Team RND Description Paper

Team leader : Jeong–ho Lee
Team member: Young–rae Jo, Seong–wook In, Ari Seo,

Seoul National University of Technology Mechatronics Lab
Room 317, Frontier Hall, Seoul National University of Technology, Gongneung 2(i)–dong,
139–743 Nowon–gu, Seoul, Republic of Korea
http://cafe.naver.com/snutrnd

Robocube–tech
Room 101–A, Business Incubation Center, Seoul National University of Technology,
Gongneung 2(i)–dong, 139–743 Nowon–gu, Seoul, Republic of Korea
http://www.robocube.co.kr

Abstract : This paper describes the specifications of our humanoid soccer robots. We have developed this robot for Robocup2015. This robot has 20 DOF, 1–CMOS sensors, 3–axis rate gyro sensors, 2–axis tilt sensor. This robot has fully autonomous system fitting the humanoid Robocup competition. We have researched stabilization and artificial intelligence of the humanoid robot. This robot is integrated our whole technique.

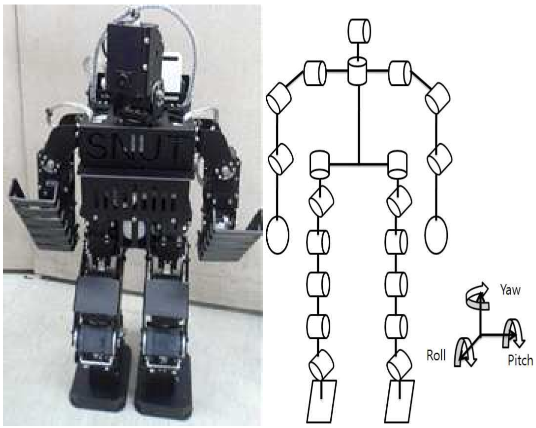
1. Introduction

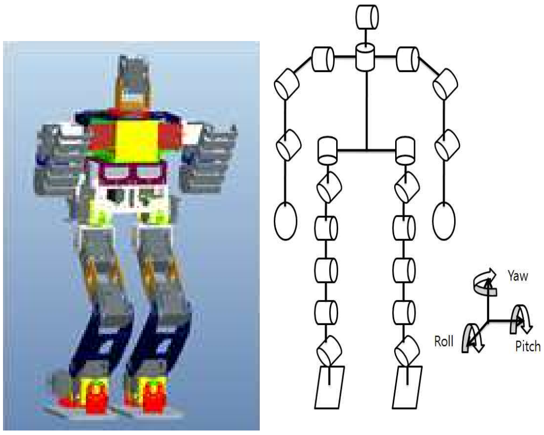
Robocup Humanoid League is an annual human–interactive robot competition which evolves every year with increasing size and the number of the participating teams, and thus creates more challenges for the participants. The participating robot must be able to fully recognize and chase the object (ball), and identify the goal and location of itself. To make this possible, RnD (Seoultech, Republic of Korea) is researching humanoids as well as developing techniques and skills that are required under Professor Kim' s supervision. RnD also is funded by Robocube. Inc. for further research possible. The team focuses on the in state competitions for advancement, and has been rewarded in various competitions such as Robogames, RobotJapan, Firacup, IRC(International Robot Contest).

The following documentation is written to illustrates the undergoing research of hardware and software for participation of RoboCup.

2. Mechanical Specifications

The following kid–sized robots are made from scratch in the team. Their names are RND–Spirit and RND–Ultra, representing the motto of the team. These robots are developed in a good fit for Kid Size robot soccer. RND–Spirit comes with 515mm height and 3.8 kg mass. RND–Ultra comes with the height of 552mm and 4 kg of mass. [Figure 2.1] and [Figure 2.2] illustrate the visualizations and the structures of the robots. [Table 2.1] and [Table2.2] summarizes the angle range of each of the joints. RND–Spirit (Kicker–Robot) contains 20 joints including the neck, arm limits. The actuator DYNAMIXEL RX–64 (manufactured by ROBOTIZ) is used for the joint movement of hip pitch, Knee Pitch , Angle pitch and the rest joints are accompanied by the actuator DYNAMIXEL RX–28 (manufactured by ROBOTIZ). RND–Ultra contains also 20 joints including the leg, upper body, neck and arms. It is different from RND–Spirit in actuator form robot. RND–Ultra is accompanied by actuator MX–26, MX–64 which are MX series. And also above two robots use different embedded PC. That' s why the two robot are a quiet different in structure.





<Figure 2.1 RND–Spirit Picture and structure> <Figure 2.2 RND–Ultra is in the process of being made>

Part	Joint	Actuator	Movable Range[deg]
Leg	Hip Roll	(RX–28) * 2	–15 to 30
	Hip Pitch	(RX–64) * 2	–20 to 55
	Hip Yaw	(RX–28) * 2	–45 to 45
	Knee Pitch	(RX–64) * 2	–45 to 45
	Ankle Roll	(RX–28) * 2	–35 to 35
	Ankle Pitch	(RX–64) * 2	–45 to 45
Arm	Shoulder Pitch	(RX–28) * 2	–70 to 70
	Shoulder Roll	(RX–28) * 2	–80 to 50
	Elbow Rolll	(RX–28) * 2	–70 to 70
Neck	Neck Yaw	(RX–28) * 1	–80 to 80
	Neck Roll	(RX–28) * 1	–70 to 50

< Table 2.1 RND–Spirit Joint angle limits >

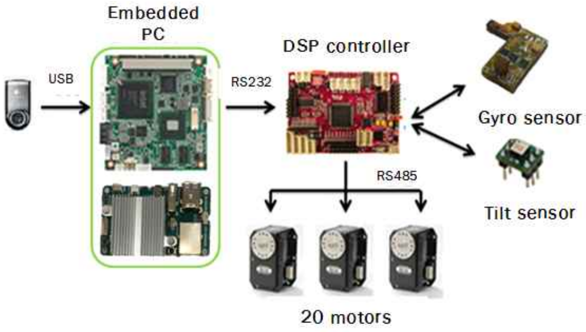
Part	Joint	Actuator	Movable Range[deg]
Leg	Hip Roll	(MX–28) * 2	–15 to 30
	Hip Pitch	(MX–64) * 2	–20 to 55
	Hip Yaw	(MX–28) * 2	–45 to 45
	Knee Pitch	(MX–64) * 2	–45 to 45
	Ankle Roll	(MX–28) * 2	–35 to 35
	Ankle Pitch	(MX–64) * 2	–45 to 45
Arm	Shoulder Pitch	(MX–28) * 2	–50 to 50
	Shoulder Roll	(MX–28) * 2	–80 to 50
	Elbow Rolll	(MX–28) * 2	–70 to 70
Neck	Neck Yaw	(MX–28) * 1	–80 to 80
	Neck Roll	(MX–28) * 1	–70 to 50

< Table 2.2 RND–Ultra Joint angle limits >

3. Electrical specifications and sensor feedback

– Overall Electrical Specification –

[Figure 3.1] show you our whole composition. We use two robot with different PC, one is operated by windows and the other is odroid with Linux. As using odroid, robot' s embeded PC is much smaller than last year. PC is provided by web–cam image data. And PC calculate the image data every frame with image processing. When finish calculating, using RS232 communication, transmit to DSP controller some instructions and information of situation at every time. When DSP controller get information, it determines goal position, or motion communicating with all motors using RS485 communication. When robots fall off by an external force, the tilt sensors deliver data to get it up on its feet. And gyro Sensor prevents the robot from falling, by providing data to DSP Controller. And DSP controller transmits the compensation value to motors.



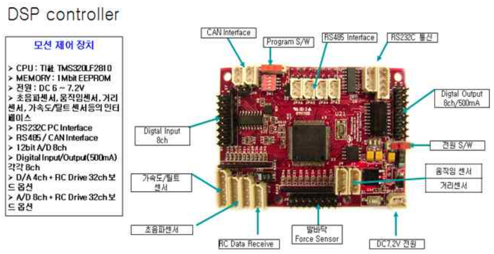
< Figure 3.1 whole composition >

– Motion Controller –

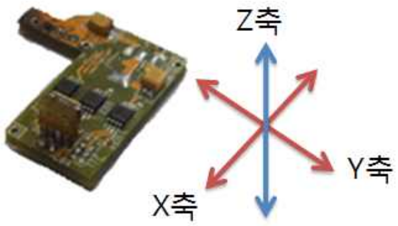
The RnD team uses TMS320F2810 by TI for the robot' s motion controller. It controls 20 servo motors, and accepts data provided by the gyro sensor and the tilt sensor. For the communication with servo motors, our team use RS485 communication. The camera give the visual data to the embedded PC which sends the motion command via RS232 communication. As shown in the [Figure 3.2], the controller is open to additional accessibility. The current rule forbids us from using sensors which help measuring distance with electromagnetic radiation, so we are only using 3–axis gyro sensor and 2–axis Tilt sensor.

– Sensor –

3–axis gyro sensor is self–made in our team. Using this sensor, the balance of the robot' s x–axis and y–axis is stabilized. For stabilization of the x–aixs, the sensor sends the counter balancing data of the hip joint (front and back, in y–axis works with left and right), and both ankles. [Figure 3.3] display gyro sensor.



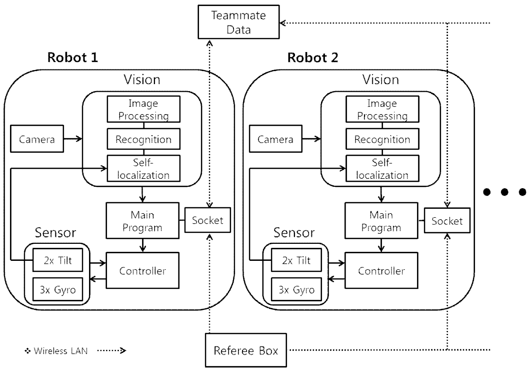
< Figure 3.2 DSP controller >



< Figure 3.3 Gyro Sensor >

4. Software specifications

- **Vision:** Refer to Pt. 6
- **Teammate Data:** Robots communicate by using the UDP Communication.
- **Main Program:** Compares, calculates the data provided by the Vision and communications through other robots, and delivers the most appropriate command to the controller.
- **Controller:** Operates the robot by controlling the motors based on the command provided by the Main Program (refer Pt.7)
- **Sensor:**
  - 2x Tilt: In situation where the robots is fallen, the sensor commands the motors to upright position and makes it stand up on its feet.
  - 3x Gyro: Stabilizes the robot in motion.



< Figure 4.1 SoftWare >

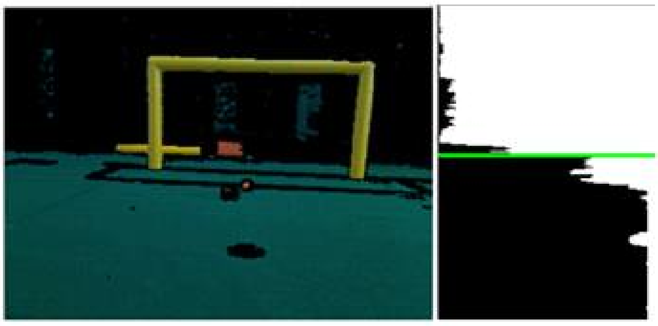
5. Image processing

5−1. Image processing

Our visualization is based on the efficient algorithm, which allows the robot to recognize the field, goal, and the ball to draw out the virtual image of the field, including the x and y axis and data of goal post. This is done by Webcam, with the speed of 20 frames per second. The colorization of the virtual map of the filed in done by Hue, Saturation, and Value (HSV) for the sake of distinguished visualization.

− Field

As shown in [Figure 5−1.1], the field is visualized by the HSV, and with the start of the field, using the horizontal projection, we can get the boundary of field. To limit the range of the visual within the limits of the field and ball, we get the field boundary.



< Figure 5−1.1 Field horizental Projection >

− Ball

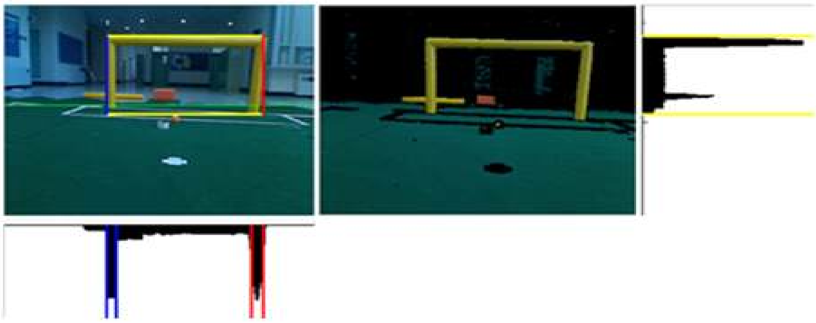
As shown in the [Figure 5−1.2] , the robot recognizes and calculates the coordinates of the ball based on the information obtained by labeling the HSV image. The coordinated location becomes a primary focus point, allowing the robot to detect the less light− sensitive surface of the ball and determine the final coordinate of the ball. Using this method, the final location of the ball becomes more accurate.

We also added the partial image editing program to improve the pace of visual recognition. This program lets the visual to shrink down to the main area of the focus, allowing the robot to calculate the data even faster. The distance between the robot and the ball is evaluated from the data given by the images.

< Figure 5–1.2 Ball Image Processing (Cam Image/ HSV Image/ ROI Image) >

– Goal Post

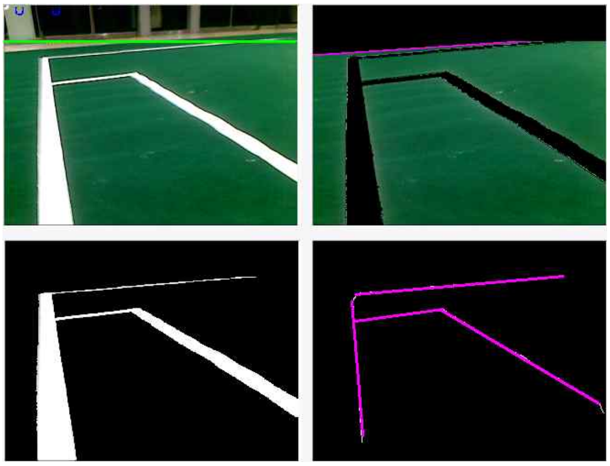
The goal, as shown in [Figure 5–1.3], is projected with vertical and horizontal visuals. Using the horizontal projection, the goal' s height and the upper goal are measured. Using the vertical projection, the length, direction, and width of the goal are measured. These data are utilized only for self locating purpose. Data from the images help the robot determine the location of the goal, ball and robot itself on the field.



< Figure 5–1.3 Goal Post Projection >

– Line

In line image processing, robots detect under the field boundary to get information of lines. There are some information of line under field boundary as shown [Figure 5–1.4]. Each line has slope, location, kind and singularity. And these help self–localization a lot.



< Figure 5–1.4 field, line image processing (real, field, line HSV, line >

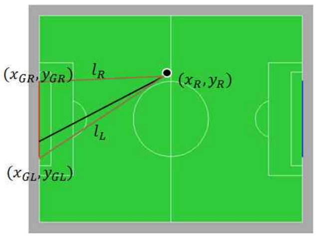
5-2. Localization

It is indeed significant for the robot to know its location in robot soccer. The localization is the key to continue the game, and create an efficient pathway to score a goal. Also, it is important for the robot to be able to distinguish between the ally’ s and opponent’ s goals. Since the two goals are identical, the robot has nothing to rely on but the localization skills.

Our team utilizes the inversely countered visualization in respect to the distance of the object between the robot and the ball. Our localization system relies on the perspective measurement of the size of the object in respect to the distance. The robot uses the information provided from the goal to determine its location. The number of the pixels present in the image is utilized accurately determine the distance. [Figure 5-2.1]

< Figure 5-2.1 >

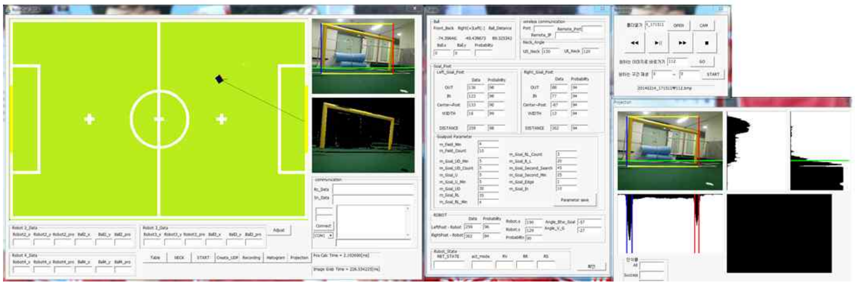
Using the distance provided by the image of the goal on the field, the robot determines its location as shown in [Figure 5-2.2]. (robot’ s coordinates (x_R, y_R) , coordinates of the goal’ s left pillar (x_{GL}, y_{GL}) , the distance between the robot and the left pillar l_L , coordinates of the goal’ s right pillar (x_{GR}, y_{GR}) , the distance between the robot and the right pillar l_R .)



$$(x_R - x_{GR})^2 + (y_R - y_{GR})^2 = l_R^2$$
$$(x_R - x_{GL})^2 + (y_R - y_{GL})^2 = l_L^2$$
$$-2(y_{GR} - y_{GL})y_R + y_{GR}^2 - y_{GL}^2 = l_R^2 - l_L^2$$
$$y_R = \frac{l_R^2 - l_L^2 - (y_{GR}^2 - y_{GL}^2)}{-2(y_{GR} - y_{GL})}$$
$$x_R = \sqrt{l_R^2 - (y_R - y_{GR})^2} + x_{GR}$$

< Figure 5-2.2 >

We have also implemented the possibility application for more accurate result of coordinates provided by the method above. The possible coordinates are based on the calculation of the differently set significance of the coordinates. In a situation where the possibility of calculating the right direction is low, the robot uses the opponent’ s robot’ s coordination, and reverse to make it its own.



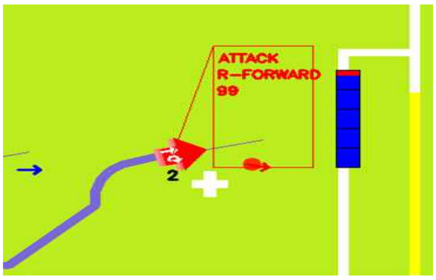
< Figure 5-2.3 >

5-3. Simulation

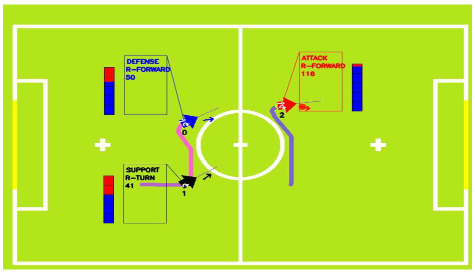
We add a interesting simulation program in our robots. We are taking simulation program with wireless communications, playing soccer similar to real soccer playing. There are two modes in simulation program, Animation mode and UDP communication mode.

First, in animation modes, we can control each robot' s location and angle in a field. When we play in animation mode, can consider logic from most cases and scenarios. As for explaining of our logic in simulation program, decision of position, action and goal point or angle, following a ball, and expression of states. Robots do not go straight toward a ball. They run to a ball or goal position in a optimal path which logic determine in every time shown as [Figure 5-3.1]. And from its path, action also can determined. Distance and angle from each robot to ball, robot calculate each probability value about position. And from those values, they can decision position such as attack, defense and support. And using hysteresis, can prevent sensitive determination.

In the UDP communication mode, each robot transmit information from image processing and self-localization to other robots via transfer protocol called UDP. They know other robot' s information so they play soccer in a same logic in animation modes. And we can know which robot has problem or not such as communication failure, shut down of PC. The most important thing in this simulation program is recognizing state of each robot at a glance. We draw robot in arrow shape and that direction means robot angle in field. In position, and use three distinct types of color. And robot has information box next to him. In this box, we can write significant information about robot. Bar blocks are also beside information boxes, it show us value we want. [Figure 5-3.2] display robots in field.



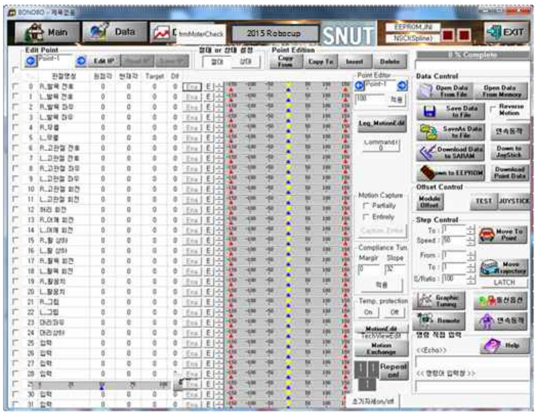
< Figure 5-3.1 Run to a ball in a optimal path>



< Figure 5-3.2 States of robots in field>

6. Motion control

We also created our own motion control as shown [Figure 6.1]. One motion can be made of 40 individual points, and the delay time and overall time are editable. This type of motion can be saved up to 256 motions and is put in use when needed. Basic movements such as Forward, Backward, Right, Left and Spin are preloaded, and one may combine any of these movements to create a unique movement. The same movement is carried on repeatedly. Movement calibration, pattern, pace, and speed are customizable by commands.



< Figure 6.1 Motion Editor Program >

7. Conclusion

Our vision algorism is completed with the data obtained by the data from the camera. With those data, robot can do self–localization on the basis of probability. And the images of the camera allow the robot to determine the distance from the ball, as well as its coordinates so that the robot is fully functional in calculating the pathway. One of advantage that our team would like to point out is that most of motor controllers and sensors that are used in our robots are built from scratch in our lab. In 2009, we participate in Robocup for the first time, and we advanced to the quarterfinal. Since then, we have studied about algorism and structure. We have intention of involvement to Robocup from 2013, so we developed this system in this year. Also, we are in the process of writing papers with the techniques which we made and used in Robocup.

< Figure 7.1 International Robot Contest 2014 >

< Figure 7.2 FIRA RoboWorld Cup 2014 >

8. reference

[1] J. Y. KIM and Y. S. KIM, 2012, WHOLE–BODY MOTION GENERATION OF ANDROID ROBOT USING MOTION CAPTURE AND NONLINEAR CONSTRAINED OPTIMIZATION, International Journal of Humanoid Robotics.

[2] The study to obtain data of human body's motion and data generation of robot movements T.Y. BAE AND Y.S. KIM ,2013,which use the optimization method, KSMTE

[3] Dae–Jin Kim, A study on improving image based position recognition for a bipedal soccer robot using probability technique, SNUT, August 2009

[4] Do–ryang Kim, A study on stable walking of a bipedal robot using inertial sensors on the body, SNUT ,August 2009

[5] tae–Young Bae, A study on application of motion capture equipment about the creation for stable operation of Android robot, NID ,August 2013