Overview of Hibikino-Musashi Hardware and Software

Shota Chikushi[1],
Masafumi Kuwada[1], Makito Ishikura[1], Takahiro Nagao[1], Ryo Itohara[2],
Shuta Watanabe[1], Kenji Hisano[1], Ryota Shimada[1], Kohei Matsumoto[1],
Moeko Tominaga[1], Naoto Machida[1],
Kazuzo Ishii[1], Hiroyuki Miyamoto[1], Takashi Sonoda[1], Yasunori Takemura[3]

[1] Kyushu Institute of Technology, Japan
[2] WASEDA University, Japan
[3] Nishinippon Institute of Technology, Japan
ishii@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 150 Robot Architecture and Specification

The current hardware configuration of the "Musashi 150" robot (Figure 1) and its fully modular mechatronics architecture including an Omni-directional moving mechanism and an Omni-vision system is shown in Figure 2. 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 150 watts DC motor from Maxon, arranged in the shape of triangle. Specification is shown in Table 1.

The maximum nominal motor speed of 7580 rpm is decelerated through a planetary gearbox GP42 with the ratio of 6:1. In addition, decelerated through belt and pulley with the ratio of 2:1. The amplified mechanical torque on the output pulley is transferred to the wheel's shaft through supported by a pair of the radial ball bearings. The velocity feedback is done by using 2000 pulses digital incremental encoders. The velocity of the wheels is controlled by three EPOS motor drivers from Maxon. Each equipment is connected Controller Area Network. 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 "Musashi 150" with maximum linear speed of 3.5m/s and acceleration of 2.1m/s^2.

The sensors using in the "Musashi 150" are an Omni-directional camera, a compass and three DC motor encoders. The electrical power is supplied by a set of Nickel-metal

hydride batteries (nominal voltage 24V/2.8Ah). The necessary voltage for the camera, compass module and the microcomputer are produced by converting 24V to 12.0V and 5.0V (Figure 3). In order to realize the shooting function, an electromagnetic kicker, designed and constructed specifically for “Musashi 150”. The kicker is based on an Induction-Coil-Gun Approach and consists of two interacting parts, the coil and the rod. This Robot is mounted “Active-Finger” use small wheel to control ball.

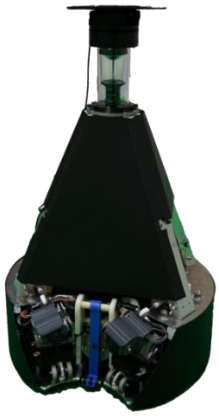

Figure 1. “Musashi 150” robot

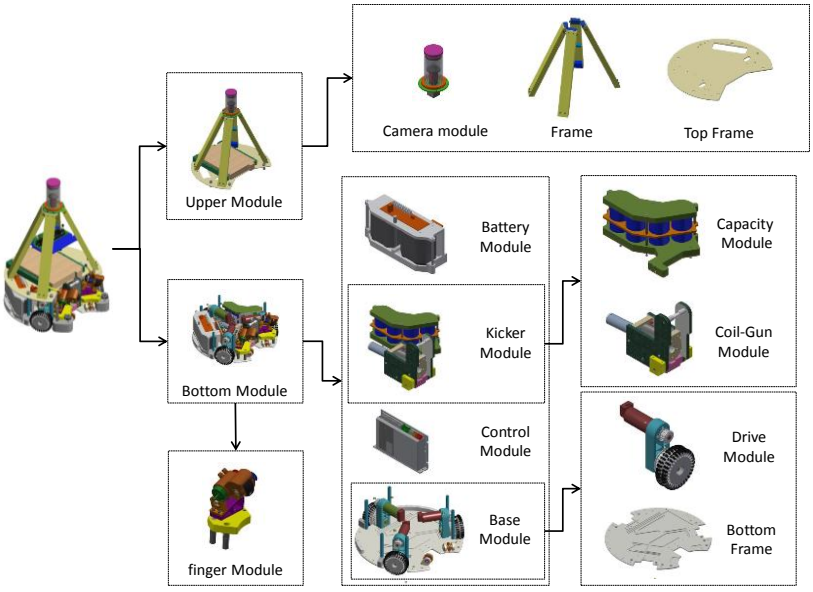

Figure 2. “Musashi 150” robot hardware configuration and modular architecture

Table 1. Specification of "Musashi 150" robot

Size	500×500×800[mm]
Weight	30.0[kg]
Actuator	DC-Motor×3(24[V],150[W]) DC-Motor×2(24[V],50[W])
Battery	Ni-MH Battery × 2 for motioning
Kicking Device	Solenoid coil
Shoot Speed	1.5～5.0[m/s]
Maximum acceleration	2.1[m/s^2]
Maximum Speed	3.5[m/s]
Sensor	Omni-directional camera ×1 Electrical compass ×1 DC-Motor Encoder ×5
Motor Driver	EPOS2 70/10 × 3 EPOS2 50/5 ×2

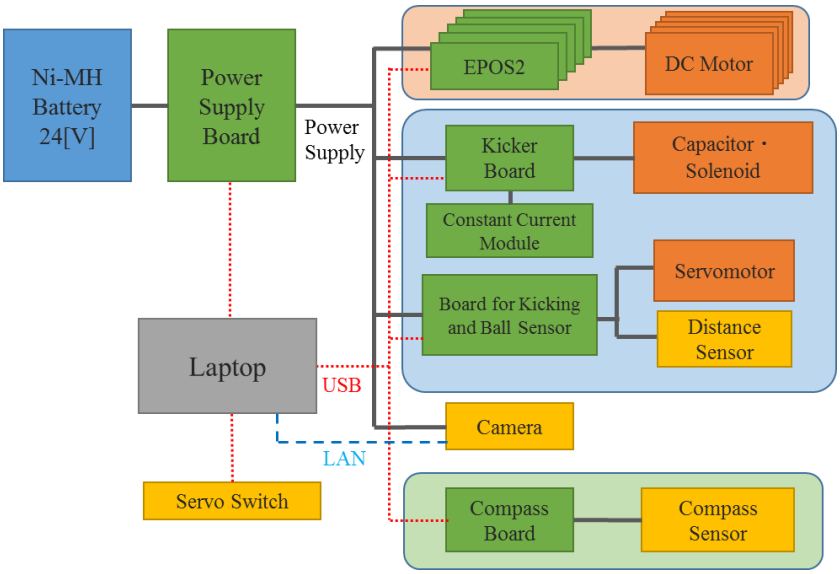

Figure 3. Flowchart of "Musashi 150" robot power system

1.2 The Development of New Solenoid Kicker[1]

For the kicking power improvement, we have developed a new solenoid kicker. The existing solenoid kicker can only give a shooting speed of 5[m/s] [2]. Therefore, we have developed a new solenoid kicker, and succeeded in improvement of shoot speed and distance of the ball. In order to convert, we set the dimensions of the existing solenoid kicker to the maximum dimensions.

In developing, we examined the four items: the diameter of the windings, the inner diameter of the coil, the outer diameter of the coil and the length of the coil. The results are shown in the Table 2. The diameter of the windings was analyzed by using a simulator. The results are shown in the Figure 4. As a result of the analysis, ϕ2[mm] was found to be the best.

We made the trial products of the solenoid kicker, and confirmed the performance. The details of the new solenoids are shown in the Table 3. We compared the existing solenoid kicker and new solenoid kicker for shoot speed and distance of the ball. The results are shown in the table. Next, we confirmed variation of distance of the ball by initial position of the plunger using new solenoid kickers. The results are shown in the Figure 5.

The new system was improved the performance, and was success on the size down. Shoot speed of the ball was 15[%] improvement. Distance of the ball was 38[%] improvement.(Table 4)

Table 2. Resalt of the analysis of the items of the solenoids

the inner diameter of the coil [mm]	**the outer diameter of the coil [mm]**	**the length of the coil [mm]**
27.0	60.5	102.5

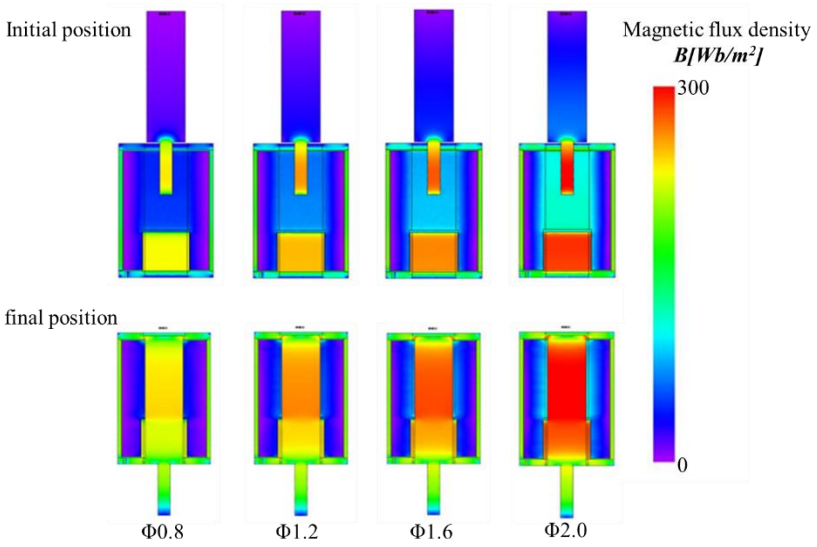

Figure 4. Analysis result of solenoid kicker (TAKAHA KIKO Co., Ltd.)

Table 3. Details of the new solenoids

New solenoid	Wire diameter [mm]	Turns	Resistance [Ω]
A	1.2	670	1.282
B	1.6	400	0.461
C	2.0	234	0.193

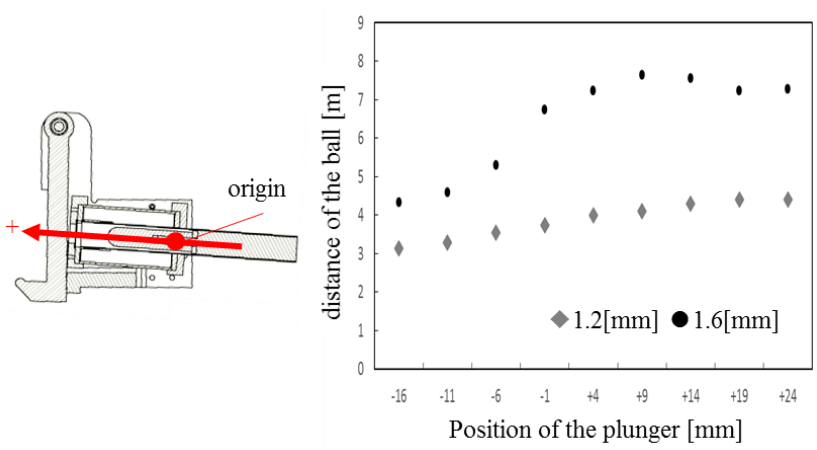

Figure 5. Relationship between the position of the plunger and distance of the ball

Table 4. Performance comparison

	Distance of the ball [mm]	Velocity [m/s]
The existing solenoid kicker	5.48	5.48
The new solenoid kicker	7.57	6.33

1.3 Active Ball Handling Mechanism[1]

Our old player robot "Musashi" that has been developed in 2005 has a mechanism for rotate the cam to closing the arms as the Ball Handling Mechanism. However, there are some problems that when robots do a sudden stop, rotate and back, they lose the ball. Furthermore, a rule has been adopted that robots must pass between the allies during the in play or some of the set plays from 2012. Based on these rules, a high cooperation action and ability for ball handling has been more necessary than before. Therefore, we implemented the new Ball Handling Mechanism for improvement of the ball handling ability with the new machine. Diagrammatical view of the Ball Handling Mechanism are shown in the Figure 6, 7. When it starts the handling of the ball, turns a lever-wheel clockwise and draws a ball to the inside of robot. The rotary speed of the wheel is calculated by a moving direction and the speed of the robot, and it allows to hold a ball while doing a natural turn. In addition, it was attached Omni wheel to the lower front of Ball Handling Mechanism, and it can support the ball, and does not hinder the turn of the ball when robots dribble. Specification of the Ball Handling Mechanism is shown in Table 5.

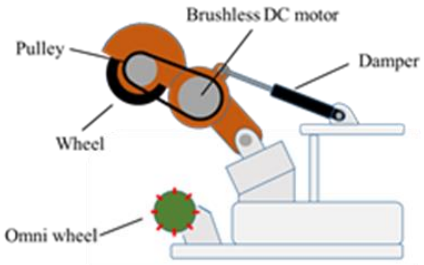

Figure 6. Side view of Ball Handling Mechanism

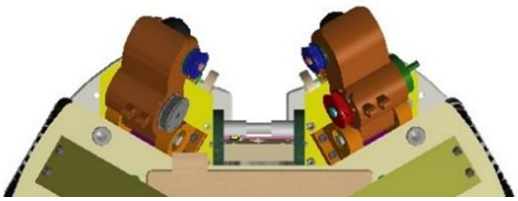

Figure 7. Top view of Ball Handling Mechanism

Table 5. Specifications of Ball Handling Mechanism

Voltage	24[V]
Maximum Power	50[W]
Maximum Motor Speed	13100[rpm]
Maximum Motor Torque	48.2[mNm]
Wheel Size	60[mm]
Reduction Rate	10.56

2 Software system

2.1 Musashi 150 Robot Software Flow Chart

Musashi 150 robot software is divided into three parts: image processing, communications, and behavior[3]. The image processing part receives an image from 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. Figure 8 indicates the Musashi 150 robot software flow chart.

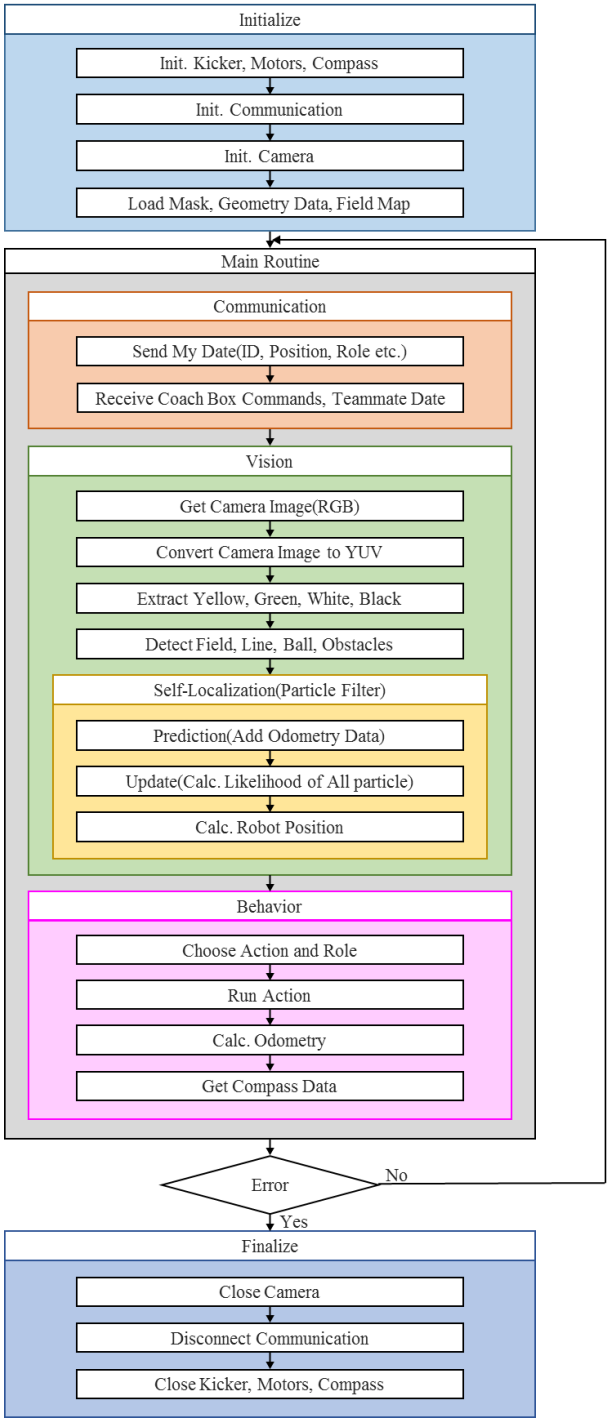

Figure 8. Flow chart of the Musashi 150 robot software

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 $p(\mathbf{x}_t|\mathbf{y}_1 \dots \mathbf{y}_t)$ 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 $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ from the previous state $\mathbf{x}_{t-1}$ and a control input $\mathbf{u}_{t-1}$. The predictive distribution is obtained by following equation.

$$p(\mathbf{x}_t|\mathbf{y}_1 \dots \mathbf{y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1})\, p(\mathbf{x}_t|\mathbf{y}_1 \dots \mathbf{y}_{t-1}) d\mathbf{x}_{t-1} \qquad (1)$$

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

Update Phase: In the second phase we update the distribution $p(\mathbf{x}_t|\mathbf{y}_1 \dots \mathbf{y}_t)$ according to the sensor data. The likelihood of $\mathbf{y}_t$ at state $\mathbf{x}_t$ is represented as $p(\mathbf{y}_t|\mathbf{x}_t)$. Where $p(\mathbf{y}_t|\mathbf{x}_t)$ is determined by normal distribution. The frequency function of normal distribution used to decide $p(\mathbf{y}_t|\mathbf{x}_t)$.

Where mean μ is 0, standard deviation σ is 0.01, variance σ^2 is 0.1, and

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

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.

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

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. Shota Chikushi *et al.*, "Hibikino-Musashi Team Description Paper", *RoboCup*, 2016.
2. Amir A.F. Nassiraei *et al.*, "Hibikino-Musashi Team Description Paper", *RoboCup*, 2011.
3. Amir A.F. Nassiraei *et al.*, "Overview of Hibikino-Musashi Hardware and Software", *RoboCup*, 2010.