KUDOS Team Description
for Humanoid Kidsize League of RoboCup 2014

JuSeong Shin, HoJin Jeon, ChanSeo Park, JaeHyun Kim, Seunghun Lee,
InWon Cho, SeRi Lee and Baek-Kyu Cho

Robotics and Control Laboratory
School of Mechanical Engineering, Kookmin University, KOREA
E-mail:baekkyucho@kookmin.ac.kr
Web:http://rclab.kookmin.ac.kr
http://kudos.kookmin.ac.kr

Abstract. Although RoboCup2013 was our first tournament, we were able to advance to Round Robin 2. Therefore, we were ranked in the top 16 and we showed the potential to grow into a better team. This year, for RoboCup2014, we have newly developed our own robot, KUbot. Some parts of KUbot that are not loaded with much weight were fabricated using a three-dimensional printer. For localization in KUbot, we implemented simultaneous localization and mapping (SLAM) using an extended Kalman filter based on vision processing, an odometer, and a compass-like sensor. In RoboCup2014, we aim to showcase KUbot, a robot that can better play soccer using SLAM and new algorithms. This paper briefly describes our preparation and research for RoboCup2014.

Keywords: KUDOS, KUbot, Humanoid robot, Locomotion algorithm, Model predictive control, Localization algorithm, SLAM

1 Introduction

Our team is called KUDOS, which is an acronym for “Kookmin University Dream of Soccer”. We named our team so for two reasons. First, the original objective of RoboCup is to field a team of robots that can win against the human soccer World Cup champions by 2050. Realizing this objective is the dream of robotics and soccer players. In this light, we chose “dream of soccer” as part of our team name. Second, “kudos” is a synonym of “prestige”. Because we aim to achieve prestige at the Humanoid KidSize League of RoboCup, the meaning of “kudos” well matches our team objective. Figure 1 shows our team mark and logo.[1]

Although RoboCup2013 was our first tournament and we had to compete with strong teams such as AUTMan (rank 2) and ZJUDancer (rank 3), we achieved good results—1 win, 1 draw, and 4 losses—to advance to Round Robin 2.[2] Therefore, we were ranked in the top 16 and we showed the potential to grow into a better team.

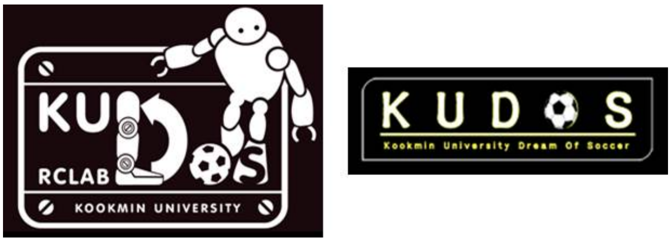


Fig. 1. KUDOS mark and nameboard

This year, for RoboCup2014, we have newly developed our own robot, KUbots, as shown in Figure 2. KUbot can stand for both “Kookmin University Robot” and “KUDOS’s Robot”. We designed KUbot based on an open-platform robot and we applied the experience we had gained from RoboCup2013.[3][4] For the robot soccer game, we developed new vision, motion, localization, and locomotion algorithms. KUbot discriminates the ball, goalpost, line, and other robots using a camera located in its head. It localizes the location of all objects and itself using simultaneous localization and mapping (SLAM) and then determines an appropriate soccer strategy.[5] Robot soccer involves various motions such as kicking, goalkeeping, and getting up autonomously. To implement stable and fast locomotion for the robot, we used the model predictive control (MPC) method.[6] We also developed an online controller, a damping controller, a landing orientation controller, and a landing position controller.[7]

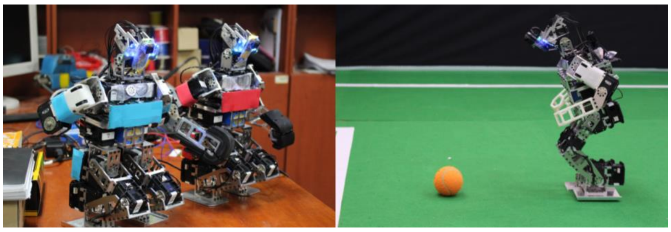


Fig. 2. KUbots

The remainder of this paper is organized as follows. Section 2 describes the hardware of KUbot. Section 3 presents the algorithm for robot soccer. Sections 4, 5, 6, and 7 respectively discuss robot vision, localization, locomotion, and motion.

2 Hardware

In RoboCup2013, we used commercially available robots. However, for RoboCup2014, we resolved to develop our own robots. Ultimately, we developed KUbot, a new robot based on an open-platform robot, as shown in Figure 3.

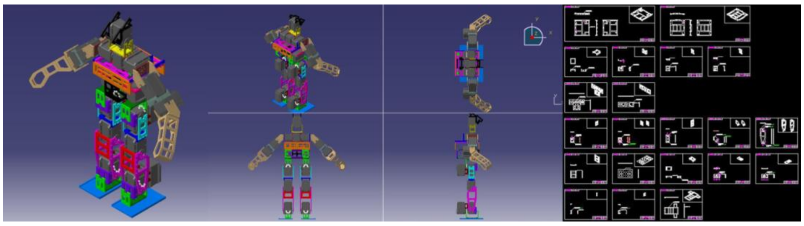


Fig. 3. Design of KUbot

KUbot is a kidsize humanoid robot that has a height of 456 mm, weight of 3 kg, and 20 degrees of freedom (DOFs). Table 1 lists its specifications in detail.

Table 1. Specifications of KUbot

Development term	2013.7∼
Height	456 mm
Weight	3 kg
Walking speed	23 cm/min(Up to 46 cm/min)
Number of DOFs	20 in total.(6 DOF Legs x 2 = 12, 3 DOF Arms x 2 = 6, 2 DOF Head x 1 = 2)
Actuator	DYNAMIXEL MX-28 Actuator (Maxon Motor + Reduction gear + Absolute encoder)
Control unit	Main Control Unit: SBC-fitPC2i Sub Control Unit: CM-730
Camera	Logitech C905
Inertia measurement unit	Gyro: 3-Axis LYPR540AH Acceleration: 3-Axis ADXL335
Power	Li-Po Battery 11.1 V, 1800 mAh
Other specs	Sound: Speaker, MIC Display: Head, Eyes, and Body LED

We focused on three points in developing KUbot. First, we designed KUbot based on open-platform robots. Therefore, we could conform to RoboCup's design rules for robots and increase the compatibility of the hardware and software. As a result, we could use the controller, actuator, and other devices of open-platform robots. Second, the strength of the frames was increased by increasing the robot thickness; this should prevent it from deforming upon collisions such as falling down. Third, we used a Stratasys Dimension 1200es three-dimensional (3D) printer[8] for fabricating some robot parts that are not loaded with much weight, such as hands and external covers. Figure 4 shows the process for making KUbot's hand.

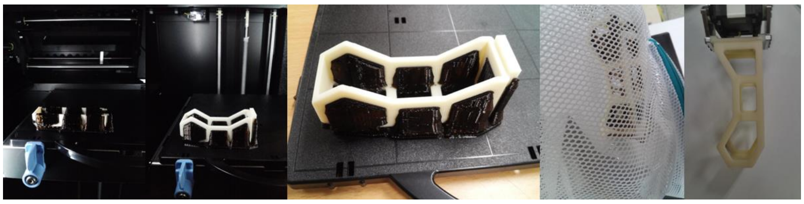


Fig. 4. Process for making KUbot's hand

3 Algorithm for Robot Soccer

We developed two algorithms for robot soccer—one for a field player and one for a goalkeeper—both of which are similar. Figure 5 shows the former. It consists of four steps: (1) the robot finds the ball, (2) the robot approaches the ball, (3) the robot finds a goalpost, and (4) the robot kicks the ball.

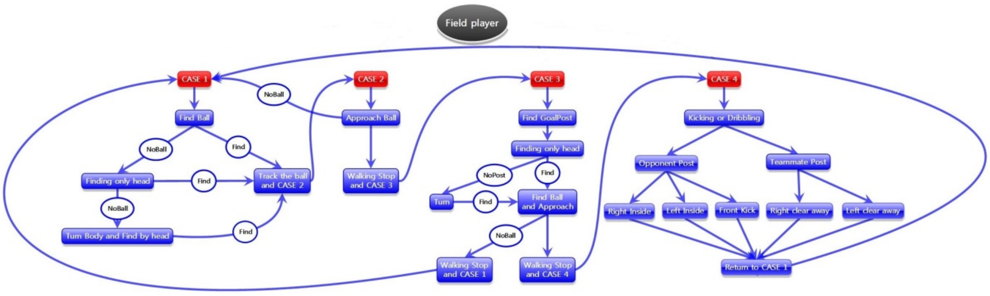


Fig. 5. Algorithm for a field player

4 Vision

It is essential for a soccer robot to recognize its environment such as the goalpost, ball, and lines. Therefore, we spent a lot of time studying vision processing. Because light is the most sensitive element, we converted the RGB-formatted image from the camera module into an HSL-formatted image in which it is relatively much easier to correct the effects of light.[9] To optimize the time required for vision processing, we reduced the image size and deleted unrequired pixels, such as those outside the green field, as shown in Figure 6.(B). Then, we detected lines by Hough transform. The ball and goalpost were detected by the color value in the optimized image, as shown in Figure 6.(C).[10] Finally, KUbot can recognize objects such as the ball, field, and goalposts.

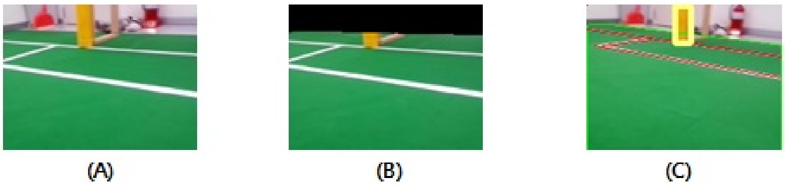


Fig. 6. KUbot's image

5 SLAM

The recent competitions in RoboCup show that SLAM is essential for achieving good results. We have implemented SLAM using an extended Kalman filter based on comprehensive information obtained using the estimated relative positions of the objects recognized in Section 4 by an odometer, gyrosensor, and so on.[11]

5.1 Distance Estimation

Because we know the camera's height and angle, it is possible to calculate the distance of a specific point inside the image from the robot. Similarly, we can estimate the distance of objects in the image, such as a ball and lines. These values are very useful in localization. Equations (1) and (2) are used to calculate the forward-directional distance of a target pixel in the image. Different formulas are used for the far side (d_1) and the close side (d_2) with respect to the central horizontal line.

$$d_1 = htan\theta + \frac{htan\alpha}{cos^2\theta\left(1 - tan\theta tan\alpha + \frac{Y}{60-Y}\right)} \tag{1}$$

$$d_2 = htan\theta - \frac{htan\alpha\left(\frac{Y}{60} - 1\right)}{(cos\theta + sin\theta tan\alpha)cos\theta} \tag{2}$$

In addition, equation (3) calculates the side-directional distance of a target pixel.

$$l = d(tan\beta)\frac{X - 80}{80} \tag{3}$$

Table 2 explains the parameters used in Eqs. (1), (2), and (3).

Table 2. Parameters used in formulas

Image size	160 x 120
(X, Y)	Target pixel's x,y coordinates in image
θ	Tilt angle of camera direction
α	Half of camera's vertical angle of view
β	Half of camera's horizontal angle of view
h	Height of camera from ground
d_1, d_2	Calculated distance in forward direction
l	Calculated distance in vertical direction

5.2 Odometer

An odometer provides the robot's position and direction by integrating its every step. Even though an odometer can contain a large error owing to certain situations such as the robot's slipping or falling, it is useful to conduct SLAM.

5.3 Compass-like sensor

Because a compass sensor (geomagnetic sensor) measures the absolute direction, it is useful to use SLAM. However, a compass sensor cannot be used in KUbot because it is affected by the magnets in the motors. Therefore, we imitate a compass sensor by integrating the yaw directional signal of a gyrosensor.

6 Locomotion

6.1 Locomotion

Locomotion pattern generation. Many studies have focused on humanoid walking.[7][12-14] We have formed a walking pattern by using MPC. MPC is a common control method for generating online motion for a dynamic system.[6] Figure 7 shows an example application of MPC to HUBO: the robot moved five steps forward and stopped.

Locomotion control algorithm. We used a damping controller, landing orientation controller, and landing position controller for stable locomotion.[15][16] The damping controller was designed to eliminate sustained structural oscillation. Thus, it is important to maintain balance. The landing position controller helps the robot land quickly and safely by controlling the ankle. When a robot walks on uneven terrain, the actual landing time of the foot may differ from the prescribed landing time. To solve this problem, we used the landing position controller to lengthen the stride on the next swing phase by the amount of loss and to slowly stretch the foot after the landing is fully completed. In general, the landing orientation controller is applied to the swing foot during landing and the damping controller is applied to the supporting foot after landing. The landing position controller modifies the prescribed position of the swing foot when the foot touches the ground earlier than its prescribed time.

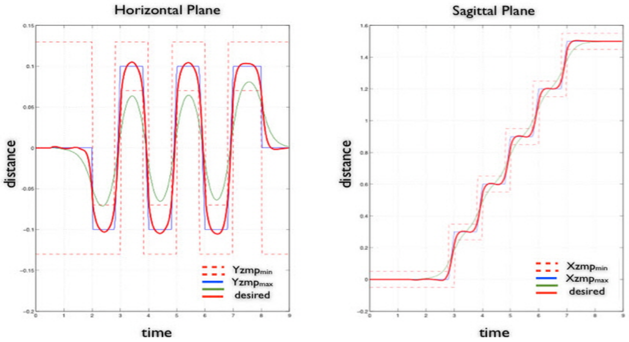
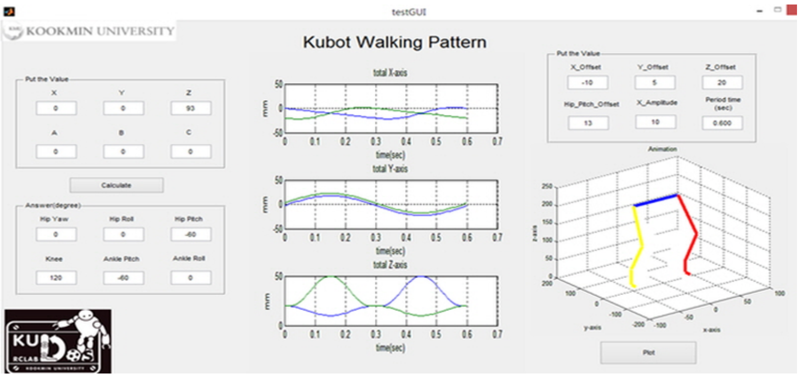


Fig. 7. COM and ZMP trajectory when the robot moves five steps forward and stops

6.2 Walking pattern generation using MATLAB

We developed a GUI program for generating walking patterns using MATLAB. We can find suitable patterns using the program instead of a real robot and save time in developing a walking pattern. The program calculates an inverse kinematics[17] to verify the joint angles of the robot, as shown in the left-hand side of Figure 8. In addition, we can confirm the motion of the robot and check the position of the zero moment point of the robot by animation, as shown in the right-hand side of Figure 8.



Put the Value

X	Y	Z
0	0	93
A	B	C
0	0	0

Answer(degree)

Hip Yaw	Hip Roll	Hip Pitch
0	0	-60
Knee	Ankle Pitch	Ankle Roll
120	-60	0

Put the Value

X_Offset	Y_Offset	Z_Offset
-10	5	20
Hip_Pitch_Offset	X_Amplitude	Period time (sec)
13	10	0.600

Fig. 8. GUI of KUbot walking pattern

7 Conclusion

We had participated in RoboCup2013 and gained good experience. For RoboCup2014, we newly developed our robots and continuously studied various methods such as SLAM, locomotion, and simulation for robot soccer. We will participate in RoboCup regularly and grow further as a team. We aim to show much better performance in RoboCup2014.

References

1. Jae Hwan Kim, Ju Seong Shin, Dong Hyun Ahn, Jin Young Kong, Sung Hoon Yang, Eun Mi Jang, and Baek-Kyu Cho, “KUDOS Team Description Paper for Humanoid Kidsize League of RoboCup2013”, 2013.
2. `http://www.robocup2013.org/humanoid-kid-size-schedule`
3. Inyoung Ha, Yusuke Tamura, and Hajime Asama, Jeakweon Han and Dennis W Hong, “Development of Open Humanoid Platform DARwIn-OP”, SICE Annual Conference, pp.2178-2181, 2011.
4. Ill-Woo Park, Jung-Yup Kim, Jungho Lee and Jun-Ho Oh, “Mechanical Design of Humanoid Robot Platform KHR-3”, IEEE-RAS International Conference on Humanoid Robots, pp.321-326, 2005.
5. Durrant-Whyte, H.; Bailey, T. “Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms”. Robotics and Automation Magazine 13 (2): 99-110. 2006
6. Pierre-Brice Wieber, “Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations”, IEEE-RAS International Conference on Humanoid Robots, pp. 137-142, 2006.
7. Baek-Kyu Cho, Jung-Hoon Kim, and Jun-Ho Oh, “Online Balance Controllers for Running in the Humanoid Robot, HUBO2”, Advanced Robotics, Volume 25, Issue 9-10, pp. 1209-1225, 2011.
8. `http://www.stratasys.com/3d-printers/design-series/performance/dimension-1200es`
9. Joblove, George H. and Greenberg, Donald. “Color spaces for computer graphics”. Computer Graphics 12 (3): 20-25. 1978
10. Richard O. Duda and Peter E. Hart. “Use of the Hough Transformation to Detect Lines and Curves in Pictures”. 1971
11. Einicke, G.A.; White, L.B. “Robust Extended Kalman Filtering”. IEEE Trans. Signal Processing 47 (9): 2596-2599. 1999
12. Baek-Kyu Cho and Jun-Ho Oh, “Dynamic Balance of a Hopping Humanoid Robot using a Linearization Method”, International Journal of Humanoid Robotics, Volume 9, Issue 3, 2012.
13. Jung-Hoon Kim, Jong-Hyun Choi, and Baek-KyuCho, “Walking Pattern Generation for a Biped Walking Robot Using Convolution Sum”, Advanced Robotics, Volume 25, Issue 9-10, pp. 1115-1137, 2011.
14. Baek-Kyu Cho, Ill-Woo Park, and Jun-Ho Oh, “Running Pattern Generation of Humanoid Biped with a Fixed Point and Its Realization”, International Journal of Humanoid Robotics, Volume 6, Issue 4, pp. 631-656, Dec. 31, 2009.
15. Jung-Hoon Kim.A “Study on the Realization of Dynamic Gait for a Biped Humanoid Robot”, PhD Thesis. Korea Advanced Institute of Science and Technology, Daejeon, South Korea, 2004.
16. Jung-Yup Kim. “On the Stable Dynamic Walking of Biped Humanoid Robots”, PhD Thesis. Korea Advanced Institute of Science and Technology, Daejeon, South Korea, 2006.
17. Saeed B.Niku., “Introduction to Robotics Analysis, Systems, Applications”, Prentice Hall, 2005.