ZJUDancer Team Description Paper
Humanoid Kid-Size League of Robocup 2018

Yusu Pan, Bo Peng, Chaofeng Jiang, Chenghui Li, Chunlin Zhou, and Rong Xiong

State Key Lab. of Industrial Control Technology,
Zhejiang University, Hangzhou, China

Abstract. This paper describes the robot system designed by ZJU-Dancer, a RoboCup Humanoid League team from Zhejiang University, China, as required by the qualification procedure for the competition to be held in Montréal, Canada during June 17-22, 2018. Full details of our robots including mechanical andelectrical design, sensors and software architecture are described. This year we work hard in appling force sensers for robot foot, and improving the performence of vision system. With reinforced robots, we hope to get a much better result in 2018.

Keywords: Humanoid Robots, Force Sensor, Deep Learning

1 Introduction

In this paper, we describe our robot system for RoboCup Humanoid League, designed by ZJUDancer.[3] This year we made a large progress in image processing and gait control, which contributes to more intelligent robots.

2 Overview of system

The robots developed by ZJUDancer for RoboCup 2018 are fully autonomous humanoid robots, with the capability to play varied parts as a team in the competition. Fig. 1 shows our robot.

Table. 1 shows the general specifications of our robots. Each robot is fixed to the size and weight limitations of the competition and connected by wireless networks. Referees directions can be sent to the robot through the network. More details will be introduced in following sections.

(a) Robot Kicking the Ball

(b) Mechanical Sketch

Fig. 1: Robot of ZJUDancer

Table 1: General Specifications

Item	**Description**
Team Name	ZJUDancer
Number of DOF	20
Height	620mm/700mm
Width	35cm
Weight	4kg
Computing Unit	NVIDIA Jetson TX1/TX2

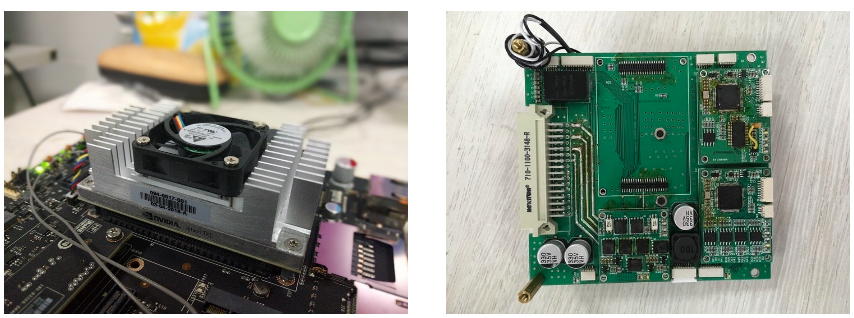
3 Hardware

3.1 Electrical Specifications

This year, our electrical controllers, including motor controller and camera controller, are integrated into the main controller, whose specifications are shown in Table. 2. The camera controller part of main controller works on object detection, self-localization, strategies and multi-robot communications. Due to Jetson TX1, the high-performance supercomputer module, we could slim the size of our robots and calculating faster meanwhile. The movement and balance maintaining are implemented by the motor controller part of main controller which executes the movement of all directions. Total electrical architecture could be seen in Fig 2.

Table 2: Electrical Specifications

Main Controller	
CPU	NVIDIA Jetson TX1/TX2
Flash	16GB/32GB
RAM	4GB/8GB
OS	Ubuntu 16.04.3 LTS



(a) NVIDIA Jetson TX1 (b) Control Board

Fig. 2: NVIDIA Jetson TX1 and Main Controller Board

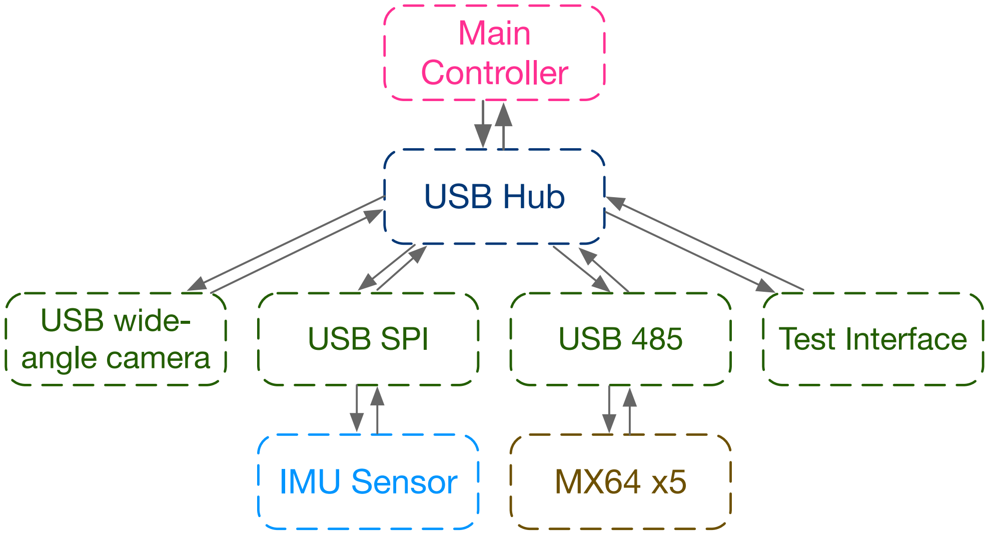


Fig. 3: Hardware Architecture

3.2 Mechanical Specifications

Our robot has two legs, two arms, one trunk and one head. We have two specifications of our robot. One of them is of nearly 620 mm height and the other one is nearly 700 mm which has longer legs and neck. The actuators we use for the small one are Dynamixel MX-64 for legs and MX-28 for the upper part of the body while the big one uses MX-106 for knees, ankles and the roll axis of hips which provide stronger moment of force to drive the bigger robot. Each robot is driven by 20 servo motors: 6 per leg, 3 in each arm and 2 in the head. The six-leg-servos allow flexible leg movements. Three orthogonal servos constitute the 3-DOF hip-joint. Two orthogonal servos form the 2-DOF ankle joint. One servo drives the knee joint. The motor distribution is different but the DOF is the same.

Since we use Nvidia Jetson TX1 as computing unit, which is much bigger and heavier than the one we used before, we try to use carbon fiber instead of aluminum alloy to build up the trunk. It seems that the new trunk is a little lighter than before and the behavior of pick up itself and walking is better than before.

Table 3: Motor Type and Distributions of DOF (Totally 20 DOF)

Part	Rotation Axis	Acutator
Neck	Yaw, Pitch	MX-28, MX-28
Shoulder	Roll, Pitch	MX-64, MX-64
Arm	Pitch	MX-64
Hip	Roll, Yaw	MX-106, MX-106
Knee	Pitch, Pitch	MX-106, MX-106

3.3 Sensors specification

- **Servo**: DYNAMIXEL MX-64, MX-28, with joint angle feedback, it helps close-loop control.
- **IMU**: Analog device ADIS16355. Featured with tri-axis gryoscope, and tri-axis accelerometer. It helps to keep balance of our robot.
- **Image sensor**: OmniVision OV2710 with 150 degree FOV, which provides wider view angle, and helps improve the efficiency of perception.

4 Software

Our software architecture has been totally refactored this year. We adopt ROS framework as network middleware, and benefit greatly from its publish-subscribe communication pattern. Then there're several seperate modules for different tasks. The overview is shown in Fig .4.

During the competition, our main controller runs 6 modules simultaneously. At first, the vision module processes incoming camera frames and recognize landmarks, robots and balls on the field. Localization module receives those information and then optimize self location estimation. Topics from game controller, teammates and localization module finally come together into behavior module, which decides where and what to do next. Action commands from behavior module are passed to motion module, and go to actuators afterwards. Besides, motion module alse collects servo feedback and IMU pose and publish them.

Detailed improvements we've made this year are in the following subsections.

4.1 Cognition

Last year, we adopted a deep learning approach for recognition of ball, robots and goals, Due to the supe-fast YOLO and powerful NVIDIA Jetson TX1, it reached nearly 20fps, which is close to real-time detection on such a low-power comsumption platform. During RoboCup 2017, this module worked excellently in both accuracy and stability.

To acceletate this task further, a lighte-weight base network called MobileNet[2] is chosen to replace the original GoogleNet-like one. Moreover, we completely re-write our deep learning framework, which is much more efficient. Object tracking technology like KCF[1] is also merged into our module.

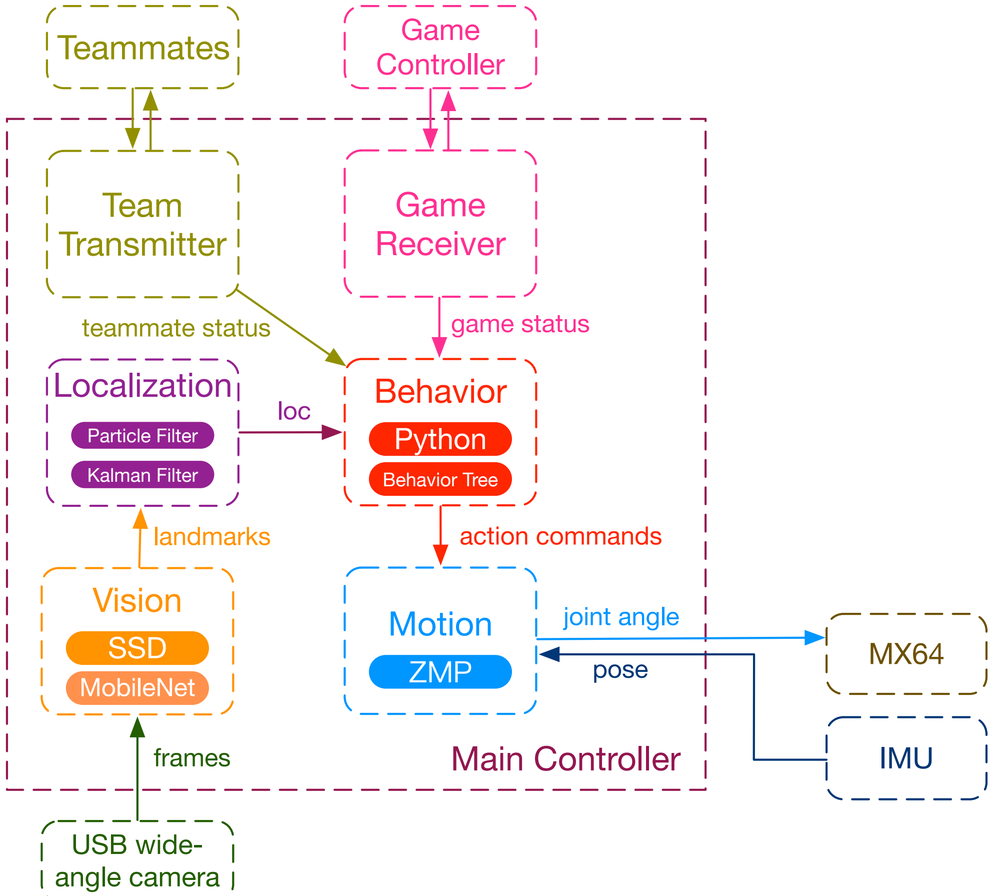
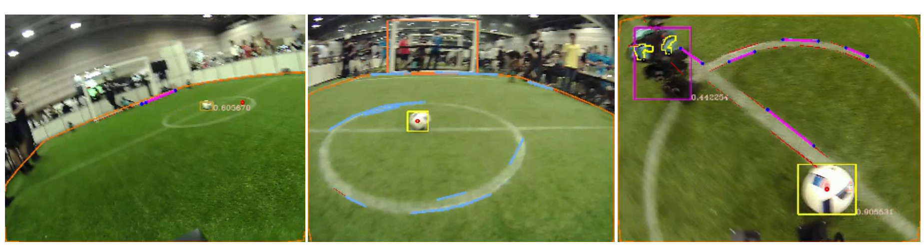


Fig. 4: Software Architecture



(a) Ball Detection (b) Goal Detection (c) Robot Detection

Fig. 5: Obejct detection

4.2 Motion

This year we refactored the code that controls the gait of our robots. Our old code was so bloated and cumbersome that it was too hard for our new members to understand and modify. To optimize the trajectory of the end point movement of the legs, we substituted the cubic spline plan with polynomial programming. And we added a new piece of code so that the robots can kick the ball in all directions.

4.3 Navigation

This year we return to particle filter and make lots of improvements on it.

4.4 Behavior

Currently, our behaviour module is based on a mixture of behaviour tree and finite state machine, implemented in Python. It reads information from the localization module and gamecontroller module, and plans action command in each main loop. This year we're trying to apply reinforcement learning on our robots. It now works pretty well on simulation environment, and real-world test will be done later.

5 Conclusion

This paper describes the main structure our robots. We ZJUDancer have made a huge progress in both hardware and software. Firstly, we've designed a brand new main controller which intergrates the function of motor controller. Secondly, we utilize deep learning in object detection. Thirdly, we're trying new methods for gait planing. We would like to share our experience and have a good match with all the teams in Montréal 2018.

References

1. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37(3), 583–596 (2015), `https://doi.org/10.1109/TPAMI.2014.2345390`
2. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861 (2017)
3. Mei, W., Pan, Y., Liu, Y., Jiang, C., Peng, B., Yu, D., Zhou, C., Xiong, R.: Zjudancer team description paper (2017)