# YAMAKASA Team Description

Takushi Tanaka[1], Masaru Ishii[1], Yasutoshi Taniguchi[1], and Hiroaki Shirakawa[2]

[1] Fukuoka Institute of Technology
3-9-14 Wajiro-Higashi Higashi-ku Fukuoka 811-0295, Japan
{tanaka, ishii, y-tani}@fit.ac.jp
http://www.fit.ac.jp/~tanaka/robocup.html
[2] Fukuoka Junior College of Technology,
3-9-14 Wajiro-Higashi Higashi-ku Fukuoka 811-0295, Japan
{ha-shira}@fjct.fit.ac.jp

**Abstract.** This paper describes main features of the Yamakasa robot team including Phase Locked Loop for motor speed control and accurate color tuning. ...

## 1 Introduction

The Yamakasa robot soccer project started up in 1998, and the team first participated in Japan Open 2001 Robot Soccer Game held in our Institute. Through these three years, we have improved the robot hardware three times and now we have a stable one with kick device for RoboCup. This paper describes main features of the Yamakasa robot team including Phase Locked Loop for motor speed control and accurate color tuning. The Yamakasa team consists of five robots and a main computer for control.

## 2 Robot Design

The robot has two wheels independently driven by two DC-motors and a kick device driven by a solenoid. The robot receives commands from a main computer through radio. All devices on the robot are controlled by a micro-processor.

### 2.1 Micro-processor

We use a PIC micro-processor (Microchip Technology PIC16F873) for robot control. It is a just size for this purpose having a serial port to receive data, timers for motor control, AD-converters, and PWM-functions. After the reset, it first sets up the frequency of radio receivers, and goes into command interpreter mode. It receives commands through the serial port, interprets the commands, drives the motors for wheels, and drives the solenoid for kick.

### 2.2 Wheel Control using Software PLL

As we used ordinary DC-motors to drive wheels, it was very difficult to control their speed accurately without feedback loop. In our new model, we attached a photo encoder to each wheel. The encoder and the motor forms a phase looked loop. The PIC micro-processor compares the phase of wheel and the phase of reference signal generated by commands, and controls the motor to coincide the phase by Pulse Width Modulation. This phase looked loop is implemented in software. Using this Phase Locked Loop, we can control revolution of wheels accurately.

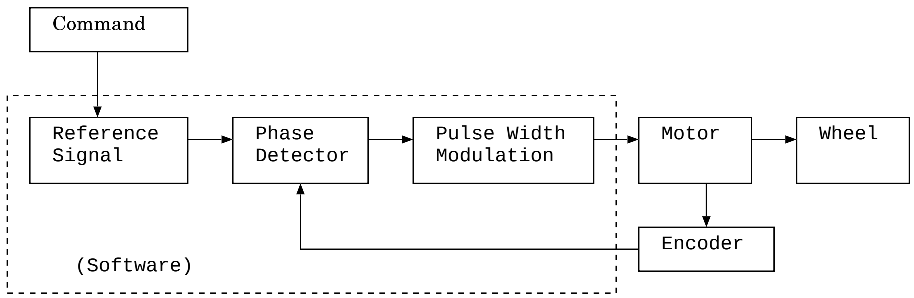


**Fig. 1.** Phase Locked Loop implemented in Software

### 2.3 Kick Device

The kick device attached to the robot front consists of a slide bar, a solenoid, and a micro switch. The micro switch detects a touching ball to the slide bar. When the ball touches the bar, the solenoid pushes the bar with discharging 0.2 F capacitor charged up to 15 volts.

### 2.4 Communication

The commands to robots are sent to radio transmitter through serial port of the main computer with 2.4 kbps. The transmitter is set to one of channels from 259.45 MHz to 260.95 MHz with 0.1MHz step, and sends signals with FSK. The robot has a programmable radio receiver controlled by the PIC micro-processor. The signals from receiver are filtered and sent to the serial port of the micro-processor to convert into commands.

### 2.5 Power Supply

The robot has only one power supply with four NiCd batteries connected in series. This makes easier to maintenance batteries. Two DC-motors for wheels

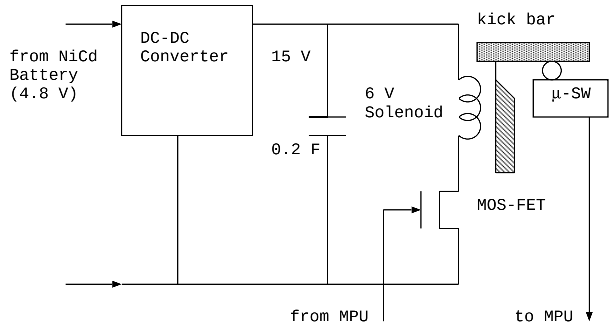


**Fig. 2.** Kick Device

are directory powered by this power supply through H-bridge ICs. The 5 V regulated voltage for receiver and PIC micro-processor is generated by DC-DC converter. The 15 V for solenoid is also generated by the same DC-DC converter.

## 3 Main Computer and Control

We have made the system using Linux (Slackware). The NTSC image signal from global vision is captured by BT878 board. The commands to each robot are sent through a serial port.

### 3.1 Vision System

Last game in Japan, we failed the color tuning because of lighting problem. The soccer field was not lighted uniformly, so we often failed to get the correct positional information of ball and robots. We used RGB-data for color references, but RGB-data directly depended on lighting. It is necessary to get accurate color information in spite of the lighting problem.

This time, we converts RGB-data into TSL-data. Here, T : Tint, S : Saturation and L : Luminance. The color data T and S are basically independent of lighting intensity. Using this TSL-data, we calibrate tables of main eight colors in soccer game before starting game, and get maximum and minimum of each parameter. We have improved the accuracy of getting ball and robots positions with these parameters.

### 3.2 Agent Program

The system has a simple world model which is refreshed by the vision system. Agent programs for five robots and a coach are running on the world model. Four agents except goal keeper and coach is the same program. Each agent makes a short plan according to its environment and sends commands to each robot. The coach agent is different. It watches whole positions of robots and a ball, and orders other agents to change their behavior.

### 3.3 Command to Robots

The command to robots consists of one 8 bit character. Upper 3 bit are used to identify robot. Lower 5 bit decide the robot behavior for 70 msec. As we use 2.4 kbps speed, we can send 8 commands through serial port during each refresh term (33 msec). Each agent for robots sends one command during this term. So, every robot receives their command every 33 msec. If a robot fails to receive the command, it stops after 70 msec. Namely, if a robot fail to receive a command, it continues previous command until next refresh. If agents want to change their robot behavior immediately, they can just send a new command including stop command.

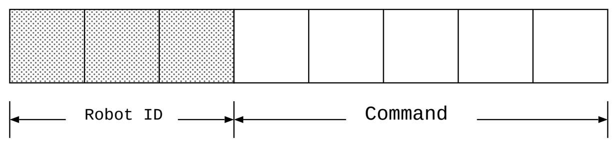


**Fig. 3.** Robot Command

## 4 Conclusions and Future Work

Recently our new system begins to work. Robot motor control and vision system work successfully. We must improve each agent program through games.