# Global System Description

## 1. Vision and Localization

The previous software structure based on Linux and LUA/C++ platform has been replaced by ROS framework, which greatly increases the efficiency of multitasking coordination and simplifies the process of application of new algorithm.

There are mainly four modules of our algorithm which are Vision, Modeling, Behavior and Motion. Vision Module is used to recognize objects in the football court and convert them into information in geometric forms. The geometric forms of these objects are then used in Modeling Module to calculate and predict the location of them by Particle Filter Localization. Behavior and Motion Modules help our robots to decide the next action to take. However, although we still reserve these basic modules, there are improvements in the following algorithm. First, we are going to use the deep learning methods to increase the accuracy of the recognition of the objects. Besides, we apply particle filtering algorithm to the process of robots self-locating. Moreover, the problem of coordination between several FSM has been solved by designing more complex behavior tree structure.

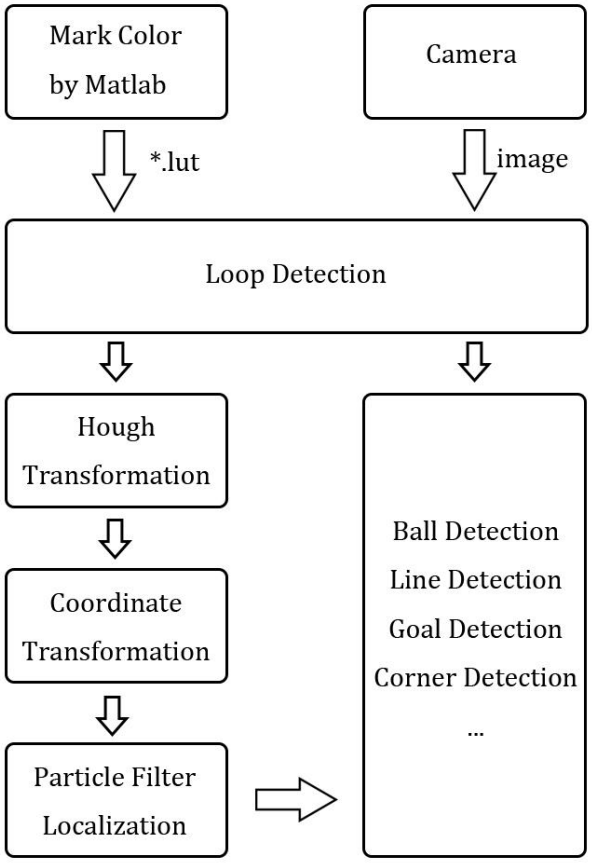


Figure.4 Localization and Detection Structure

## 2. Behavior

Inspired by the hierarchical state machine (HSM) programmed in XABSL[1], we introduce HSM into our algorithms utilized to generate behavior. Our HSM is implemented with Lua embedded in C++. The framework of the algorithm is composed of several super-states and each super-state is constructed by a group of subordinate states. In our case, those super-states include playing game, listening to controller, standing up and playing defense. Correspondingly, those subordinate states, which are all basic actions able to be executed directly by the robot, involve finding ball, approaching ball, shoot at goal and so on. A typical graph of our algorithm is shown in Fig.4.

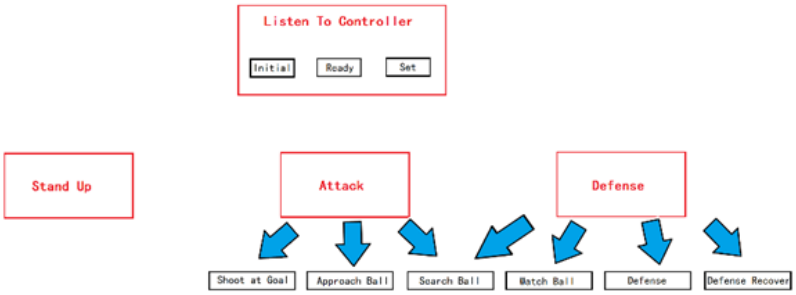


Fig.5 A typical graph of our algorithm

## 3.Walk

As a main difference from last version of robot, the new version integrates NVIDIA Jetson TX2 into the robot. Thus, the computation of generating gaits such as omnidirectional walking, getting up after falling down, kicking the ball and so on, is transferred from motion controller (LPC1768) to

decision controller (Jetson TX2). The prime advance is that the robot can produce gaits exploiting the abundant computing power of Jetson TX2.

We store the parameters of basic gaits that have been adjusted by our team into the motion controller. After processing the data input from sensors, the decision controller will send a signal which includes the number of the gait to be executed and some changes to the default parameters stored to the controller. Then the controller will send signals to the actuators to generate the certain gait. By this means, the robot can adapt to the changes of circumstances in the court by real-time corrected parameters instead relying on fixed gaits.

## 4. Application of Deep Learning in Ball searching

The task of this part is to quickly recognize the low-resolution image and return a vector representing the position and distance of the ball. The motion mechanism looks for the ball in the direction indicated by the vector.

Our artificial neural network system adopts nonlinear decision making, which is constructed by two layers of sigmoid elements, that is, one layer of hidden layer and one layer of output layer. The size of the hidden layer is determined according to the actual calculation ability and the technical details of the processor. The size of the output layer is no less than 3, and the relative position of the ball is determined according to the output vector.

In the aspect of input coding, we adopt gray-scale maps of 32p resolution which has been preprocessed. In order to increase the operation speed, the gray scale of each pixel is obtained by random sampling of the corresponding region of the source image. This strategy makes it easy to convert input into vector processing.

In the aspect of output coding, we combine direction and distance in the vector to indicate the position of the ball. In order to accelerate the training process, the minimum value of the standard output of the training sample is slightly greater than 0, and the maximum value is slightly less than 1.

The neural network structure adopts the standard three-layer acyclic sigmoid network. In the process of training, the neural network modifies the weight vector while cross-verifying the size of the hidden layer. That is to say, lack of hidden layers will lead to poor fitting effect and waste of training samples. In contrast, too many hidden layers will lead to over-fitting and high training cost. We will add impulse term to enhance accuracy and speed up training in the process of applying backpropagation algorithm.

## References


1. S. Behnke. Online Trajectory Generation for Omnidirectional Biped Walking. Proceedings of the 2006 IEEE International Conference on Robotics and Automation. Florida, 2006, pp. 1597-1603.
2. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H Hirukawa. Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point. IEEE International Conference on Robotics and Automation. pp. 1620-1626, 2003.
3. D. Gouaillier, C. Collete, C. Kilner. Omni-directional Closed-loop Walk for NAO, 2010

IEEE-RAS International Conference on Humanoid Robots. Nashville, TN, USA, 2010, pp. 448-454.

4. J. Alcaraz-Jimenez, D. Herrero-Perez, H. Martinez-Barbera. Motion Planning for Omnidirectional Dynamic Gait in Humanoid Soccer Robots. Journal of Physical Agents, 5(1): 25-34. 2011.

5. J. Strom, G. Slavov, E. Shown. Omnidirectional Walking using ZMP and Preview Control for the NAO Humanoid Robot. RoboCup 2009: Robot Soccer World Cup XIII. Springer, 2009, pp. 378-389.

6. C. Graf, A. Hartl, T. Rofer, T. Laue. A Robust Closed-Loop Gait for the Standard Platform League Humanoid. Proceedings of the 4th Workshop on Humanoid Soccer Robots in conjunction with the 2009 IEEE-RAS International Conference on Humanoid Robots. Paris, France, 2009, pp. 30-37.

7. O. Chutatape, Linfeng Guo. A modified Hough transform for line detection and its performance. Pattern Recognition, 32(2): 181-192, 1999.

8. S. Thrun, W. Burgard, D. Fox. Probabilistic Robotics. MIT press, 2005

9. T. Rofer, T. Laue, M. Weber, et al. German Team RoboCup 2005. [Online]. Available: http://www.germanteam.org/GT2005.pdf

10. M. Lotzsch, M. Risler, and M. Jungel. XABSL - a pragmatic approach to behavior engineering. Proceedings of IEEE/RSJ Intl. Conf. of Intelligent Robots and Systems.

11. Wang Xinhui, Yu Chang, Fu Jiawei. Team TH-MOS. [Online]. Available: https://submission.robocuphumanoid.org/uploads//TH_MOS-tdp-5c4d9e12552bf.pdf