Skuba 2009 Team Description

Kanjanapan Sukvichai[1], Piyamate Wasuntapichaikul[2], Jirat Srisabye[2], and Yodyium Tipsuwan[2]

[1] Dept. of Electrical Engineering, Faculty of Engineering, Kasetsart University.
[2] Dept. of Computer Engineering, Faculty of Engineering, Kasetsart University.
50 Phaholyothin Rd, Ladyao Jatujak, Bangkok, 10900, Thailand
baugp@hotmail.com
http://iml.cpe.ku.ac.th/skuba

Abstract. This paper is used to describe the Skuba Small-Size League robot team. Skuba robot is designed under the World RoboCup 2009 rules in order to participate in the ssl competition in Graz, Austria. The overview describes both the robot hardware and the overall software architecture of our team.

Keywords: Small-size, Robocup, Vision, Robot Control, Artificial Intelligence.

1 Introduction

Skuba is a small-size league robot team from Kasetsart University [1], which entered the World RoboCup competition since 2006. Skuba got the third place in the world ranking last year from the World RoboCup 2008 in Suzhou, China. During the last year competition, problems about the robot low level controller and multi-agent game plans were revealed.

This year, robot low level controller is redesigned along with the new open loop skills. Both are implemented in Skuba robot 2009. Omni-directional wheels robot is one of the most popular mobile robot which is used in most of the teams because of its maneuverability. The major problem for many teams is how to tune the low level controller gains. The surface parameters are changed according to the time because the carpet is damaged from the robot wheels. Therefore, all of the low level controller gains for every wheel have to be adapted every match. Torque control scheme is implemented in this year in order to solve this problem. Torque controller consists of the PI control and the torque converter. The new idea of the modified robot kinematics is implemented in order to make the open loop game plans possible.

The vision system process two video signals from the cameras mounted on top of the field. It computes the positions and the orientations of the ball and robots on the field then transmit the information back to the AI system

The AI system receives the information and makes strategic decisions. The decisions are converted to commands that are sent back to the robots via a wireless link. The robots execute these commands and set actions as ordered by the AI system.

2 Robot

The major issue about Skuba 2008 robot is the robustness. Most mechanical parts are general purpose grade aluminum which can be damaged easily. The Skuba 2009 robot is built based on Skuba 2008 design with some modified parts. The major parts were built using aircraft grade aluminum alloy to improve strength such as the kicker and the chip-kicker.

Each robot consists of four omni-directional wheels which are driven by 30 watt Maxon flat brushless motors. The kicker has ability to kick the ball at speeds up to 14 m/s using a solenoid. The chip-kicker is a large powerful flat solenoid attached with a 45 degree hinged wedge located on the bottom of the robot which can chip the ball up to 7.5 m before it hits the ground. Both of the solenoids are driven from two 2700μF capacitors charged to 250V. Kicking devices are controlled by a separate board located below the middle plate. The kicking speed is fully variable and limited to 10 m/s according to the rule.

The controller of the robot hardware is done by using a single-chip Spartan 3 FPGA from Xilinx. The FPGA contains a soft 32-bit microprocessor core and peripherals. This embedded processor executes the low level motor control loop, communication and debugging. The motor controller, quadrature decoder, kicker board controller, PWM generation and onboard serial interfaces are implemented using FPGA. The robot receives control commands from the computer and sends back the status for monitoring using a bidirectional 2.4GHz wireless module. A Kicker board is a boost converter circuit using a small inductor, the board is seperated from the main electronics for safety.

The size of robot has a diameter of 178 mm and a height of 144mm.The dribbler covers to 20% of the ball. The 3D model of the robot and the real robot are shown in Fig. 1 and Fig. 2 respectively.

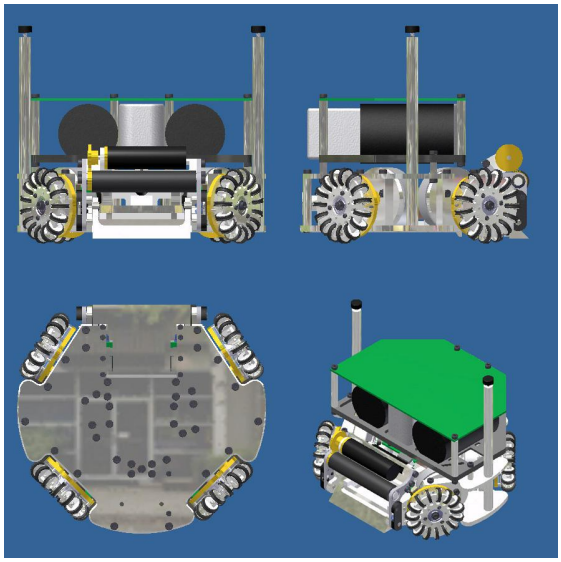

Fig. 1. 3D mechanical model of Skuba 2009 robot

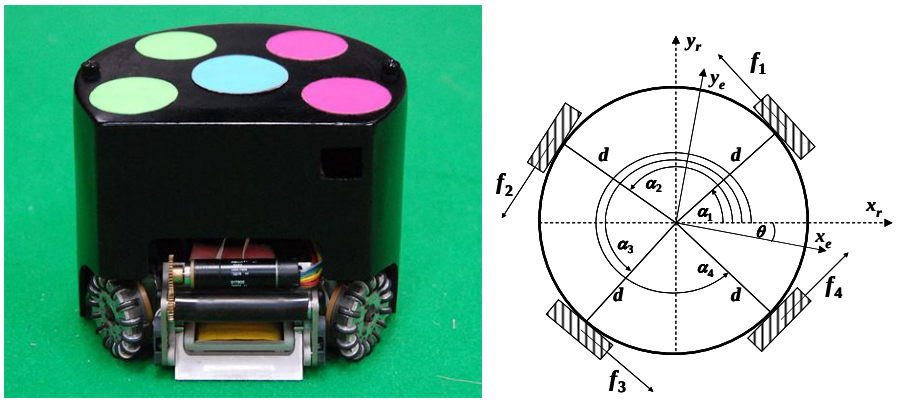

Fig. 2. Skuba 2009 robot

2.1 Robot Dynamics

Dynamics of a robot is derived in order to provide information about its behavior. Kinematics alone is not enough to see the effect of inputs to the outputs because the robot kinematics lacks information about robot masses and inertias. The dynamic of a robot can be derived by many different methods such as Newton's law [2] [3] and Lagrange equation [4]. In this paper, Newton's law is used to solve the robot dynamic equation.

Newton's second law is applied to robot chassis in Fig 2 and the dynamic equation can be obtained as (1) though (3).

$$\ddot{x} = \frac{1}{M}(-f_1 \sin\alpha_1 - f_2 \sin\alpha_2 + f_3 \sin\alpha_3 + f_4 \sin\alpha_4) - \vec{f}_f\Big|_x \tag{1}$$

$$\ddot{y} = \frac{1}{M}(f_1 \cos\alpha_1 - f_2 \cos\alpha_2 - f_3 \cos\alpha_3 + f_4 \cos\alpha_4) - \vec{f}_f\Big|_y \tag{2}$$

$$J\ddot{\theta} = d(f_1 + f_2 + f_3 + f_4) - T_{trac} \tag{3}$$

where,

- $\ddot{x}$ is the robot linear acceleration along the x-axis of the global reference frame
- $\ddot{y}$ is the robot linear acceleration along the y-axis of the global reference frame
- M is the total robot mass
- f_i is the wheel i motorized force
- $\vec{f}_f$ is the friction force vector
- α_i is the angle between wheel i and the robot x-axis
- $\ddot{\theta}$ is the robot angular acceleration about the z-axis of the global frame
- J is the robot inertia
- d is the distance between wheels and the robot center
- T_{trac} is the robot traction torque

The robot inertia, friction force and traction torque are not directly found from the robot mechanic configuration. These parameters can be found by experiments. The robot inertia is constant for all different floor surfaces while the friction force and traction torque are changed according to floor surfaces.

The friction force and traction torque are not necessary found at this point because these two constraints are different for different floor surfaces and their effect can be reduced by using the control scheme which is discussed in the next topic. The wheel force can be written in a motor torque form as:

$$\ddot{x} = \frac{\mathbf{1}}{M}\left(-\frac{\tau_{m1}}{r}\sin\alpha_1 - \frac{\tau_{m2}}{r}\sin\alpha_2 + \frac{\tau_{m3}}{r}\sin\alpha_3 + \frac{\tau_{m4}}{r}\sin\alpha_4\right) - \vec{f}_f\Big|_x \tag{5}$$

$$\ddot{y} = \frac{\mathbf{1}}{M}\left(\frac{\tau_{m1}}{r}\cos\alpha_1 - \frac{\tau_{m2}}{r}\cos\alpha_2 - \frac{\tau_{m3}}{r}\cos\alpha_3 + \frac{\tau_{m4}}{r}\cos\alpha_4\right) - \vec{f}_f\Big|_y \tag{6}$$

$$\ddot{\theta} = \frac{d}{J}\left(\frac{\tau_{m1}}{r} + \frac{\tau_{m2}}{r} + \frac{\tau_{m3}}{r} + \frac{\tau_{m4}}{r}\right) - T_{trac} \tag{7}$$

where,

r is a wheel radius

Equation (5) though (7) show that the dynamic of the robot can be directly controlled by using the motor torques.

2.2 Modified Robot Kinematics

The pervious topic, the dynamics of the robot is derived. Although the dynamics can be correctly used to predict the robot behavior but it is hard to directly implement and it needs a long computing time. In this topic, the regular mobile robot kinematics is modified. First the friction force and traction torque vector are defined as a system disturbance. The normal kinematics can be written as:

$$\zeta_r = \psi \cdot \zeta_{Designed} \tag{13}$$

where,

$$\zeta_r = \begin{bmatrix} \dot{\phi}_1 & \dot{\phi}_2 & \dot{\phi}_3 & \dot{\phi}_4 \end{bmatrix}^T$$

$$\zeta_{Designed} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$$

$$\psi = \begin{bmatrix} \cos\theta\cdot\sin\alpha_1 + \cos\alpha_1\cdot\sin\theta & \sin\theta\cdot\sin\alpha_1 - \cos\alpha_1\cdot\cos\theta & -d \\ \cos\theta\cdot\sin\alpha_2 + \cos\alpha_2\cdot\sin\theta & \sin\theta\cdot\sin\alpha_2 - \cos\alpha_2\cdot\cos\theta & -d \\ \cos\theta\cdot\sin\alpha_3 + \cos\alpha_3\cdot\sin\theta & \sin\theta\cdot\sin\alpha_3 - \cos\alpha_3\cdot\cos\theta & -d \\ \cos\theta\cdot\sin\alpha_4 + \cos\alpha_4\cdot\sin\theta & \sin\theta\cdot\sin\alpha_4 - \cos\alpha_4\cdot\cos\theta & -d \end{bmatrix}$$

Designed robot velocity ($\zeta_{Designed}$) is used to generate robot's wheel angular velocity vector (ζ_r). This wheels angular vector is the control signal which is sent from PC to interested mobile robot. The output linear velocity ($\zeta_{Captured}$) is captured by a bird eye view camera. The output velocity contains information about disturbances, therefore by comparing the designed velocity and the output velocity. The output velocity can be defined as (14) when assuming that disturbance is constant

for the specific surface. The disturbance is modeled and separated to the disturbance from the robot coupling velocity and the disturbance from the surface friction.

$$\zeta_{Captured} = (\psi^{\dagger} + \varepsilon) \cdot \zeta_r + \Delta \tag{14}$$

where,

$\psi^{\dagger}$ is the pseudo inverse of the kinematic equation

ε is the disturbance gain matrix due to the robot coupling velocity

Δ is the disturbance vector due to the surface friction

The disturbance matrices can be found from experiments. The first designed robot velocity ($\zeta^{1}_{Designed}$) is applied to the robot in order to get the first output velocity ($\zeta^{1}_{Captured}$) in the first experiment. The first experiment is repeated in the second experiment with the second designed robot velocity ($\zeta^{2}_{Designed}$) and the second output velocity ($\zeta^{2}_{Captured}$) is captured. The disturbance matrices now can be found by adding (13) to (14) for both experiments.

$$\zeta^{1}_{Captured} = (\psi^{\dagger} + \varepsilon) \cdot \psi \cdot \zeta^{1}_{Designed} + \Delta \tag{15}$$

$$\zeta^{2}_{Captured} = (\psi^{\dagger} + \varepsilon) \cdot \psi \cdot \zeta^{2}_{Designed} + \Delta \tag{16}$$

Subtract (15) by (16):

$$\zeta^{1}_{Captured} - \zeta^{2}_{Captured} = (\psi^{\dagger} + \varepsilon) \cdot \psi \cdot \zeta^{1}_{Designed} - (\psi^{\dagger} + \varepsilon) \cdot \psi \cdot \zeta^{2}_{Designed}$$

$$\varepsilon = ((\zeta^{1}_{Captured} - \zeta^{2}_{Captured}) \cdot (\zeta^{1}_{Designed} - \zeta^{2}_{Designed})^{\dagger} - I) \cdot \psi^{\dagger} \tag{17}$$

Substitute (17) to (15) and Δ is found.

2.3 Motor Model and Torque Control

A Maxon brushless motor is selected for the robot. The dynamic model of the motor can be derived by using the energy conservation law as shown in [5]. The dynamic equation for the brushless motor is

$$u \cdot \frac{\tau_m}{k_m} = \frac{\pi}{\mathbf{30,000}} \cdot \dot{\phi} \cdot \tau_m + R \cdot \left(\frac{\tau_m}{k_m} \right)^2 \tag{8}$$

where,

u is the input voltage

τ_m is the motor output torque

k_m is the motor torque constant

$\dot{\phi}$ is the motor angular velocity

R is the motor coil resistance

Equation (8) is not easy to implement to the control law. Therefore, this equation has to be modified by using the Maxon parameters relationship, which is shown in its datasheet, and the final dynamic equation of the motor is

$$\tau_m = \left(\frac{k_m}{R}\right)\cdot u - \left(\frac{k_m}{R\cdot k_n}\right)\cdot \dot{\phi} \tag{9}$$

where,

k_n is the motor speed constant

The control scheme is set using the discrete Proportional-Integral control law and torque dynamic equation (9). The control system runs at 600Hz cycle [1]. The error between desired angular velocity and real filtered angular velocity of each wheel is the input of the PI controller with the PI gains K_p and K_I respectively. The controller is shown in Fig. 2 and the control law can be described as (10) though (12).

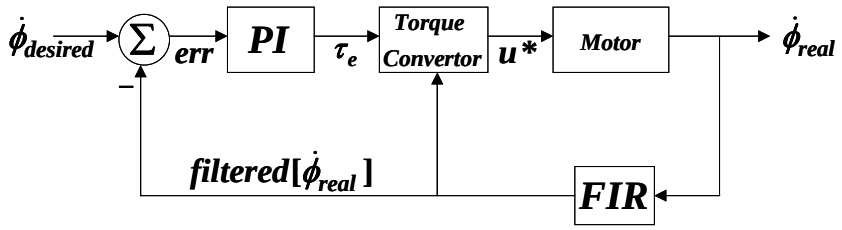

Fig. 3. Torque controller scheme

$$err[j] = \dot{\phi}_{desired}[j] - filtered[\dot{\phi}_{real}[j]] \tag{10}$$

$$\tau_d[j] = k_p \cdot err[j] + k_I \cdot \sum_{j=1}^{N}(err[j]) \tag{11}$$

$$u^*[j] = \frac{\tau_d[j]}{\left(\frac{k_m}{R}\right)\cdot V_{cc} - \left(\frac{k_m}{R\cdot k_n}\right)\cdot filtered[\dot{\phi}_{real}[j]]} \tag{12}$$

where,

N is the number of samples

V_{cc} is the driver supply voltage

The output from (12) is converted to the Pulse Width Modulation (PWM) signal and directly used as input signal for every poles of the motor. The difference between the regular discrete PI controller for the wheel angular velocity and the torque controller is the torque converter block which is shown in Fig 3 and defined as (12).

3 Vision

Our vision structure diagram is shown in Fig 4.

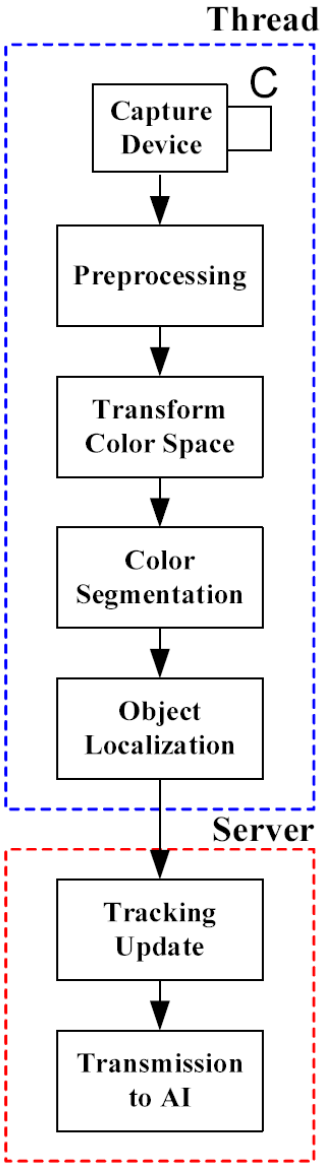

Fig. 4. Vision system structure

- **Capture Device**
  Our team applies the global vision and uses the output signal of two cameras. We employ *AVT Stingray F-046C* 1394b firewire camera which is capable of grabbing 780 x 580 images at 62 fps

- **Preprocessing**
  The preprocessing is used to improve the quality of the image.

- **Transform Color Space**
  We transform color model to the HSV space, which consists of a hue, a saturate and a value. The HSV space is more stable than RGB space in different light properties.

- **Color Segmentation**
  The color segmentation assigns each image pixel into color classes. Currently, we classify and segment color by CMVision2.1 library [6].

- **Object Localization**
  After color segmentation, we receive all the color regions. The filtering process discards incorrect regions. Then, object localization computes the position and orientation of objects in the field from the final regions.

- **Tracking Update**
  Objects which is received from localization has a lot of noise, so we need to track it. Our approach is working by the Kalman Filter.

- **Transmit to AI**
  This component consists of network link communication between the vision system and system

3.1 Camera Calibration

Camera calibration is a part in Object localization. We compute the internal and external parameters of the cameras using the Tsai [7] algorithm. These parameters are used to correct the distortion produced by the camera lenses.

4 Multilayer, Learning-based Artificial Intelligence

Multi-layered learning based agent architecture is applied to the RoboCup domain. Upper layer are used to control the activation and priority of behaviors in layers below, and only the lowest layer interacts directly with the robot. This year, the program is rebuilt from scratch by using strategy structure based on "StrategyModule" from Cornell Big Red 2002.

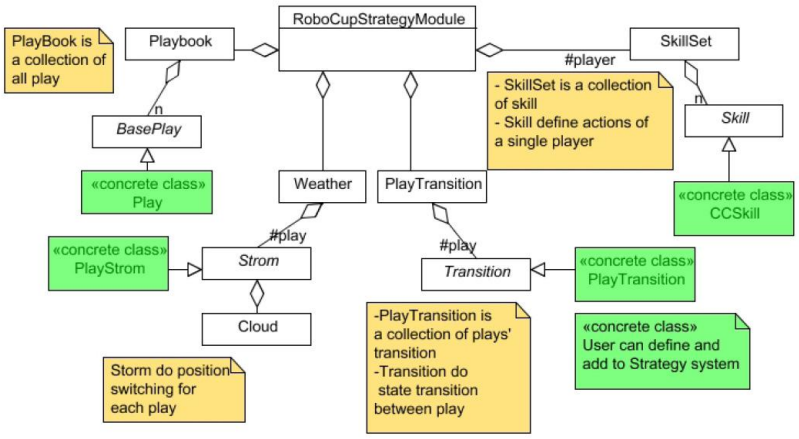

Fig. 5. Strategy structure

4.1 Play

Play illustrates a specific global state of the AI and the general goal the positions are attempting to achieve at a given time. The system will transit from one play to another by learning-based method. We score the successful play more than the failed play.

4.2 Skill

Skill is a basic action of robot, such as "MoveToBallskill" or "Kickskill". We can use a Neural Network for train each skill independently for the best efficiency. The modified robot kinematics is used in our new skill such as open loop pass and kick skill.

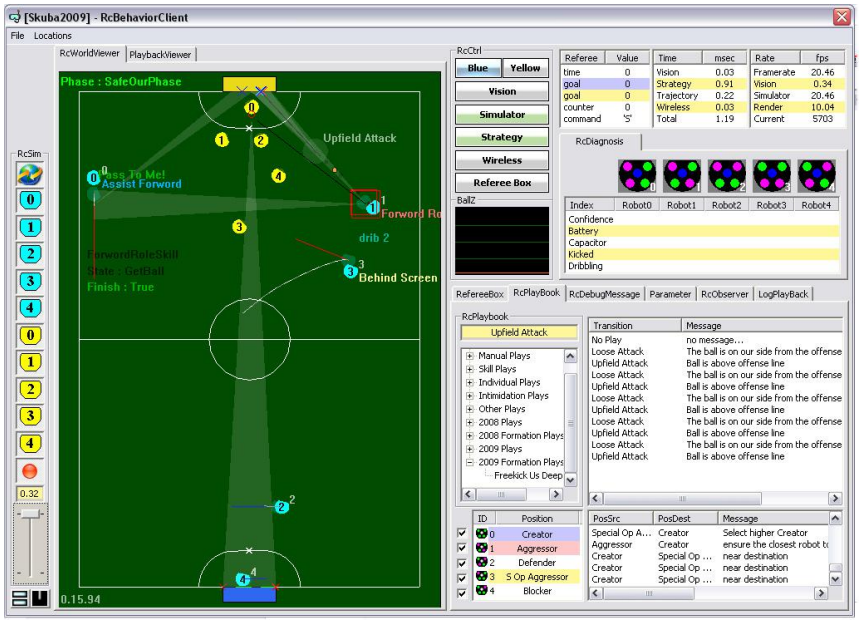

Fig. 6. Skuba's user interface

5 Conclusion

The new robot hardware design and the new approach of low level controller have been implemented and they improved the speed, precision, and flexibility of the robots. With some filters, we could acquire precisely coordinates of all players. The modified robot kinetics is used in the simulator and it can improve the robot overall efficiency. We believe that the RoboCup Small-Size League is and will continue to be an excellent domain to drive research on high-performance real-time autonomous robotics. We hope that our robot performs better in this competition than the last year competition. We are looking forward to share experiences with other great teams around the world.

References

1. Srisabye, J., Hoonsuwan, P., Bowarnkitiwong, S., Onman, C., Wasuntapichaikul, P., Signhakarn, A., et al., Skuba 2008 Team Description of the World RobCup 2008, Kasetsart University, Thailand.
2. Oliveira, H., Sousa, A., Moreira, A., Casto, P., Precise Modeling of a Four Wheeled Omni-directional Robot, Proc. Robotica'2008 (pp. 57-62), 2008.
3. Rojas, R., Forster, A., Holonomic Control of a robot with an omnidirectional drive, Kunstliche Intelligenz, BottcherIT Verlag, 2006.

4. Klancer, G., Zupancic, B., Karba, R., Modelling and Simulation of a group of mobile robots, Simulation and Modelling Practice and Theory vol. 15 (pp. 647-658), ScienceDirect, Elsevier, 2007.
5. Maxon motor, Key information on – maxon DC motor and maxon EC, Maxon Motor Catalogue 07(pp. 157–173), 2007.
6. Bruce, J.: CMVision realtime color vision system. (The CORAL Group's Color Machine Vision Project) http://www.cs.cmu.edu/˜jbruce/cmvision/.
7. Tsai, R.Y.: A versatile camera calibration technique for high accuracy 3D machine vision using off-the-shell TV cameras and lenses, IEEE Journal of robotics and Automation, 1987.