

Skuba 2010 Team Description

Piyamate Wasuntapichaikul¹, Jirat Srisabye¹, and Kanjanapan Sukvichai²

¹ Department of Computer Engineering

² Department of Electrical Engineering

Faculty of Engineering, Kasetsart University

50 Phaholyothin Rd., Ladyao, Jatujak, Bangkok, 10900, Thailand

baugp@hotmail.com, jiratto@gmail.com, fengkp@ku.ac.th

Abstract. This paper presents a brief description of Skuba, a Small-Size League RoboCup robot team. The robot system is designed under the RoboCup 2010 rules in order to participate in the RoboCup competition in Singapore. The system consists of two main components which are explained in the Robot and the Software Architecture section. The major improvements in this year are the automatic calibration software for the Motion Controller and the Kicker which are explained in those two sections.

1 Introduction

Skuba is a small-size league soccer robot team from Kasetsart University, which has entered the RoboCup competition since 2006. We got the championship last year from the RoboCup 2009 in Graz, Austria and another championship in December from RoboCup China Open 2009 in Dalian, China.

The robot system consists of two main components: the robot hardware and the software. The software makes strategic decisions for the robot team by using information about the object positions from the vision system. The global vision system run by the shared vision software, SSL-Vision, uses two cameras mounted over field. The software executes plans by calculating the robot actions and then sends the commands to each robot.

This year, the main focus of our development is the automatic calibration. Even though every robot is built by the same design and material, there are still some errors from the manufacturing and assembling process. Furthermore, some parameters can be changed according to the competition environments. These issues involve the need of calibration for the accuracy of the system. The kicker and the low level controller parameters used to be manually calibrated. These calibrations are time consuming and need manpower to do the experiments. The automatic calibration process simply uses the same procedure as the manual calibration does, but it's done automatically by the software.

2 Robot

Our team has ten identical robots, six of them were built in 2008 and another four were built in 2009 with some minor changes in material and mechanical design. We are not planning to make any major changes to the design. The robot hardware is the same as used in last year. More details about the robot hardware can be found in [1].

Each robot consists of four omni-directional wheels which are driven by 30 watt Maxon flat brushless motors. Each motor is equipped with a 360 CPR optical encoder to provide signals for speed measurement. The robot uses the dribbling device to improve ball handling capability, the dribbler is a round bar covered with a silicone tube and connected to a high speed brushless motor. The bar can spin up to 13000 rpm. The kicker has ability to kick the ball at speeds up to 14 m/s using a solenoid. The chip-kicker is a flat solenoid attached with a 45 degree hinged wedge located on the bottom of the robot which can kick the ball up to 7.5 m before it hits the ground. Both of the solenoids are driven from two 2700 μ F 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 runs at 30 MIPS and interconnected peripherals. This embedded processor executes the low level motor control loop, communication and debugging. The brushless motor controller, quadrature decoder, kicker board controller, PWM generation and onboard serial interfaces are implemented using FPGA logic gates. 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 separated from the main electronics for safety.

The robot has a diameter of 176 mm and a height of 147mm. The dribbler covers up to 20% of the ball diameter. The 3D model of the robot and the real robot are shown in Fig. 1 and Fig. 2 respectively.

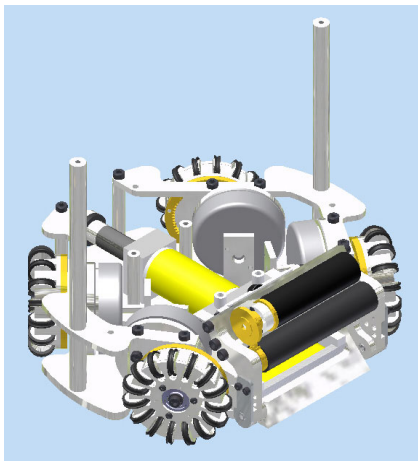


Fig. 1. 3D mechanical model of the robot

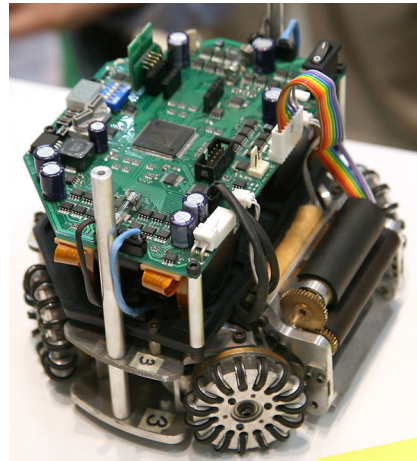


Fig. 2. Real robot

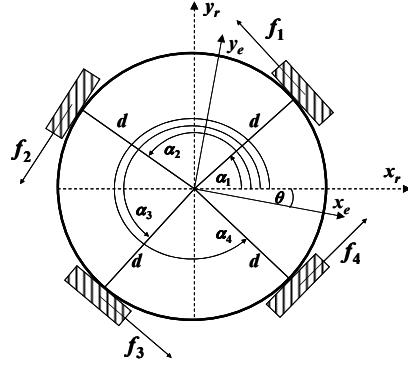


Fig. 3. Robot wheel configuration

2.1 Modified Robot Kinematics

Normally, when the software sends the velocity command to the robot, it doesn't perform any velocity feedback control and it assumes that the robot's motion controller has already taken care of this. But due to the loss from friction, wheel slippage and other real world problems, the robot cannot move as fast as commanded. The regular robot kinematics describes an ideal situation where there's no system disturbance. In order to control the robot more accurately, the robot kinematics is modified with the some disturbance parameters. The friction force and traction torque vector are defined.

The normal kinematics can be written as:

$$\zeta_r = \Psi \cdot \zeta_{Desired} \quad (7)$$

where,

$$\zeta_r = [\dot{\phi}_1 \quad \dot{\phi}_2 \quad \dot{\phi}_3 \quad \dot{\phi}_4]^T$$

$$\zeta_{Desired} = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$$

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

Desired robot velocity ($\zeta_{Desired}$) 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 ($\zeta_{Observed}$) is observed by a bird eye view camera. The output velocity contains information about disturbances, therefore by comparing the desired velocity and the output velocity. The output velocity can be defined as (8) when assuming that disturbance is constant for the specific surface. The two disturbances are modeled.

$$\zeta_{Observed} = (\psi^\dagger + \varepsilon) \cdot \zeta_r + \Delta \quad (8)$$

where,

- ψ^\dagger is the pseudo inverse of the kinematic equation
- ε is the disturbance gain matrix due to the robot coupling velocity friction
- Δ is the disturbance vector due to the surface friction

The disturbance matrices can be found from experiments. From data in last year, the disturbance vector of the surface friction is constant but the coupling velocity friction is a nonlinear function with respect to the robot translation and angular velocity. With these two disturbance parameters, the robot command can be compensated and result in the actual robot output velocity command.

The surface friction is easy to find just by using two observed experimental data while the coupling velocity friction matrix can be estimated using the calibration software. The software performs the experiment by running the robot at different speeds and observing the output velocity from the robot. Then, the disturbance ε can be estimated by using a second order polynomial least squares fitting method.

By using modified kinematics to generate the control command, the robot can move more accurately. The comparison of the experimental result is shown in Fig. 4.

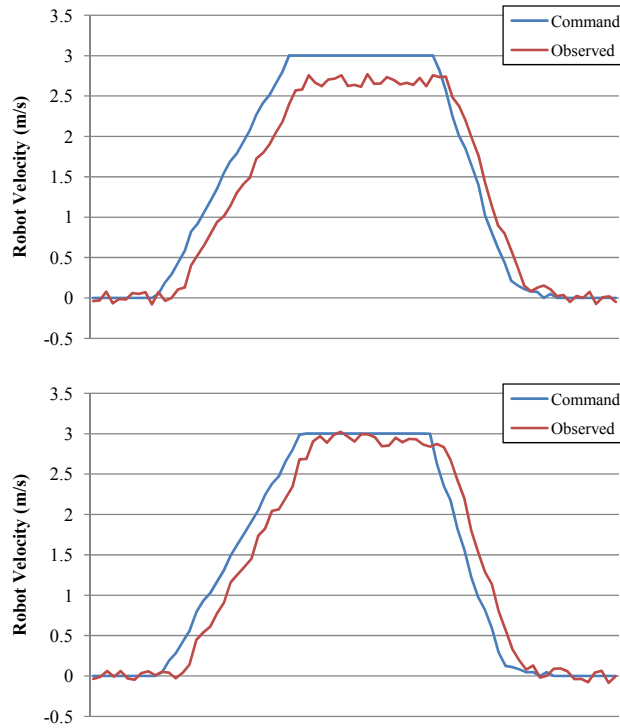


Fig. 4. The robot observed velocity profile using normal kinematics (top) and modified kinematics (bottom)

3 Software Architecture

The overall software architecture is illustrated in Fig. 5. The software consists of several modules organized as a multilayer architecture. This software has been being continuously developed since RoboCup 2006 based on the strategy structure of Cornell Big Red 2002's software. More detailed information of the software can be found in [1].

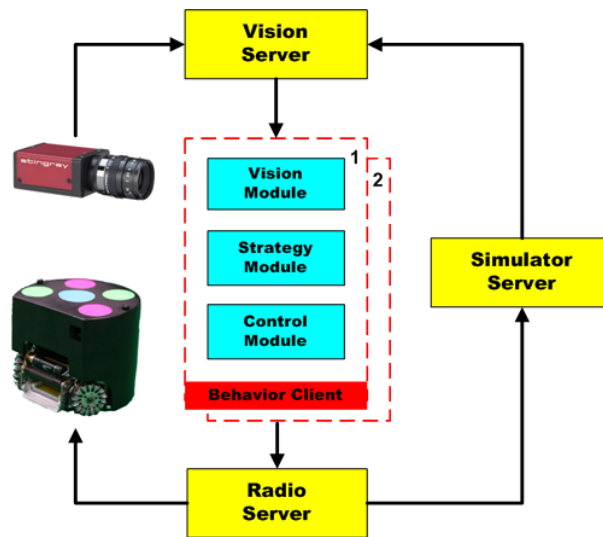


Fig. 5. The software architecture

3.1 SSL-Vision

The use of shared vision system named SSL-Vision is required by the competition rule. This new vision software can be integrated into the system by simply replacing our Vision Server software. With some code changes in the vision protocol, the existing software works with the shared vision system successfully. The SSL-Vision also provides geometric parameters which are very useful for the chip-kicker calibration which is described in the next section.

3.2 Kicker Automatic Calibration

The automatic calibration system is added to the software in order to reduce manpower and time during the team setup process. The calibration software performs sets of experiment in order to obtain the relationship between input and output parameters. Similarly to the motion controller calibration method described in section 2.1, the kicker calibration software is used to estimate the relationship between both the chip-kicking distance or ball speed and the magnitude of the kick command sent

to the robot. In the calibration procedure, the speed of the ball is easily obtained from the vision module while the chip-kicking distance requires trajectory estimation method based on the model given in [2]. This method uses the camera parameters from the SSL-Vision and several ball positions in consecutive frames to approximate the actual trajectory of the ball in the air. Given the parabolic trajectory of the ball, the falling point is calculated and the chip-kicking distance is obtained. The snapshots of the calibration process of the chip-kicker are shown in Fig. 6 and 7. The relationship is then modeled using second order polynomial and can be estimated from the experimental results using least squares polynomial fitting. The example of the kicker calibration result is shown in Fig. 8.



Fig. 6. Snapshots of the chip-kicker calibration process

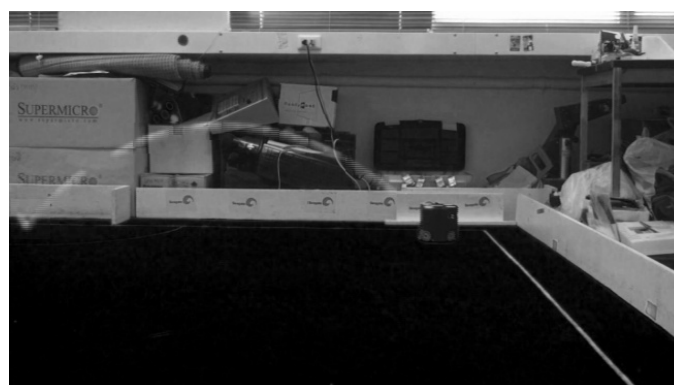


Fig. 7. Parabolic trajectory of the chip-kicked ball

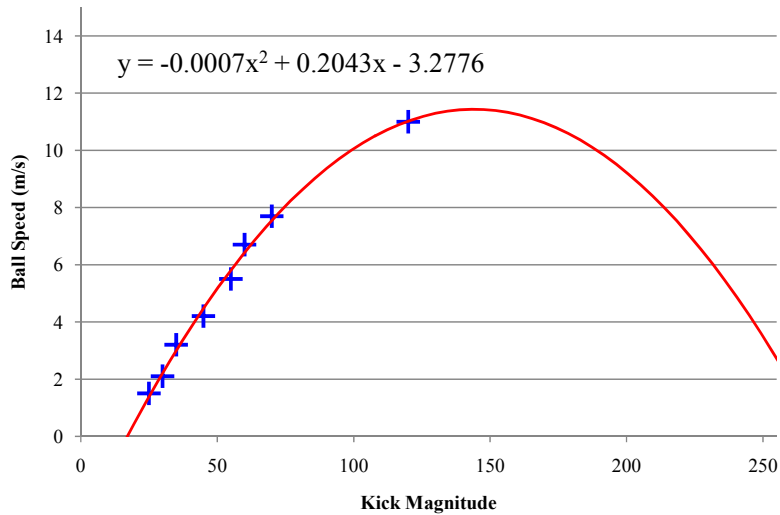


Fig. 8. Relationship between the ball speed and the magnitude of the kick command obtained using second order polynomial least squares fitting

4 Conclusion

Table 1. Competition results for Skuba SSL RoboCup team

Competition	Result
RoboCup Thailand Championship 2005	3 rd Place
RoboCup Thailand Championship 2006	Quarter Final
RoboCup 2006	Round Robin
RoboCup Thailand Championship 2007	3 rd Place
RoboCup Thailand Championship 2008	2 nd Place
RoboCup 2008	3 rd Place
RoboCup 2009	1 st Place
RoboCup China Open 2009	1 st Place

Our system has been continuously improving since the beginning. Last year, we introduced some improvements about the low level motion controller and the robot hardware. The new calibration software for this year was proven to be very useful. In RoboCup China Open 2009, the automatic calibration software greatly reduced the amount of team setup time which allowed us to focus more on the strategic planning. The software which runs the robot team was built in 2006 and improved each year. It has given us very successful competition results for the last several years, the results are summarized in table 1. We hope that our robot team will perform better in this year and we are looking forward to sharing experiences with other great teams around the world.

References

1. Srisabye, J., Wasuntapichaikul, P., Onman, C., Sukvichai, K., et al.: Skuba 2009 Extended Team Description. In: Proceedings of RoboCup 2009
2. Kim, T., Seo, Y., Hong, K.S.: Physics-based 3d position analysis of a soccer ball from monocular image sequences. Sixth International Conference on Computer Vision (1998) 721–726