Plasma-Z
Team Description for RoboCup 2009

Chitchanok Chuengsatiansup[1], Thiraphat Charoensriphngsa[1],
Kanit Wongsuphasawat[1], Komsit Rattana[1], Pawawat Duongsodsri[1],
Aphilux Buathong[2], Kittipat Wejwittayaklung[3],
Manop Wongsaisuwan[4], Wittaya Wannasuphoprasit[3]

Department of Computer Engineering[1]
Department of Industrial Engineering[2]
Department of Mechanical Engineering[3]
Department of Electrical Engineering[4]
Chulalongkorn University
Phayathai Road, Bangkok 10330, THAILAND
http://www.eic.eng.chula.ac.th
elizaemol@gmail.com, tjump_c@hotmail.com

Abstract. This paper describes the currently developed Plasma-Z robot soccer team. We present an overview for the entire system which consists of 4 major systems: mechanical, electrical, vision, and AI.

1 Introduction

Plasma-Z is a robot soccer team from Engineering Innovator Club, Faculty of Engineering, Chulalongkorn University, Thailand. Plasma-Z has joined the RoboCup Small-Sized League since 2003. So far, Plasma-Z has won the RoboCup Thailand Championship in 2003, 2004, 2005, 2006, 2007, and 2008. Moreover, Plasma-Z was the champion in World RoboCup Small-Sized Robot League at Suzhou, China, 2008. We have been experimenting and developing many parts of our robot system for even better performance to join the World RoboCup 2009.

2 Mechanical System

Plasma-Z's robot has been improved continually since 2002. We have gained a lot of experience during past 6 years in SSL robot competition and acquired notices of robotic mechanical design from our seniors. The mechanical design for 2009 is aimed to continue all preceding robots' strengths while decreasing their weaknesses.

2.1 Overall

This year, the design concepts are mainly focused on robust mechanism, agility, easy controlling, and all desirable based abilities. We utilize the *CATIA* V5R16 CAD (Computer Aid Design) to simulate all parts of the robot before real manufacturing processes. The latest robot's cover is made of a 2 mm-thick fiberglass, in order to protect inner mechanical and electrical components. A material selection is revised, for example, preceding rubber bands, using for pulling a kicker back, are replaced by springs since we have learned how unreliable that material is.

The robot has a diameter of 176 mm, height of 148 mm, and weight of about 4.2 kg. Owing to the weight, our robots can easily steal the ball possession from opponents, and interrupt their play and break off their game effectively. Moreover, it has many features such as 4 omni-directional wheels, chip-kick system, flat-kick system, dribbling system. Both top and bottom chassis have thickness of 3 mm and coated by hard anodize process in order to prevent an electrical conduction (in other words, short circuit) and also harden them as well.

2.2 Driving System

Plasma-Z uses an omni-directional drive. With this driving system, the robot obtains 3 degree of freedoms in movement: translating in X and Y direction and rotation in Z direction. Because of the increasing in size of the field in 2008, the high final velocity was considered to be important. Therefore, we designed a big wheel instead of reducing the gearbox ratio substantially since high torque of the wheel is also important. As the result, the diameter of the wheel is about 61 mm. An angle of the front wheels is 124º , and the back couple is 90º. With practical motor speed of about 4200 rpm, the theoretical robot's forward average speed is 2.91 m/sec, and average acceleration is 3.31 m/sec^2. An aluminium motor mount is also coated by hard anodize process. A front plate of each omni-wheel is designed for protecting its polyoxymethylene (POM) rollers while a threaded back plate is made of stainless steel.

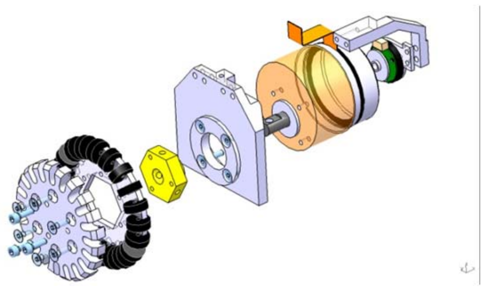

Fig.1 Exploded view of the driving system

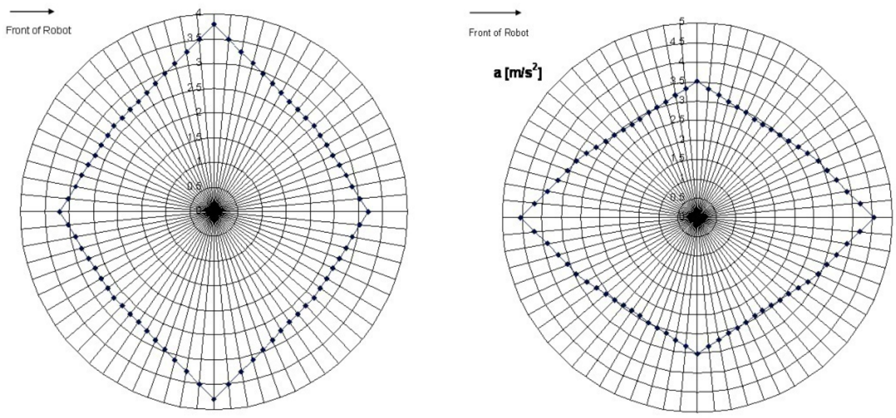

Fig.2 Velocity profile of the robot (left) and acceleration profile of the robot (right)

2.3 Shooting System

First of all, chip-kick system is distinctively changed since an extension of a new spur gearhead. The best solution is reshaping the solenoid from circular aspect to rectangular one, and placed into an available space. Rods of both flat and chip kick are made of S45C iron which has high magnetic permeability and eventually leads to powerful shooting result. This current version, the robot is able to kick straightly up to 10.7 m/sec, and chip the ball more than 3.7 m in length. During, the game, the robot can choose the speed of shooting the ball via adjusting time duration of turning on the IGBT gate.

The flat-kick is attached at the bottom chassis which acts like a huge heatsink for heat extracting, and supported by U-shape 2 mm stainless steel plate, while the chip-kick is held by the top chassis. These chassis plates play a role as a backbone of the robot because they fix 4 motors mounted all together. The top chassis is also a base plate for placing an electrical PCB and robot's cover.

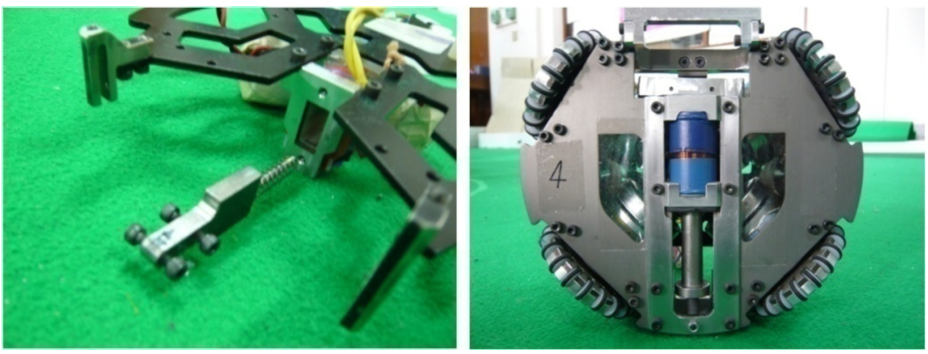

Fig.3 Chip-kick solenoid rod (left) and bottom view of robot (right)

2.4 Dribbling System

The dribbler spins the ball backwards to the center of the robot, so the soccer machine can possess the ball while turning around to the direction of an opponent's goal. One of the main design concepts is the groove on the dribbler's axle in order to extend the area of the friction between the ball and the dribble shaft and also to make the ball roll into the grooved shape which makes the ball always roll in the center of the robot. This will help to increase the efficiency of shooting system as it shoots the ball right at the center of the golf ball and make the direction to be more accurate. A material used for dribbling is an ordinary silicone tube and a foam tube at each farther side. Both of them can be found at a stationery store. There is no any suspensions except an up-down linear sliding, but this system is very effective since it presses the ball by its weight and needs only a little space for placing at the front of the robot.

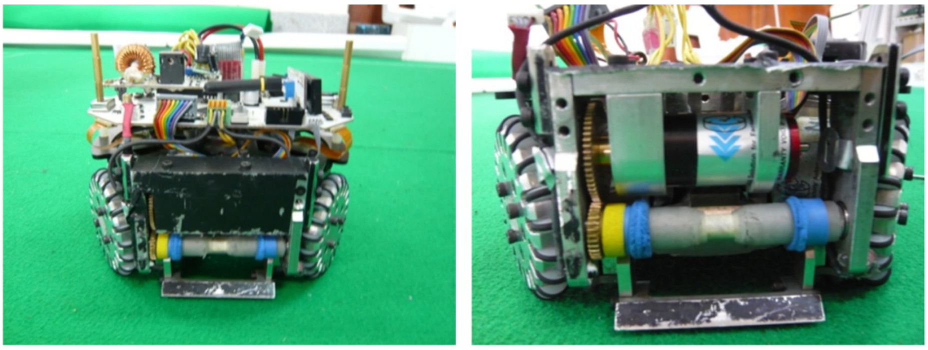

Fig.4 Front view of the robot (left) and inner mechanism of dribbler (right)

3 Electrical System

The major changes during past 3 years are the main processor part, the motor driving part, and the wireless communication module. In this current version, the main processor in integrated in the FPGA contrasted to the former version which was separated. This design not only reduces the processing cycle time, but also increases more space on the PCB. The motor driver has changed from H-bridge circuit to 3-phase inverter circuit that drives 3-phase Brushless DC (BLDC) motors. This makes the movement of the robot faster, and the speed can be controlled more accurately. The wireless communication was not changed much in the circuit layout, but the change was the selection of RF module.

3.1 FPGA design

FPGA system is composed of 8 modules. Each module has its own duty separated from others. There is one main processor module. A significant input in robot soccer is RF data which is an AI command for controlling the robot. The outputs are all the actions that the robot performs which is the movement, shooting the ball, dribbling the ball, LED display and buzzer, and RF data out.

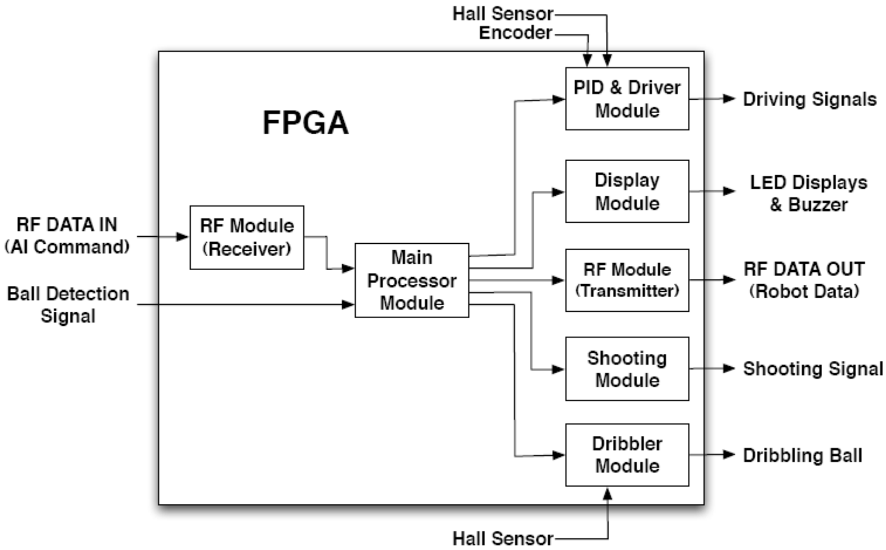

Fig.5 FPGA design

3.2 Motor Drive System

This system is composed of 5 BLDC motors. Four of them are used as the robot omni-directional wheels and one for dribbling the ball. The input signals from the motor are two of digital quadrature encoder feedback signals and three of built-in hall sensors. The signals from the miniature encoder and hall sensors are separated from FPGA by 74LVC245A IC Buffer. The output signals from FPGA to the driving circuit are 6 MOSFET gate drive signals to switch the current flowing through each stator coil of the BLDC motor. The signal path between FPGA and driving circuit is separated by 74LS07 IC Buffer.

To control the movement of the robot, the PID control is applied to the driver module implemented in the FPGA to control four motors – used as omni-directional wheels. This module combines with encoder submodule, PID calculation submodule, and motor driving submodule. Firstly, the robot receives velocity command which consists of velocities in XY-coordination and one angular velocity for each robot from the AI system. Next, the main processor module translates velocities and rotation command into velocity of each wheel. Then, the main processor module sends each wheel velocity command to this module for controlling it. The initial velocities measuring from encoders are calculated in PID module to the final velocity for each BLDC motor. That will make the motor rotate with an updated velocity as desired. Then, the output signals from PID send to motor driving module for choosing the gate to turn on in a driving circuit.

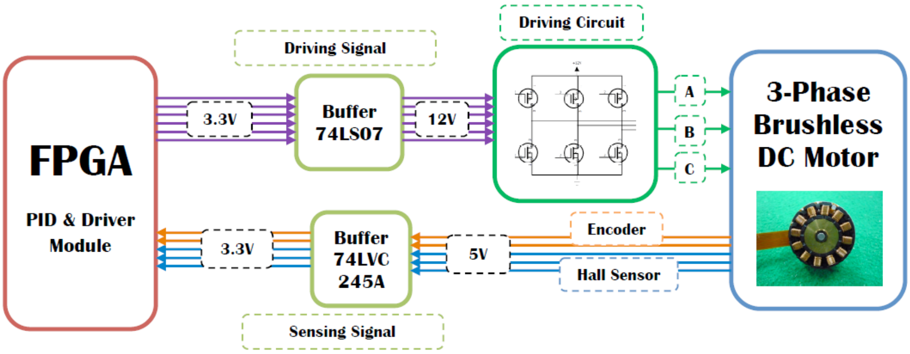

Fig.6 Motor drive system

3.3 Shooting System

This system contains controller system, charging system, capacitors and solenoids, It is separated from the main board so that it can be operated on its own. Four input signals are generated to control this system – charge enabling, flat shooting, chip shooting, and voltage reference. Ground of the main board and shooting board is separated by the optocouples.

As for charging system, pulse width generated by IC 555 is used to switch the MOSFET for boosting circuit. Then, two capacitors are charged to 250 Volts. Capacitor voltage is controlled by a comparator logic using an operational amplifier.

Shooting module implemented in FPGA is used for sending pulse, charging, and opening IGBT gate which enables current flow from capacitor through solenoid to flat shooting and chip shooting. AI can choose 2 ways of shooting the ball which are normal shooting and forced shooting. With normal shooting, shooting occurs only when the ball has detected by the ball detection part, but forced shooting can be used in any situations.

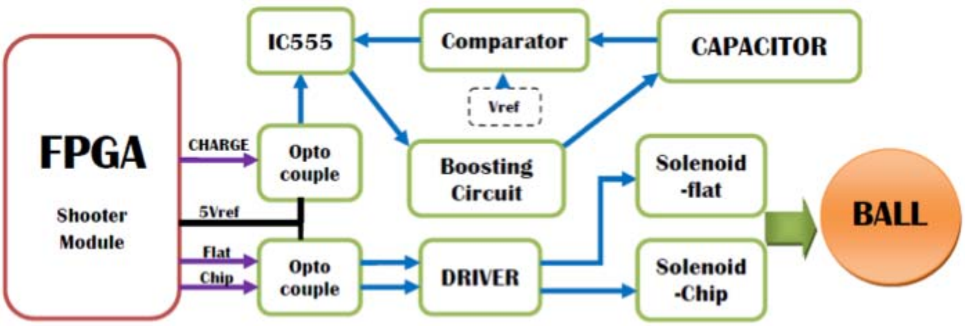

Fig.7 Shooting system

3.4 Dribbling System

The robot uses a BLDC motor to dribble the ball. Dribble module in FPGA is assigned to control this system by generating the PWM signal to drive the speed and using the hall sensor to measure the position of rotor. PID is not necessary because the aim is just holding the ball with the robot, not controlling the speed of rotating. Therefore, a fixed speed for rotation is sufficient to hold the ball with the soccer robot.

3.5 Communication System

LINX, RXM-900-HP3-PPS and TXM-900-HP3-PPS are the module which used for receiving and transmitting data in this system. AI system communicates with RF module in each robot by using this system to send out and obtain data in serial form. RF module implemented in FPGA can be divided into two parts. The first part is RF receiver which is used to receive RF data from AI computer. There is a CRC checker submodule used to detect accidental alteration of receiving data. The second part is RF transmitter which is used for sending out the robot's data to AI system. This also works with CRC creator for creating CRC used as a checksum in AI system when receiving data from each robot.

4 Vision System

The overview of this system is that we use 2 cameras for capturing the soccer field, then sending the images to the vision computer for processing. After that, we can detect the position of our robots, the opponent robots, and the ball. Finally, we send this data to AI system over the network.

4.1 Camera Calibration

Before the software can process the images received from the cameras, it needs to know some parameters of the camera. The parameters are grouped into intrinsic and extrinsic parameters. First, intrinsic parameters are the data depending on the camera and lens. For example, some of the intrinsic parameters are the focal length and distortion of the lens. Second, extrinsic parameters are based on the position of the camera consists of the translation metric (X, Y, Z axis) and the rotation metric (3 angles around axis). We perform this function only once after finish setting up the cameras. Then we will use the returned parameters until we change their position.

4.2 Capturing the Image

After having been captured by the cameras, the image is sent to the computer as an array of pixels. Each raw image the software received is in Bayer pattern because the cameras we use has only one CCD. So, it has to be transformed into RGB color model before being processed. Even though the camera already has a built-in function which can transform the image into RGB model, it will make the image sent to the computer much larger than Bayer pattern (about 3 times). When the image arrives the computer, it comes along with the distortion of the lens. Thus, the software has to resolve this problem by using the intrinsic parameters from the previous function before detecting the position of the robots and the ball.

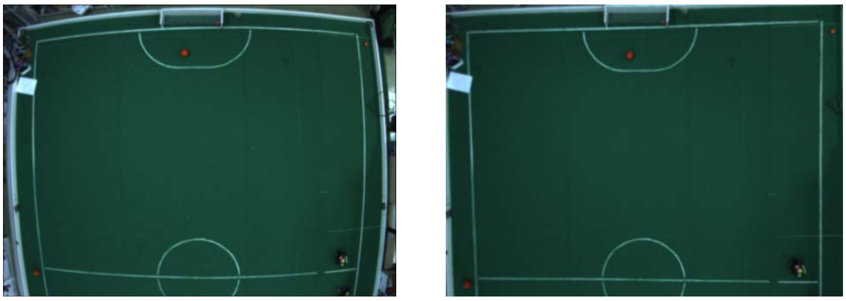

Fig.8 Distorted image (left) and undistorted image (right)

4.3 Blob Extraction

After the software has got the image in RGB model, it extracts interesting blobs. However, RGB color model is still not appropriate for specifying the color because it is too sensitive to brightness and shading. For instance, the variation of the color 'blue' we refer to is including light blue and dark blue. Only 'V' value that is changed. We can build a function to convert the color model for every frame, but it seems to waste too much time. So, we choose to build a look-up table for any colors in RGB model. The software will search in the look-up table using RGB value, and the table will return its HSV value. This method can save time instead of converting every frame. The last thing we need to do before extracting blobs is specifying criteria for the interesting blobs. The criterion contains 3 parts. First, the color range, we are interesting to the team color (blue and yellow) and the color used for identifying robots. Second, the blob size, it is the range of pixels indicating the blob we interested. If the software found that the blob size is out of range, it can imply that it is a noise and that thing can be discarded. Last, eigenvector is used for determining the expansion of the blob. As the software is only interested in the circle and long rectangle blob, a blob that has large expansion of eigenvector is a long rectangle, otherwise is a circle.

4.4 Robot's Head Recognition

Normally, the software recognizes only our robot's head. It uses each blob for identifying the number of our robots. There are shown in binary format. One blob represents one bit. We use a green blob for representing bit 0, and pink for bit 1.

According to **Fig.9** (right), the number of this robot is 001_2 in binary format of 1_{10} in decimal. For opponent robots, the software will only find a circle blob at the center of its head and imply that doesn't get noise which is a circle and has a color same as the opponent's.

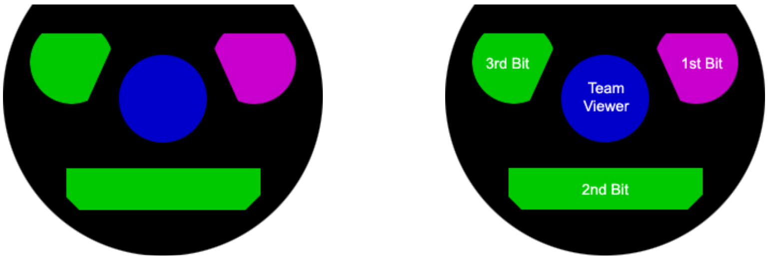

Fig.9 Robot's head pattern (left) and illustration of colors representing robot number (right)

4.5 Identifying the Robot's and Ball's Position

Extrinsic parameters we have got from the calibration take an important role here. Although the software can get the robot's and ball's position from the image, it is thought that their height is zero which makes its position inaccurate (as shown in **Fig.10**). Thus, the software needs to know their height to calculate the real-world position.

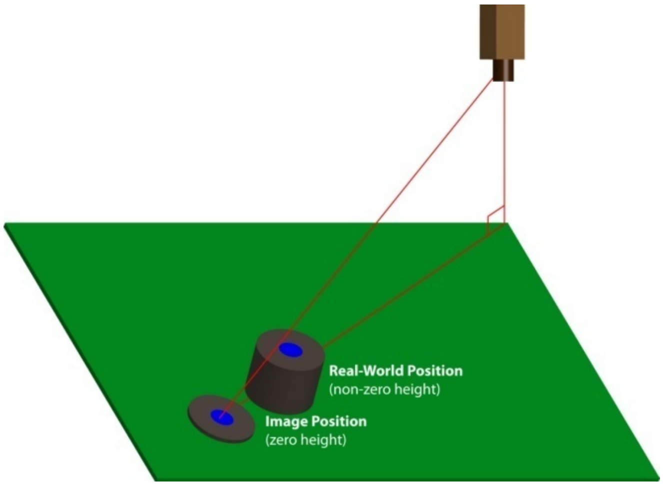

Fig.10 Comparison between zero and non-zero robot's height

4.6 Sending Robot's and Ball's Position to AI System

The software will open the socket for sending the position via the local area network (LAN). The software sends the robots' and ball's position to the AI computer every frame after finished processing the captured image. That means the sending rate is same as the frame rate of our cameras.

5 AI System

AI, whose duty is to receive data from vision session for processing before transferring to command all robots, can divide into two main parts: vision filtering and strategy.

5.1 Vision Filtering

Since the raw data from two cameras which contains positions and head directions of our robot, positions of opponent robot, and position of the ball is delayed, the vision filtering session is designed to collect and analyze the data to determine the actual position. For this purpose, Kalman's predictor is used. Furthermore, the past data is used to determine other useful information such as the moving directions of our robot and the ball.

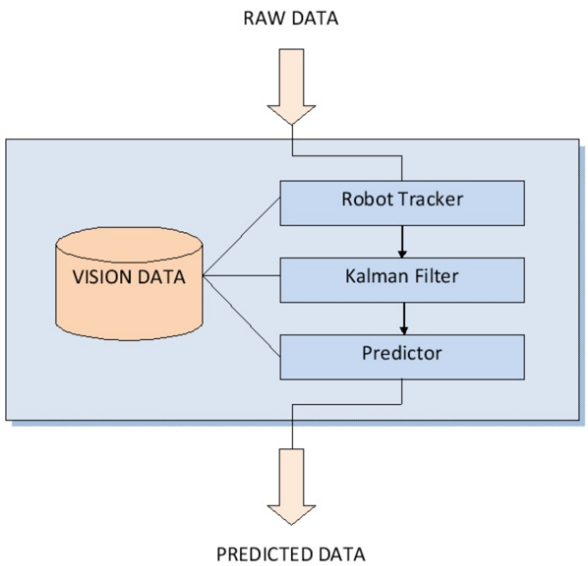

Fig.11 Vision filtering

5.2 Strategy

Strategy is considered to be the most important part of AI. It is the brain for planning strategy and coordinating among robots in both attack mode and defense mode. A layer architecture is used in this part.

Manager. Manager performs like a human soccer team manager. By receiving data such as game score, ball possession, opponent pattern, and referee box signal, manager applies the most suitable play for that moment.

Play. Play provides a real strategy of AI which consists of many game plans. Play will determine the pattern of the game and notify to all robots by assigning robots into 'Group.' Play also selects zone for robots and may issue some commands directly to robot role such as force role to pass the ball to another robot or shooting the ball.

Group. Group is a group of collaborative robots to perform a mission such as attacking group, defensive group.

Role. Role is performed as robot behavior which is assigned by 'Play' to control robot to perform a specific action such as manipulate ball to zone, run to zone, scramble ball from opponent, get ball. The 'Role' mechanism is first assign a particular skill to the robot, then generate the best point for robot action and set another parameter to 'Skill.'

Skill. Skill is a set of basic knowledge for every robot, such as how to move to a point, how to get the ball and shoot. Skill module generates path (a set of points), dribbling and kicking command that will be proceeded by varying of trajectory module which selected by 'Skill' module. Each skill has different main idea of generating path for robot. For example, 'get ball skill' differs from 'move to point skill' in many ways. We can study and test each skill independently for the best performance.

Trajectory. Trajectory is a set of methods that generate velocities to control robots. Since each skill focuses on different points, for example, some skills require fast motion while others require accurate position, different trajectories are created to serve these various needed.

Control. Control, which controls the robots' movement, is the lowest layer of AI. Control receives control data such as velocity, angular velocity from 'Trajectory.'

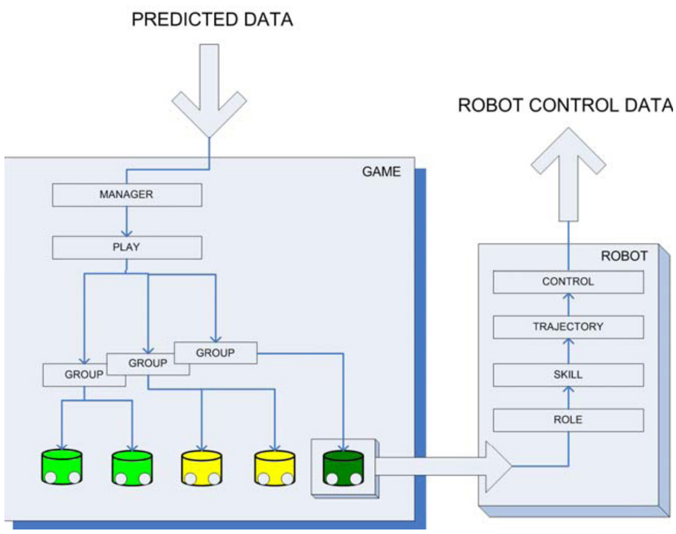

Fig.12 AI's layer architecture

6 Conclusion

After many years of research and development, we have faced various kinds of problems and also found some good solutions for those. The experience we gained results in the improvement of our team in the competitions both national and international. Nevertheless, we will continually develop our system for even better performance.

References

1. Arkin, R.C.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
2. Brown, R.G. and P.Y.C. Hwang.: Introduction to Random Signals and Applied Kalman Filtering. John Wiley & Sons (1992)
3. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston (1991)
4. Pornsarayouth, S.: Accuracy Improvement of Omni-Directional Mobile Robot using Gyroscope and Accelerometer. MA thesis, Chulalongkorn University (2008)
5. A Flexible New Technique for Camera Calibration, http://research.microsoft.com/en-us/um/people/zhang/calib/
6. Altium Nexar TSK51, http://www.keil.com/dd/chip/3730.htm
7. Blob extraction library, http://opencv.willowgarage.com/wiki/cvBlobsLib#Algorithm
8. Small Size Robot League, http://small-size.informatik.uni-bremen.de
9. Solenoids and Actuators, http://robots.freehostia.com/Solenoids/Solenoids.html
10. Tsai's Camera Calibration, http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/
11. Tsai's Camera Model, http://www.cs.cmu.edu/~rgw/TsaiDesc.html