Overview of Hibikino-Musashi Hardware and Software

Amir A.F. Nassiraei [1],Shuichi Ishida[1], Noriyuki Shinpuku[1],
Miyuki Hayashi[1], Naoya Hirao[1], Kazutaka Fukuda[1], Kazunori Fujimoto[1],
Kazutomo Takenaka[2],
Ivan Godler[2], Kazuo Ishii[1] and Hiroyuki Miyamoto[1]

[1]Kyushu Institute of Technology, Japan
[2]The University of Kitakyushu, Japan
ishida-shuichi@edu.brain.kyutech.ac.jp

Abstract. This paper presents an overview to the hardware and software of "Musashi robot" developed for the RoboCup Middle-Size League. First we describe the hardware architecture including our approach to the modularity concept. Next, we introduce our robot software flow chart.

1. Hardware system

1.1 Musashi Robot Hardware Architecture

The current hardware configuration of the "Musashi" robot and its fully modular mechatronics architecture including an omni- directional moving mechanism and an omni-vision system is shown in Fig.1 [1-3]. The modular robot architecture provides an effective way to improve reliability, robustness, ease of maintenance and transportation by decomposing hardware complexity into the smaller and compact modules. The robot is equipped with three 70 watts DC motor from Maxon, arranged in the shape of triangle. The maximum nominal motor speed of 7000 rpm is reduced through a planetary gearbox GP42 with the ratio of 12:1. The amplified mechanical torque on the output shaft of the gearbox is transferred to the wheel's shaft through the direct coupling and supported by a pair of the radial ball bearings with housing- T shaped type. The velocity feedback is done by using 500 pulses digital incremental encoders. The velocity of the wheels is controlled by three Faulhaber motor drivers (MCBL 2805), each equipped with a RS232 communication port. The controllers read the pulse trains from the motor encoders and produce PWM output voltages for the motors based on a PID algorithm. The result is a mobile robot with maximum linear speed of 2.4m/s and acceleration of $2.5m/s^2$.

The only sensors using in the "Musashi Robot" are an omni-directional camera, a compass and three DC motor encoders. The electrical power is supplied by a set of Li-Polymer batteries (nominal voltage 25.9V/2Ah). The necessary required voltage for the camera, compass module and the micro computer power supply are produced by converting 25.9V to 12.0V and 5.0V. The DC to DC converter is a typical converter that converts 15V DC to 90V DC to provide power for the solenoid. (Fig. 2). The power consumption of the robot is in average about 40W, and the operation duration of the robot is estimated to be 0.5h. In order to realize the shooting capability, the solenoid mechanism is introduced as a strong and compact kicking device. The key idea is the solenoid is passed an electric current, and the plunger is pushed out.

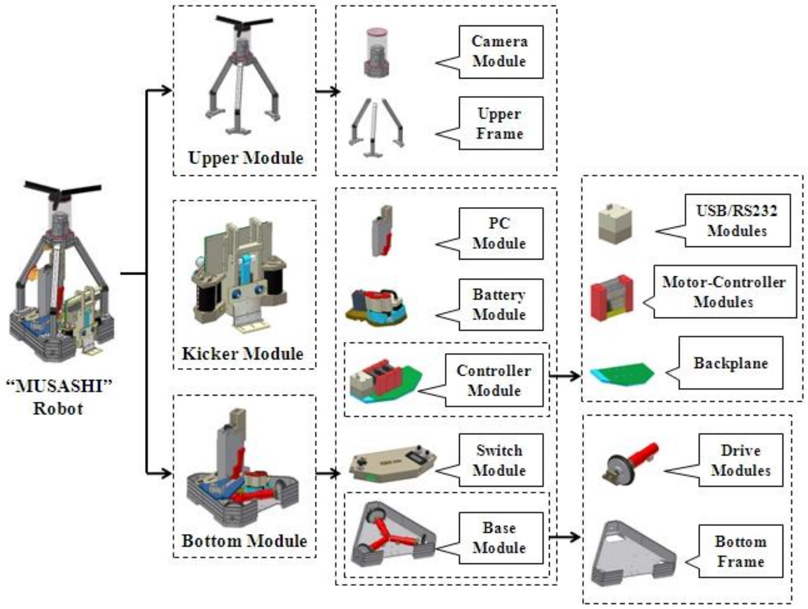

Fig.1 "Musashi" robot hardware configuration and modular architecture

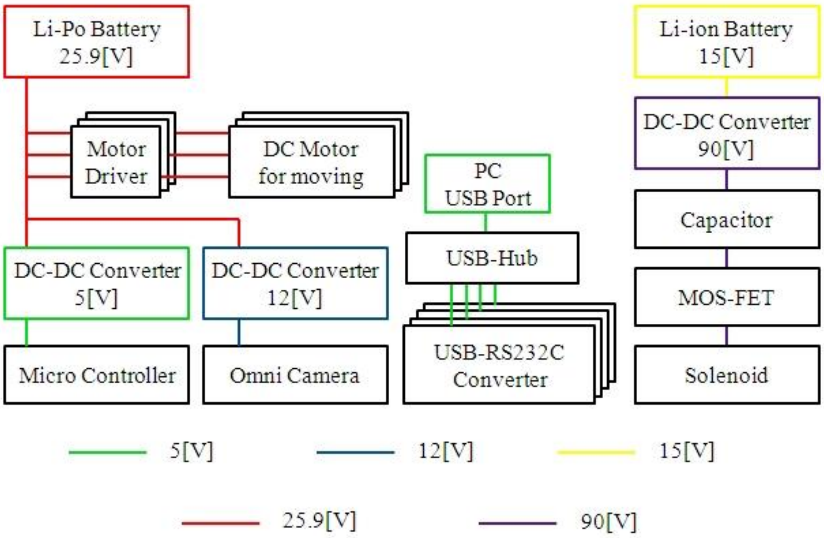

Fig.2 Flowchart of "Musashi" robot power system

1.2 “Modularity” Concept

Step 1: Description of the robot system architecture

To describe and emphasize the concept of modularity, it is necessary to present a short overview of the design of the robot system architecture. Musashi is equipped with a laptop on which the image processing, control, communication and data exchange are performed. The behavior commands such as start, stop, and corner kick are received from a referee box PC located outside the field via a wireless LAN. To achieve a safe, simple, and robust system, the samplings of sensor data and actuator control are executed using conventional interfaces: IEEE 1394, USB, and RS232. The communication between an omni-directional camera and a mounted laptop PC is performed using the IEEE 1394 interface. The laptop PC sends the motor control commands (target velocities) to the motor drivers via a USB interface and USB/serial converters because the motor driver has only a RS232 serial port (Fig. 3). Another USB/RS232 converter is used for communication between the laptop PC and the circuit of the kicking device.

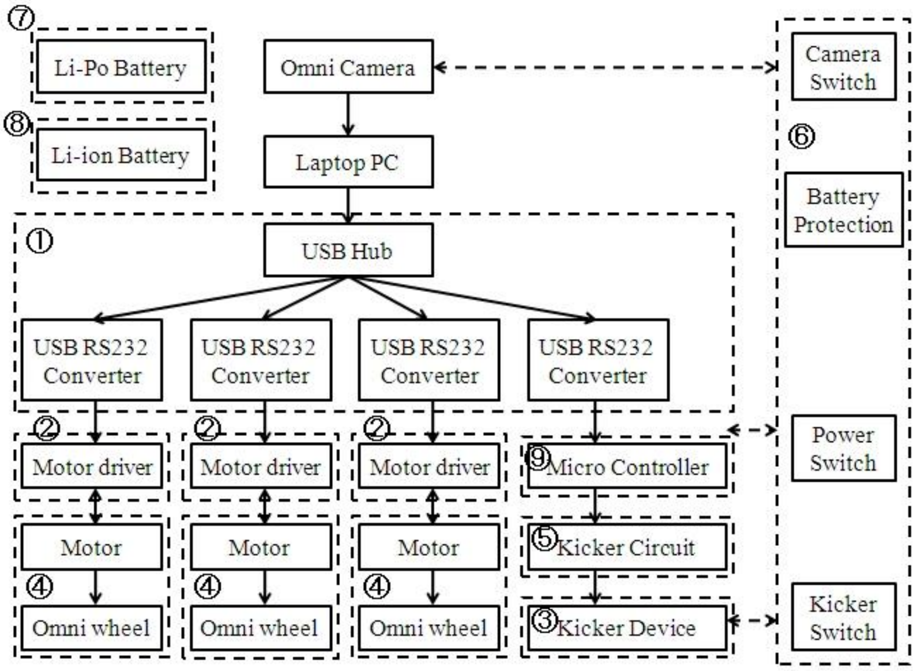

Fig. 3 Flowchart of the Musashi architecture (Each dash square area describes basic modules. Area 1 indicates the USB module, area 2 the motor driver (MD) module, area 3 the kicking device (KD) module, area 4 the motor and wheel (MW) module, area 5 the kicker circuit (KC) module, area 6 the main, power and camera switches (SW) module, area 7 the Li-Po battery module, area 8 the Li-ion battery module and area9 the micro controller (MC) module).

Step 2: Definition of basic module

Based on the flowchart of the robot architecture (Fig. 3), the basic modules should be determined by considering similar hardware structures or similar mechanical connections. Fig. 3 shows the seven basic modules for Musashi hardware illustrated with dash square area and number. For example, a USB hub and four USB/RS232 converters can be a module (USB module) because they have a common interface. Another consideration is the mechanical similarity such as an omni-wheel and a motor (MW modules). It is important to note that one of the considerations in the design of a basic module is that a mechanical interface should be able to attach the basic modules to the back plane "directly". The mechanical interface including the connector and the fixture should be designed to accept external force by the fixture, not to occur wrong installation to the back plane, and to be easy to exchange the modules.

Another consideration for definition of the basic module, we consider the safety. For example, battery module is defined by considering safety. Our team's battery is Li-polymer battery, therefore, we should consider explosion. The battery module consists of top and bottom parts. Both parts is covered a battery and free wires. Therefore, the dangerous area of the battery is hidden almost 60 % for the module parts. Also, the battery module has the battery box, so that means the dangerous area of the battery is reduced to less than 20 % by the use of battery module.

As mentioned above, the modularity concept and safety concept has a good relationship for making the reliable robot.

Step 3: Definition of the merged module

The concept of modularity can be extended by merging the basic modules in an effort to decrease the number of wires. A merged module can be theorized by considering the flow chart connections of the single modules and realized by design and implementation of the "back plane" concept. A back plane can be regarded as a basic module's communication port in a merged module. For example, considering the connections among the USB, kicker circuit (KC) and motor driver (MD) modules, a back plane can be designed to merge the five modules (a USB, a KC, and three MD modules) and solve the problem of complex wiring connections. The new merged module is referred to as a central control module (Fig. 4).

The Musashi modular architecture is summarized as follows. We designed the robot as comprising two main modules, a bottom module and an upper module (Fig. 5), considering the ease of assembling, maintenance, troubleshooting, and transportation. Consequently, the bottom module consists of six single modules (a switch, a battery, a kicking device, and three motor-wheel modules) and one merged module (the central control module). Using this approach, we could solve problems (e) and (f) described in the second section.

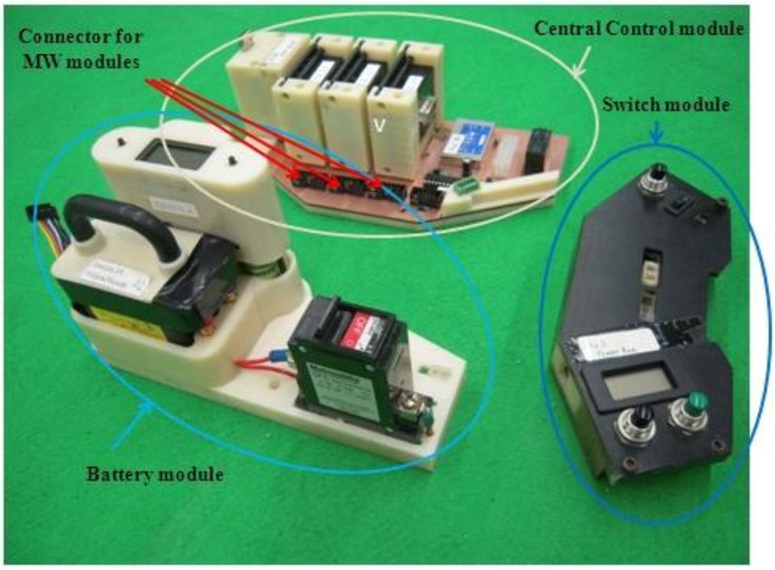

Fig.4 Modules of Musashi hardware (Battery and SW modules can be recognized as basic modules, and Central Control module as an merged module.).

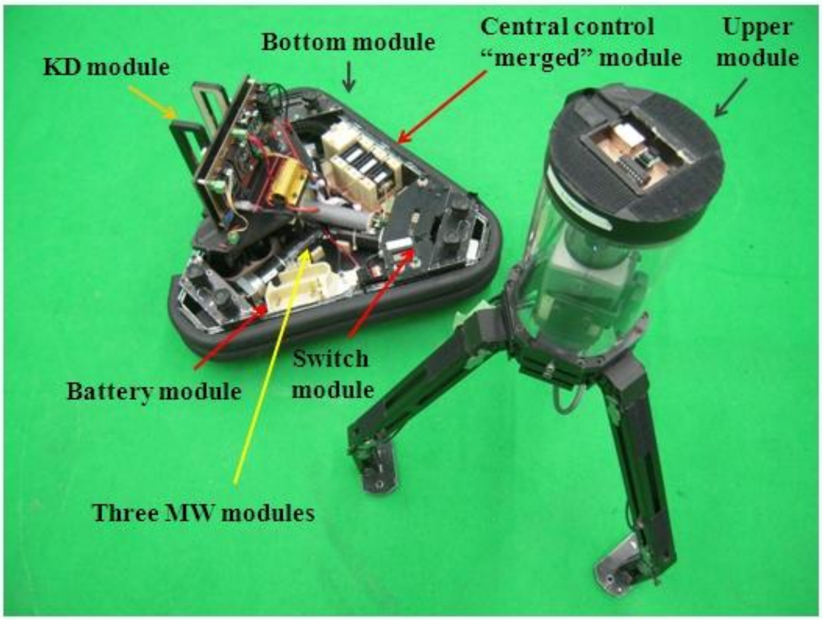

Fig.5 Main merged modules of Musashi: bottom module and top module.

1.3 Development of the electromagnetic kicking Device

The previous kicking device was spring based type. The key idea was to charge a series of strong torsion spring by using a special design of a cam in shape to shoot and lift the ball. The disadvantage of our spring type kicking device was to not able to change the shooting power, then we could not use the kicker for passing function specially in set-player. In this direction, an electromagnetic kicking device was developed in order to increase the team strategy variation using pass function.

The developed kicking device was designed and constructed specifically for "Musashi" robot (Fig.2). The kicker is based on an Induction-Coil-Gun Approach and consists of two interacting parts, the coil and the plunger. The coil is wound over a nonconducting tube with a central longitudinal whole. The tube is constructed from PVC and has 2 [mm] thickness. Inside the tube lies the movable rode which is made out of a ferromagnetic material. By sending a current through the coil, a magnetic field emerges which magnetizes the rod in the same sense as the coil. So the side of the rod, facing the rod, sees an opposing pole and the rod shoots out of the tube. This effect (of the two separate magnets) is the attraction of the rod inside the coil as a shooting mechanism. The rod is constructed out of two parts: First, the front part, which is constructed out of MC Nylon and which is in the deactivated state inside the coil and second the rear part, which is constructed from a ferromagnetic material like steel. As the metal part moves inside the tube, the front part pushes against a lever arm. To control the kicker functionality a Peripheral Interface Controller (PIC) based module is developed.

The DC/DC converter is a typical element which can convert 24V DC to 90V DC to provide power for the 90V solenoid. The kicking device is equipped by four 22000 [μF] capacitors. Total 88000[μF] is used as a buffer that can be charged from the sub battery. The capacitors will be charged automatically when the kicker switch turns on. According to the team strategy, PC installed on the robot can send the different signal, "kick signal" or "pass signal" to PIC (PIC18F1320) installed on kicker controller board. The kicking device control depends on the amount of current through the coil. The kicking device power is controlled based on the amount of current which can through the coil. The coil current is controlled by five MOS-FETs which are functioned as a switch between the capacitors and the solenoid. In instance, If the kick signal send to the PIC, the current is thrown to the coil for 10 [msec], and the robot will kick the ball with speed of 7 [m/s]. And in case of the short pass signal, the current is thrown to the coil for 5 [msec] resulting in passing the ball for set-player. After kicking or passing, capacitor is charged for 4 seconds automatically. The kicking device will become on standby for next time.

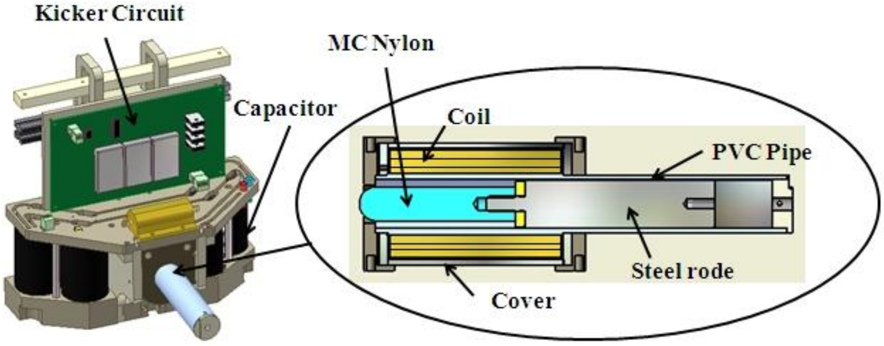

Fig.6 Overview of the developed Electromagnetic Kicking Device

2 Software System

2.1 Musashi Robot Software Flow Chart

Musashi robot software is divided into three parts: image processing, communications, and behavior. The image processing part receives an image from IEEE1394 camera and extracts necessary information (the distance and the angle to the object); such as ball position, obstacles position and self position. Position of the robot is estimated by self localization method based on the Monte Carlo Localization (MCL) algorithm and dead reckoning. The communications part communicates the signal from a referee box and data between robots as multipoint-to-multipoint. From these two processes, robots choice behavior and run their behavior. Fig. 8 indicates the Musashi robot software flow chart.

2.2 Self Localization using MCL and dead reckoning

The goals had no color because the rules of middle size league were changed in 2008. Therefore, we couldn't use the traditional way, using the color of goals as landmark, for calculating the robot position. Our self localization method is based on Monte Carlo Localization (MCL) using the information of field lines and information from odometry.

2.2.1 Line detection

Our vision system uses omni-directional images in the YUV color space. We extract white and green from these images to avoid detecting white objects existing out of the field. To detect field lines, we scan the image using multi-layered scan lines arranged in radial direction and search the crossing points between the field lines and the scan lines.

2.2.2 Robot Self-Localization

We use the MCL method which is one kind of particle filters for robot self localization. This method is used widely for mobile robot localization since it has good real-time performance and robustness. The field model is a Cartesian coordinate system with the origin at the center of the field in our algorithm. The robot state is represented by a vector $\mathbf{x}_t$ which consists of position (x, y) and direction θ . We calculate the posterior probability distribution $p(\mathbf{x}_t \mid \mathbf{y}_1 ... \mathbf{y}_t)$ from the state of a robot $\mathbf{x}_t$ and the sensor data at $\mathbf{y}_t$ the current time t. In the particle filter, a probability distribution is represented by a set of N random samples. This method proceeds in two phases.

Prediction Phase: In the first phase we predict a current state of the robot. This is specified as a conditional distribution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ from the previous state $\mathbf{x}_{t-1}$ and a control input $\mathbf{u}_{t-1}$. The predictive distribution is obtained by following equation.

$$p(\mathbf{x}_t \mid \mathbf{y}_1 .. \mathbf{y}_{t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{y}_1 .. \mathbf{y}_{t-1}) d\mathbf{x}_{t-1} \quad (1)$$

Where $\mathbf{u}_{t-1}$ is the odometry data and it added to each particle.

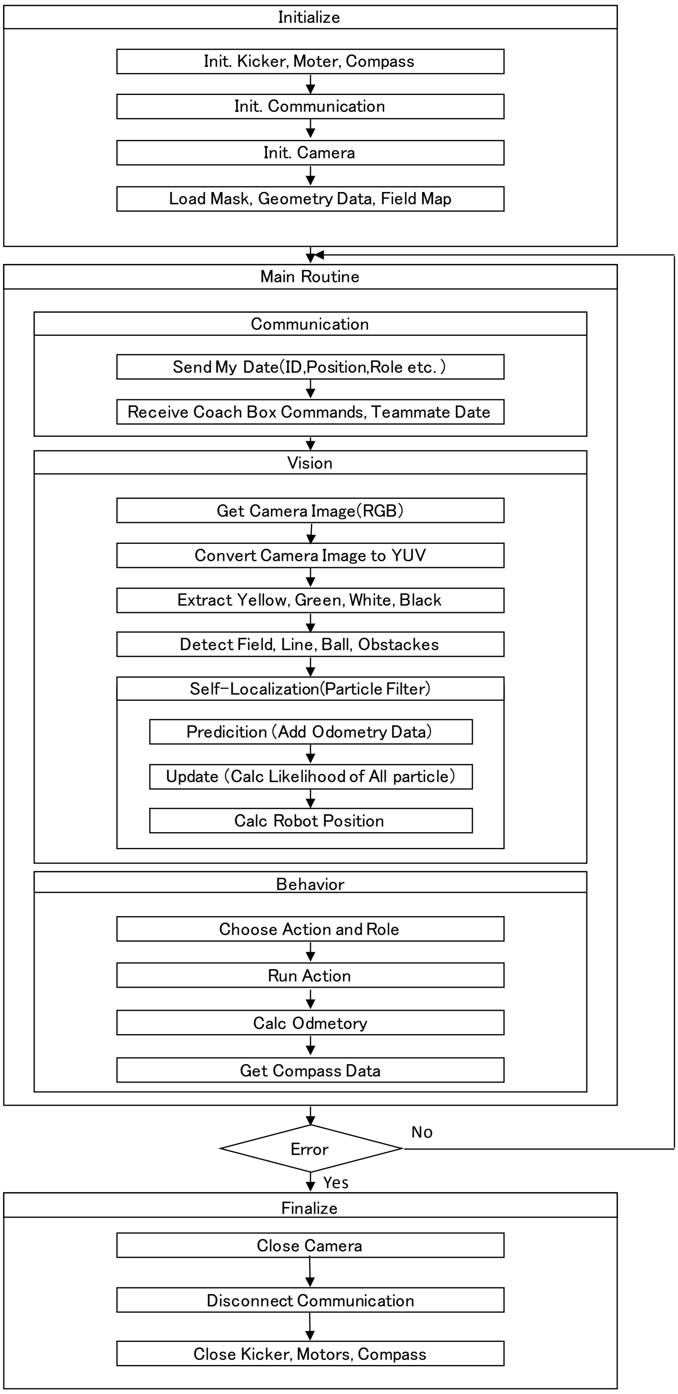

Fig.7 Flow chart of the Musashi robot software

Update Phase: In the second phase we update the distribution $p(\mathbf{x}_t \mid \mathbf{y}_1 \ldots \mathbf{y}_t)$ according to the sensor data. The likelihood of $\mathbf{y}_t$ at state $\mathbf{x}_t$ is represented as $p(\mathbf{y}_t \mid \mathbf{x}_t)$. Where $p(\mathbf{y}_t \mid \mathbf{x}_t)$ is determined by normal distribution. The frequency function of normal distribution used to decide $p(\mathbf{y}_t \mid \mathbf{x}_t)$.
Where mean μ is 0, standard deviation σ is 0.01, variance σ^2 is 0.1, and

$$x = \log(1.0 + D_i) - \log(1.0 + D_r) \quad (2)$$

Where D_i and D_r are ideal value and real value of distance between the robot and line, respectively. The posterior distribution is obtained using Bayes theorem.

$$p(\mathbf{x}_t \mid \mathbf{y}_1 \ldots \mathbf{y}_t) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid \mathbf{y}_1 \ldots \mathbf{y}_{t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_1 \ldots \mathbf{y}_{t-1})} \quad (3)$$

Where $\mathbf{y}_t$ is distance to the field line. After updating the likelihood of all particles, they are normalized and re-sampled. Re-sampling proceed according to the weight of each particle: new particles are generated around the particles that have high likelihood.

Our task is to use MCL in the RoboCup environment. There is possibility of it fails in detection of direction because of the symmetric shape of the field. We solved this problem by using a direction sensor.

References

[1] A.A.F.Nassiraei, “Concept of Intelligent Mechanical Design for Autonomous Mobile Robots”, Kyushu Institute of Tech., Ph.D thesis, 2007

[2] A.A.F. Nassiraei, Y.Takemura, et. al. “Concept of Mechatronics Modular Design for an Autonomous Mobile Soccer robot”, CIRA 2007, Jacksonville, pp.178~183

[3] A.A.F. Nassiraei, K Ishii, “How Does “Intelligent Mechanical Design Concept” Help Us to Enhance Robot’s Function?,” Intelligent Unmanned Systems: Theory and Applications, Volume 192, pp. 155-178, Springer 2009.

[4] RoboCup Middle Size Robot League Rules and Regulations for 2010 : http://robocupmsl.googlegroups.com/web/msl-rules-2010-05-12.pdf