm*80cm, weight: about 37kg). The rest of this part only details the active ball handling system and shooting system.

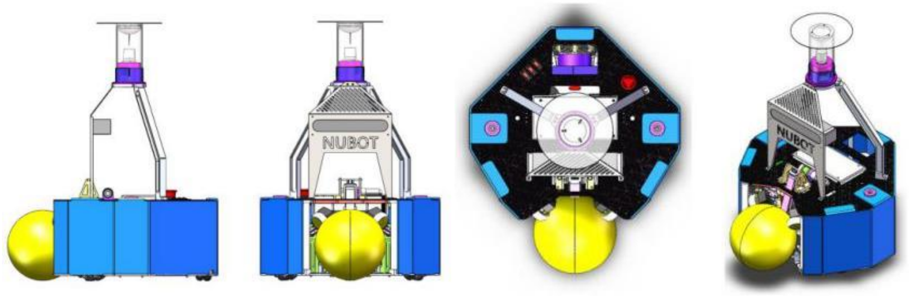

Fig. 1. The NuBot robots.

1.1 Base Frame

In our omnidirectional wheeled platform, we use custom-designed omnidirectional wheel, which is illustrated in Fig. 2 (left). Three such omnidirectional wheels are uniformly arranged on the base as shown in Fig. 2 (right).

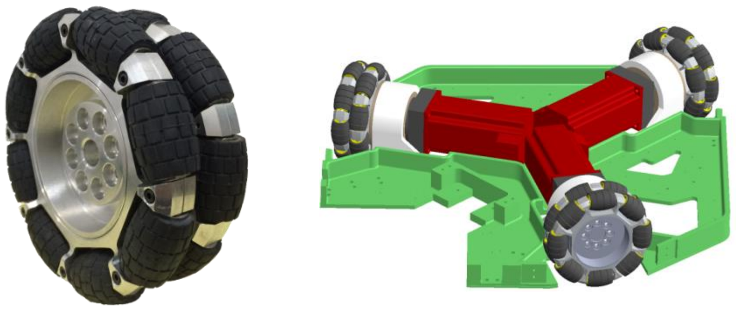

Fig. 2. The omnidirectional wheel and the base frame of the NuBot soccer robot.

1.2 The Active Ball Handling System

The active ball handling system, which is designed for dribbling the ball, is made up of the active ball handling mechanism and its close-loop control system. As illustrated in Fig. 3, there are two symmetrical assemblies, and each contains a wheel, a DC motor with a right angle reducer, a linear displacement transducer and a support mechanism. The wheels are driven by the DC motor and are always pressed by the ball, therefore they can generate various frictional force to the ball, making it rotate in desired directions and speeds together with the soccer robot. During dribbling, the robot will constantly adjust the speed 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, and then stretch the transducer; otherwise, the support mechanism will fall and compress the transducer. The information obtained from two transducers can be used to calculate the actual ball distance based on a given detailed geometry model and careful calibrations. This system effectively solves the ball handling control problem.

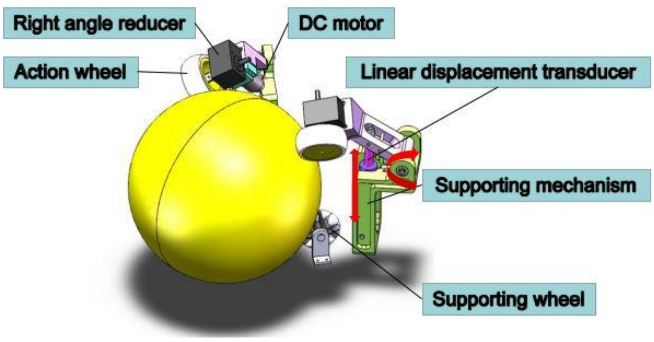

Fig. 3. Our active ball handling system.

1.3 The Electromagnet Shooting System

The shooting system is based on an electromagnet with a high impulsive force. As depicted in Fig. 4, it consists of a solenoid, an electromagnet core, a shooting rod, a capacitor, and a DC motor. The shooting rod can be adjusted in height to allow for different shooting modes, namely lob shot and ground pass. Two modes are realized using the DC motor to pull the hinge of the shooting rod to different positions. Firstly, 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 selected mode. Then the control circuit board will switch on the solenoid by discharging the capacitor, thus producing 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 string, and the capacitor will be recharged again and wait for the next shooting action. In addition, we incline the electromagnet so that the electromagnet core can go back to the initial position more easily. Therefore, this system is simple yet capable of various shooting angles.

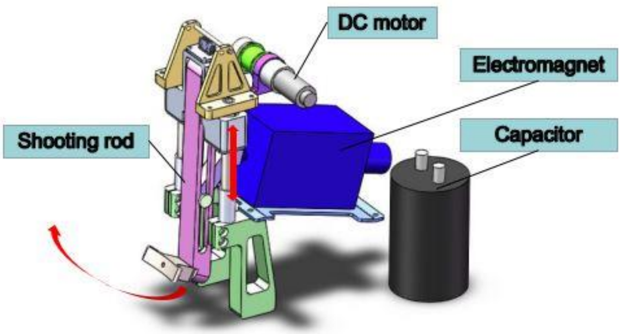

Fig. 4. The electromagnet shooting system of the NuBot soccer robot.

2 Industrial Electrical System

In recent years, the risk of the fierce collision between robots increased in the highly dynamic MSL competition. To improve the real-time performance and robustness of our robot control system, we design our current electrical system using PC-based control technology as shown in Fig. 5. 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 the dedicated hardware. 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 limitations on the number of I/O modules, sensor modules and actuator modules are 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.

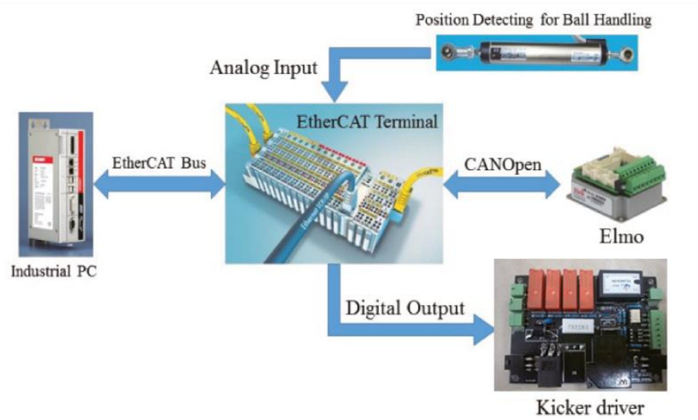

Fig. 5. The NuBot electrical system.

3 The Software Based On ROS

Robot Operating System (ROS) provides a set of software libraries and tools for building robot applications across multiple computing platforms, which has many advantages: ease of use, high-efficiency, cross-platform, supporting multiple programming languages and distributed computing, code reusability. In addition, it is open source under BSD license.

We have built our software based on ROS for our new robots. The operating system is Ubuntu 14.04, and the version of ROS is indigo. The software framework, as shown in Fig. 6, is divided into 4 main parts: the Prosilica Camera node, the OmniVision node and the Kinect node; the NuBot Control node; the NuBot HWControl node; the RTDB and the WorldModel node. As for the goalie, the software framework is the same except that there are two Kinect nodes. These nodes will be described in the following sub-sections.

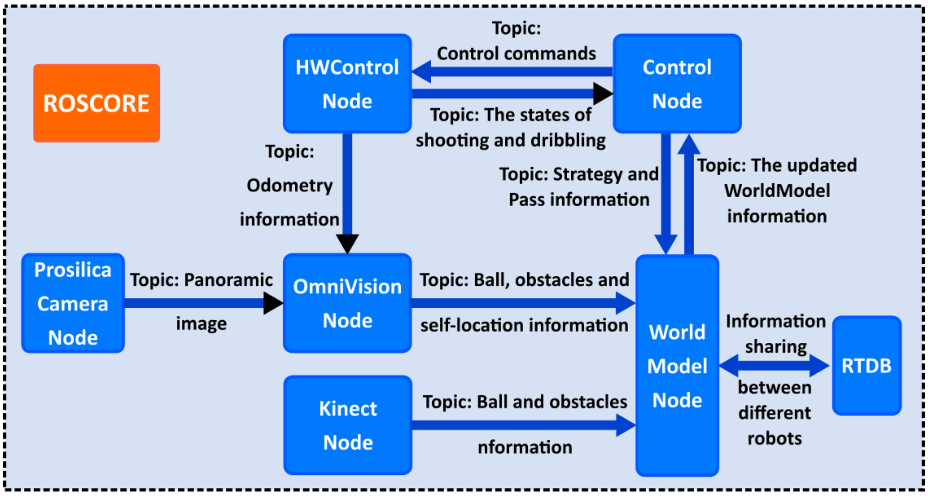

Fig. 6. The software framework based on ROS.