Chapter 3

Mechatronics Educational System Using Multiple Mobile Robots with Behavior-Based Control Approach

A unique education system is proposed for students pursuing mechanical engineering to learn basic mechatronics skills and techniques practically. The system is composed of three subsystems. The first subsystem is used for second year students to learn mainly input/output port operations, periodically switching LED lights ON/OFF and a stepper motor control. The second subsystem is effective for third year students to learn AD conversion for several sensory information systems, DA conversion for a DC motor control and a design of a discrete-time PI controller. Further, the third subsystem is a system of multiple mobile robots – part of the graduation studies so that fourth year students can learn the concept of subsumption control architecture for schooling behavior.

Chapter written by Fusaomi NAGATA, Keigo WATANABE and Maki K. HABIB.

Mechanical Engineering Education Edited by J. Paulo Davim

3.1. Introduction

Up to now, many studies on educational systems have been conducted. In this chapter, a unique mechatronics educational system is proposed for mechanical engineers, to enable them to efficiently learn basic skills and techniques of mechatronics. For this purpose, the C++ of Microsoft Visual Studio is recommended as a software development environment, because it is one of the languages with professional technical skills demanded by different industries. The developed educational system is composed of three subsystems. The first subsystem is used for second year students to learn input or output port operations using hexadecimal code, periodically switching light emitting diode (LED) lights ON or OFF by using a Windows timer, such as a sampling period of 10 ms and stepper motor control built in a position control module and a mobile robot with two wheels. The second subsystem is used for third year students to learn analog digital (AD) conversion supporting different analog sensory information, digital analog (DA) conversion for direct current (DC) motor control and proportional-integral-derivative (PID) control method. Further, the third subsystem consists of a system with multiple mobile robots supporting the final year graduation thesis study of fourth year students to learn the concept of subsumption architecture for swarm intelligence, such as schooling behavior. The effectiveness of the proposed systems is evaluated through experimental instructions at Tokyo University of Science, Yamaguchi, Japan.

3.2. Mechatronics education subsystem I

3.2.1. *Hardware of mechatronics educational subsystem I*

Figure 3.1 shows the first mechatronics system consisting of an LED display board, a position control device and a mobile robot with two wheels [KIT 11, KIT 12]. The system is used in the experimental lecture of “Mechatronics Experiment I” for second year students. Figure 3.2 shows the block diagram of the LED display experiment. Sixteen LEDs are arrayed with two rows as shown in Figure 3.3 and each row is related to output port B and C, respectively. In Figure 3.3, hexadecimal number 0X55 and 0XAA are set to port B and port C, respectively. A lighting pattern can be designed by setting the necessary hexadecimal code number; also, a lighting period can be changed by using a Windows timer interrupt associated with a sample rate, such as, 10 ms.

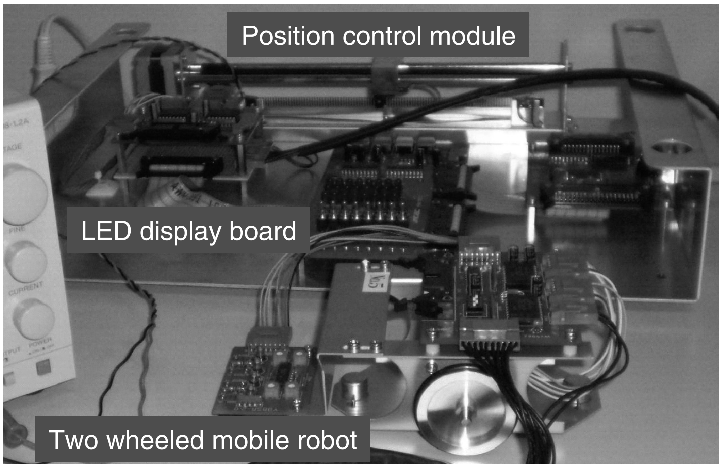

Figure 3.1. *Mechatronics subsystem consists of LED display board, a position control device, and a mobile robot with two wheels, which is used in the experimental lecture of “Mechatronics Experiment I”*

Figure 3.2. *Block diagram of LED display experiment using universal serial bus (USB) interface*

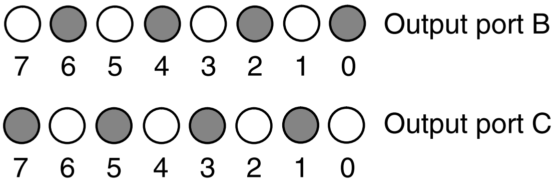

Figure 3.3. *Relation between two-arrayed LEDs and output ports B and C*

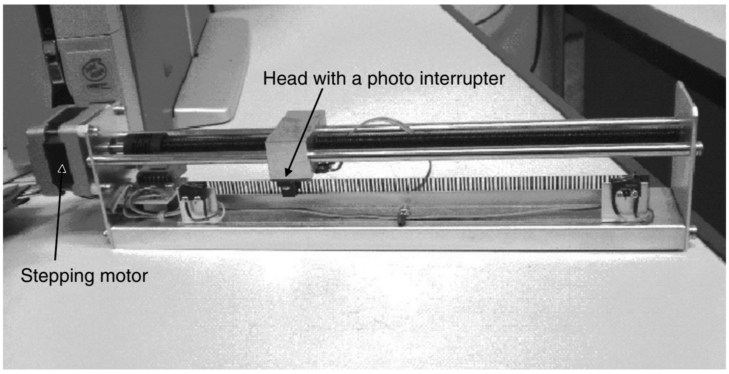

Figure 3.4. *Position control device with a photo interrupter driven by a stepper motor*

Figure 3.4 shows the position control device driven by a stepper motor. A moving head is integrated with a ball screw mechanism. A photo interrupter is attached to the moving head to detect the position of the slit while moving. The stepper motor is driven by pulse signals. One pulse signal rotates the stepper motor 1.8 degrees, and this moves the moving head by 6.25 μm through the ball screw mechanism. The pulse signal is generated by the lower four bits of output port B. The relation between the binary code number sent to port B and the type of motion is listed as follows:

1) 00000110 (0X06) Low-level code for clockwise rotation (CWL).

2) 00000111 (0X07) High-level code for clockwise rotation (CWH).

3) 00000010 (0X02) Low-level code for counterclockwise rotation (CCWL).

4) 00000011 (0X03) High-level code for counterclockwise rotation (CCWH).

5) 00001000 (0X08) Motor excitation off.

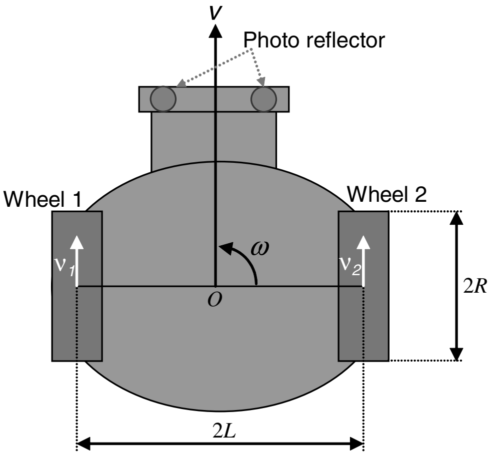

Figure 3.5. *Top view of two-wheeled mobile robot with two photo reflectors*

Figure 3.5 illustrates the mobile robot with two wheels and two photo reflectors. Wheels 1 and 2 are driven by stepper motors 1 and 2, respectively. The stepper motors 1 and 2 are excited by the lower four bits of output ports B and C, respectively. In Figure 3.5, v and ω are the translational velocity and the rotational velocity of the robot in the robot coordinate system; $2L$ is the distance between the two wheels and R is the radius of each wheel. The robot can perform forward and backward motions or rotational motion. When forward and backward motions are applied, the velocity of each wheel, v_1, v_2, is set as

$$v = v_1 = \omega_1 R = v_2 = -\omega_2 R \qquad [3.1]$$

Also, when the rotational motion is conducted, the rotational velocity ω of the robot is given by

$$\omega = \frac{\omega_1 R}{L} = \frac{\omega_2 R}{L} \qquad [3.2]$$

where ω_1 and ω_2 ($\omega_1 = \omega_2$) are the rotational velocities of two wheels, respectively.

3.2.2. *Basic dialog for students' experiment*

This section introduces the basic interactive window dialog developed for second year students' Mechatronics Experiment I. Students can effectively design a control program by using a dialog panel and learn basic mechatronical peripheral techniques. The Mechatronics Experiment I is considered for second year students at the department of mechanical engineering, which is composed of three topics, i.e. periodically switching LED lights ON or OFF experiments, a positioning experiment for position control and line tracing experiments of a two-wheeled mobile robot. As an example, Figure 3.6 shows the developed interactive window dialog designed for the line tracing experiment using the mobile robot stated in Mechatronics Experiment I. For example, if the "CW rotation of Motor 1's" button is clicked, then the stepper motor 1 rotates 1.8 degrees in the clockwise direction. Students can not only confirm the motion but also see the corresponding program codes in another window.

At first, the students learn how to program important basic functions such as moving forward or backward, turning left or right and sensing two photo reflectors through the dialog developed by using Microsoft Visual C#. Then, students can make a program for a line trace by making use of the basic functions. Fifteen hours (five hours, three days) are allotted to the Mechatronics Experiment I at Tokyo University of Science, Yamaguchi, Japan.

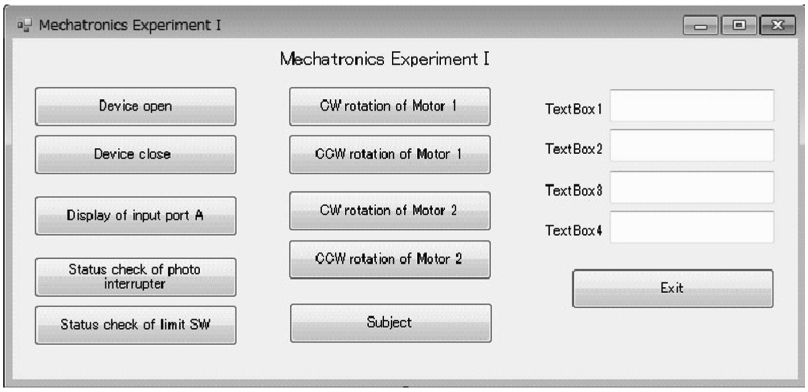

Figure 3.6. *Windows dialog designed for Mechatronics Experiment I*

3.3. Mechatronics educational subsystem II

3.3.1. *Hardware of mechatronics educational subsystem II*

Figure 3.7 shows the hardware of the second mechatronics system consisting of three sensors for measuring temperature, distance and brightness, AD/DA conversion interfaces, a liquid crystal display (LCD) display panel and a DC motor with a sensor of photo interrupter, which is provided by C-TASK Co., Ltd. The system is used in the experimental lecture of “Mechatronics Experiment II” for third year students [KIT 11, KIT 12].

Here, the basic software developed for velocity control of the DC motor is explained. A ladder-type DA converter is connected to output port 1, so that the voltage within the range from 0 to 3.3 V can have an output with the resolution of 256 steps. The velocity control of the DC motor can be performed by changing the voltage. The velocity of the motor is controlled by a simple proportional-integral (PI) action in the discrete time k given by

$$\tau(k+1) = \tau(k) + \Delta\tau(k) \qquad [3.3]$$

$$\Delta\tau(k) = K_p\{v_d - v_s(k)\} + K_i\sum_{n=1}^{k}\{v_d - v_s(n)\} \qquad [3.4]$$

where $\tau(k)$ is the output torque to the DC motor at the discrete time k; n is the incremental counter to make the sum of error for integral action; K_p and K_i are the P-gain and I-gain, respectively; v_d is the reference value of rotational velocity; and $v_s(k)$ is the rotational velocity measured by using a photo interrupter shown in Figure 3.8. A rotational wheel with a small hole is fixed to the shaft of the DC motor. When the hole passes through the photo interrupter, the output signal of the photo interrupter is going through transition from high level to low level, and this signal is a pulse that can be detected and counted. The rotational velocity is defined as the counted number of pulses within a constant time period (sample time).

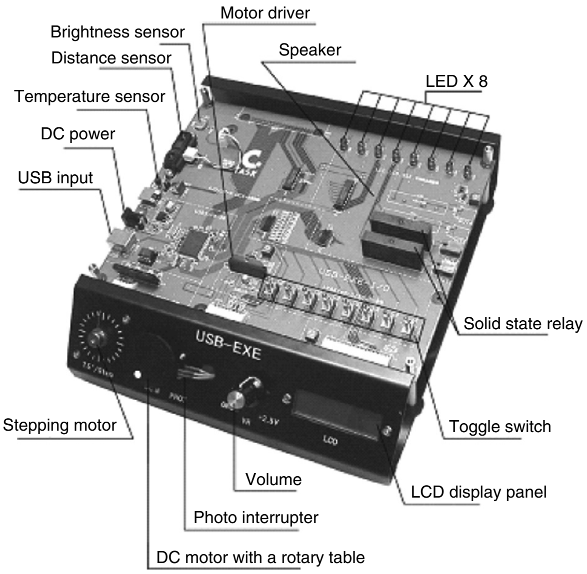

Figure 3.7. *Mechatronics experiment system consisting of three sensors for measuring temperature, distance and brightness, AD/DA conversion interfaces, an LCD display panel and a DC motor with a photo interrupter*

3.3.2. *Basic dialog for students' experiment*

A basic interactive window dialog was designed to support Mechatronics Experiments II. Students can efficiently make a control program by using the interactive window dialog and study basic mechatronical peripheral techniques concerning AD conversion of sensory information from analog sensors, pulse width modulation (PWM) control of a DC motor and PI control of rotational velocity. Mechatronics Experiment II is considered for third year students, which is mainly composed of three topics, i.e. AD conversion of several analog signals from sensors, such as distance, temperature and brightness, and DA conversion for velocity control of a DC motor. Students try to make a program for a discrete-time PI controller using equations [3.3] and [3.4] for controlling the velocity of the DC motor.

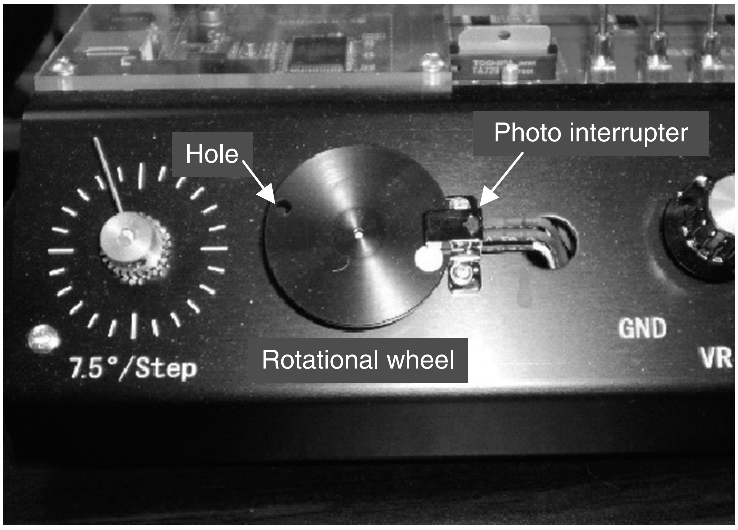

Figure 3.8. *Rotational wheel fixed to a DC motor with a sensor of photo interrupter*

Figure 3.9 shows the interactive window dialog designed for the purpose of the Mechatronics Experiment II. For example, if the “Velocity control of the DC motor” button is clicked, then the DC motor rotates following the velocity set by “Reference” in the dialog. Students can not only confirm the motion of the DC motor but also see the corresponding control program codes in another window. Ten hours (five hours, two days) are allotted to the set of experiments under Mechatronics Experiment II.

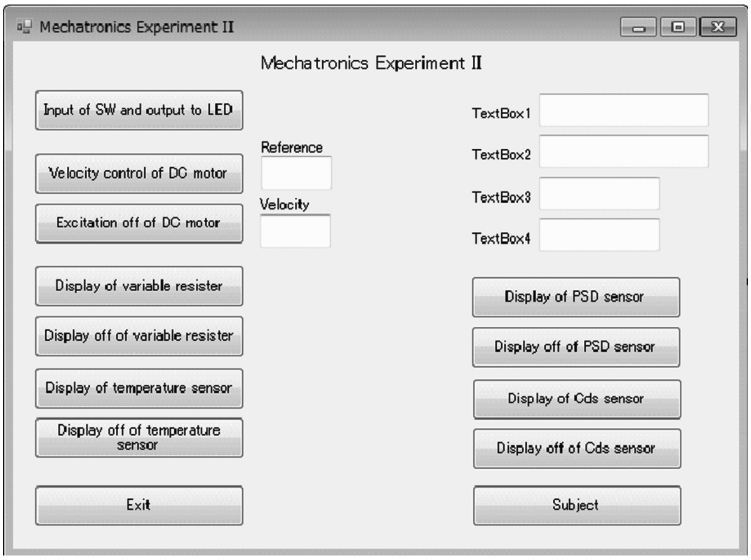

Figure 3.9. *Windows dialog designed for Mechatronics Experiment II*

3.4. Mechatronics educational subsystem III

This section includes a multiple mobile robots system integrated with subsumption control architecture to support final year study.

3.4.1. *Mobile robot with three wheels*

Students, who have gone through the lectures of Mechatronics Experiments I and II can easily cope with this part to support their final year study needs in their fourth year. This section of experiments includes multiple mobile robots; each is equipped with six distance sensors as shown in Figure 3.10. These robots are used to demonstrate the emergence of schooling behavior. The integrated distance sensors for each robot are called position sensitive detector (PSD) sensors, which can measure the distance to an object. Six PSD sensors are equally fixed, 60 degrees from each other, around the boundary of each mobile robot as shown in Figures 3.10 and 11 [YAM 10, NAG 11, YAM 11].

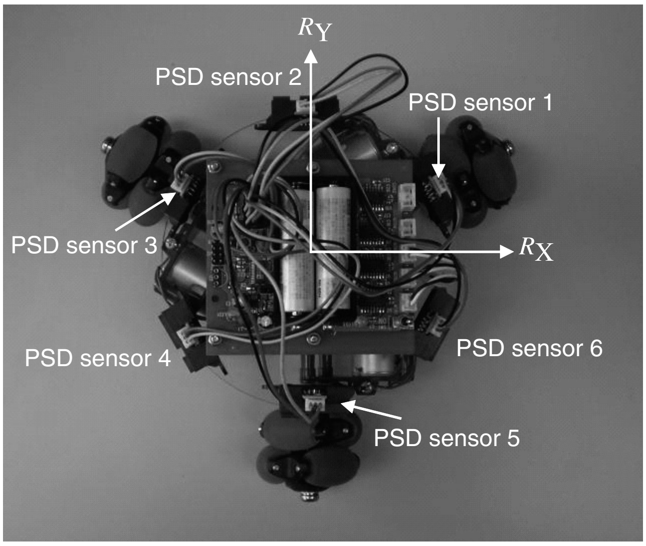

Figure 3.10. *Mobile robot with three wheels and six PSD sensors*

Next, the kinematic control method of the mobile robot is explained. Figure 3.12 illustrates the kinematic model of the robot. ω_i (i = 1, 2, 3) is the angular velocity of each wheel. Also, v_i (i = 1, 2, 3) is the velocity of each wheel and it is given by

$$v_i = r\omega_i (i = 1, 2, 3) \quad [3.5]$$

where r is the radius of each wheel. If the position and orientation vector of the robot, i.e. the origin in the robot coordinate system $\sum_R$ (O–^{R}X^{R}Y), is given by $\mathbf{x}_r = [x_r \ y_r \ \varphi_r]^T$, then the velocity can be represented by $\dot{\mathbf{x}}_r = [\dot{x}_r \ \dot{y}_r \ \dot{\varphi}_r]^T$. The following kinematic equations are obtained from Figure 3.12 [WAT 98, TAN 99].

$$v_1 = -\frac{1}{2}\dot{x}_r + \frac{\sqrt{3}}{2}\dot{y}_r + L\dot{\varphi}_r \qquad [3.6]$$

$$v_2 = -\frac{1}{2}\dot{x}_r - \frac{\sqrt{3}}{2}\dot{y}_r + L\dot{\varphi}_r \qquad [3.7]$$

$$v_3 = \dot{x}_r + L\dot{\varphi}_r \qquad [3.8]$$

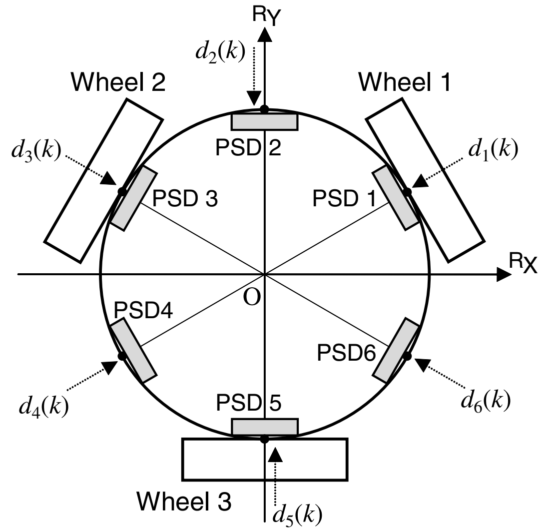

Figure 3.11. *Positions of six PSD sensors fixed around a mobile robot*

where L is the distance between the center of the robot and the center of each wheel. Equations [3.6] through to [3.8] lead to

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} & L \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & L \\ 1 & 0 & L \end{pmatrix} \begin{pmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\varphi}_r \end{pmatrix} \tag{3.9}$$

By using equation [3.9], the robot can be controlled kinematically. As special cases, Table 3.1 shows the velocity components to move to the direction of each PSD sensor. When designing the schooling mode using multiple mobile robots, the six basic velocities in Table 3.1 are used. The important point is that the direction of velocity, a mobile robot generates in $\sum_R$, depends on the ratio $\dot{x}_{ri} : \dot{y}_{ri}$. If the ratio is kept as tabulated in Table 3.1, the velocity norm may be changed voluntarily.

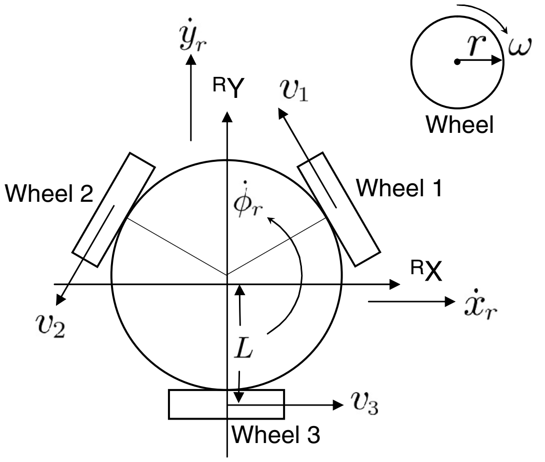

Figure 3.12. *Kinematics of the mobile robot with three wheels*

i	1	2	3	4	5	6
$\dot{x}_{ri}$	$\sqrt{3}$	0	$-\sqrt{3}$	$-\sqrt{3}$	0	$\sqrt{3}$
$\dot{y}_{ri}$	1	0	1	–1	–2	–1

Table 3.1. *Velocity components to move to the direction of each PSD sensor, in which $i(i = 1,\ldots,6)$ denotes the number of PSD sensors*

As mentioned above, the third Mechatronics educational subsystem is designed using multiple mobile robots, each having three wheels and six PSD sensors. This subsystem is used to support study needs of fourth year students, to learn the subsumption control architecture for schooling behavior. The subsumption control architecture is called behavior-based artificial intelligence, which was first proposed by Brooks [BRO 86]. Students can practically know the concept of the proposed subsumption control architecture for schooling behavior, which provides a method for structuring reactive systems from lower level to higher level, using layered sets of rules, i.e. reactive behaviors according to the change of a robot's environment. The details will be introduced in later sections.

3.4.2. *Network-based multiple mobile robots system*

A network-based multiple mobile robots system is proposed to design a high-level software architecture for multiple mobile robots using robot platforms with limited capabilities. Figure 3.13 shows the layout arrangement of the network-based multiple mobile robots system. Each robot can only transmit six examples of PSD sensory information $\boldsymbol{d}_i(k) = [d_{i1}(k) \ldots d_{i6}(k)]^T$ to the supervisory server through Bluetooth communication. The subscript i denotes the identification number of a robot. The supervisory server returns a set of a simple behavior and a short execution time, e.g. 200 ms to the corresponding robot. Eight kinds of simple reaction behaviors as tabulated in Table 3.2 are prepared for the mobile robots. When a set of a command code and an execution time is transmitted from the supervisory server to a mobile robot, the mobile robot conducts the motion exactly within the specified execution time. Three agents “Avoid objects,” “Turn to left or right” and “Move forward” to realize a schooling behavior are designed by using the basic instructions shown in Table 3.2.

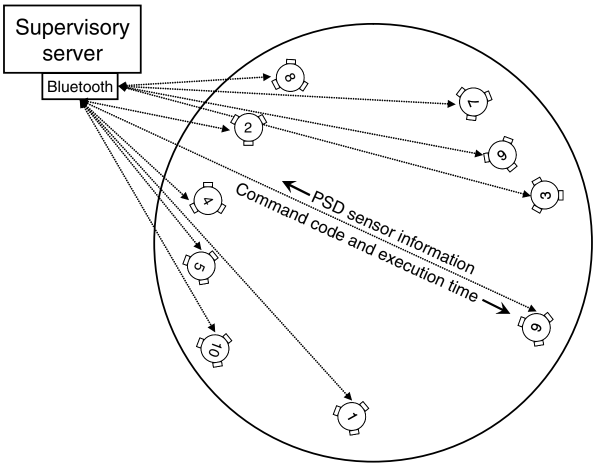

Figure 3.13. *Network-based multiple mobile robots system aiming at the emergence of a schooling behavior*

3.4.3. *Subsumption control architecture implemented on supervisory server*

The software development environment is C# of Windows Visual Studio, which is used to develop and implement high-level software architecture such as subsumption control architecture according to application requirements. Figure 3.14 shows an example of the subsumption control architecture implemented on the supervisory server. The controller includes three agents: “Avoid objects”, “Turn to left or right” and “Move forward” for the schooling behavior of multiple mobile robots. The first agent has higher priority in terms of dispatch. This section introduces the three types of agents and the corresponding output command codes shown in Table 3.2. The eight commands shown in Table 3.2 are the most simply subdivided actions for the mobile robot.

Cmd. code	Corresponding motion
1	Move to the direction of PSD sensor 1
2	Move to the direction of PSD sensor 2
3	Move to the direction of PSD sensor 3
4	Move to the direction of PSD sensor 4
5	Move to the direction of PSD sensor 5
6	Move to the direction of PSD sensor 6
7	Rotate to clockwise direction
8	Rotate to counterclockwise direction

Table 3.2. *Simple reaction behaviors for a mobile robot*

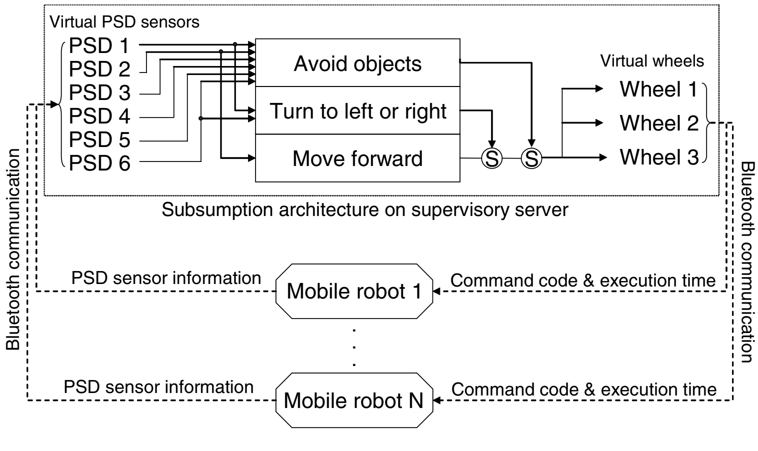

Figure 3.14. *Subsumption control architecture for schooling behavior, which is implemented on supervisory server personal computer (PC)*

The server receives PSD sensory information $d_i(k)$ ($1 \leq i \leq N$) from all mobile robots for every sampling period of 10 ms, in which N is the number of available mobile robots. By analyzing $d_i(k)$, the controller dispatches the current execution right to one of the three agents for the ith mobile robot. This process is periodically applied to all mobile robots in the order from 1 to N. In the schooling mode described in this section, all mobile robots try to regularly move along the inner area of a circular fence keeping a distance to both the fence and other mobile robots. This mode allows the robots to behave like carps in a Japanese artificial circular pond.

3.4.3.1. *Move forward*

When a robot moves counterclockwise along a circular fence as shown in Figure 3.13, the agent of "Move forward" is always active. Accordingly, if higher priority agents such as "Turn to left or right" and "Avoid objects" are inactive, then the agent "Move forward" can have the execution right and the following control law is applied.

$$\dot{\mathbf{x}}_i = v \frac{\dot{\mathbf{x}}_{i2}}{\left\| \dot{\mathbf{x}}_{i2} \right\|} \qquad [3.10]$$

where $\dot{\mathbf{x}}_i = \left[\dot{x}_i \ \dot{y}_i \right]^T$ is the virtual translational velocity for the ith mobile robot and $\dot{\mathbf{x}}_{i2} = \left[0 \ 1 \right]^T$ is the normalized velocity vector for moving to PSD sensor 2. The position of PSD sensor 2 is assumed to be the front of the mobile robot. v is the scalar signifying the magnitude of velocity. While the agent "Move forward" has the execution right, a set of a command code 2 and an execute time T, e.g. 200 ms, are continuously transmitted to the mobile robot within every specified execution time.

3.4.3.2. *Turn to left or right*

In this case, the orientation of a mobile robot is controlled by the agent "Turn to left or right". This agent becomes active when $d_{i1}(k) < d_{\mathrm{ref}}$ and $d_{i6}(k) < d_{\mathrm{ref}}$ are simultaneously satisfied. d_{ref} is called the active reference range. Further, if this agent has the execution right, then the following control law is applied.

$$\dot{\varphi}_i = K_\varphi \{d_{i6}(k) - d_{i1}(k)\} \qquad [3.11]$$

where $\dot{\varphi}_i$ is the virtual rotational velocity of ith mobile robot, $d_{i1}(k)$ and $d_{i6}(k)$ are the values of PSD sensors 1 and 6 transmitted from the ith mobile robot, respectively. As can be seen from Figure 3.11, PSD sensors 1 and 6 are selected for the orientation control moving in the counterclockwise direction. K_φ is the gain that can control the orientation of the robot to be parallel to the inner side of the circular fence. In this case, a set of a command code 7 or 8 and an execution time T are continuously transmitted to the ith mobile robot for every specified execution time. The command code 7 or 8 is determined by the sign of $\dot{\varphi}_i$. The agent "Turn to left or right" has a higher priority than "Move forward". Thus, while the agent "Turn to left or right" has the execution right, then the agent "Move forward" is suppressed, i.e. blocked.

3.4.3.3. *Avoid objects*

The agent "Avoid objects" has the highest priority. When $\exists d_{ij}(k)\ [d_{ij}(k) < d_d]$ becomes true, then this agent becomes active, i.e. has the execution right at the same time. After becoming active, this agent generates the virtual velocity given by

$$\dot{\mathbf{x}}_i = -v \frac{\dot{\mathbf{x}}_{ij}}{\left\| \dot{\mathbf{x}}_{ij} \right\|} \{d_d - \min_j d_{ij}(k)\} \quad \{\exists\, d_{ij}(k)\ [d_{ij}(k) < d_d]\} \qquad [3.12]$$

where d_d is the minimum desired limit distance between a robot and an object or obstacle. The supervisory server transmits a set of command codes $j + 3$ (in case of $j = 1$, 2 or 3) or $j - 3$ (in case of $j = 4$, 5 or 6) and an execution time T to a mobile robot to avoid a collision with the nearest robot or object. Due to the activation of the "Avoid objects" agent, the mobile robot can move away from moving objects, including other mobile robots within the collision critical zone. If the distance to the nearest object is smaller than d_d, the robot tries to expand the distance to be d_d. When multiple PSD sensors simultaneously detect shorter distances than d_d, the mobile robot tries to preferentially move away from the nearest object.

Figure 3.15 shows an experimental scene of schooling behavior, in which multiple mobile robots are controlled based on the subsumption control architecture incorporated in the supervisory server as shown in Figure 3.14. It was confirmed from the experiments that the multiple mobile robots could perform a desirable schooling behavior.

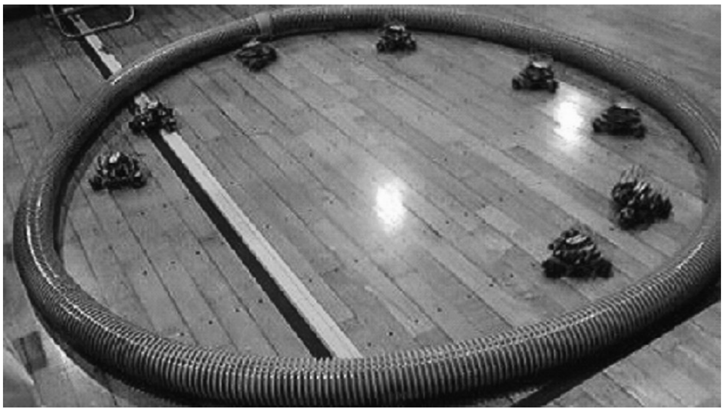

Figure 3.15. *An experiment of schooling behavior based on the proposed network-based subsumption architecture*

3.4.4. *Agent dispatcher*

The agent dispatcher supervises the controller shown in Figure 3.14. The dispatcher does not immediately move the execution right to a higher priority agent when a lower priority agent is running, because each agent works as a simple reactive behavior for every sampling period, according to sensory information. Instead of this, whenever a reaction behavior is executed during a specified execution time, the dispatcher updates the activity of each agent and gives the execution right to an updated active agent with the highest priority.

3.4.5. *Multiple sensory sensors*

The mobile robot introduced in this chapter basically has six AD conversion channels, all of which, however are already connected to six PSD sensors. To deal with sensory information from other sensors such as temperature, humidity, and force, a compact AD conversion module AGB65-ADC provided by Asakusagiken Co., Ltd. is mounted on the mobile robot. The AGB65-ADC has an AD converter that can deal with 16 analog channels. Figure 3.16 illustrates the block diagram showing the connection scheme within a mobile robot: sensors, an AGB65-ADC module and a Bluetooth module called Bluemaster.

The mobile robot has one serial port for data communication, so that text codes can be transmitted to and received from the Bluetooth module through AGB65-BT to the serial port 2 in AGB65-ADC. The Bluetooth module is used to communicate with the supervisory server PC. It should be noted that when communicating with the AD converter in AGB65-ADC to obtain the sensor's information through serial port 1, command sequence including a hexadecimal header code “0Xff” must be transmitted. In other words, the redirectable serial port on a AGB65-ADC board can switch data to serial port 1 or serial port 2 automatically, which is the important redirect function of AGB65-ADC. Using the AD conversion module AGB65-ADC, students can easily add other types of sensory sensors further to the six PSD sensors to a mobile robot.

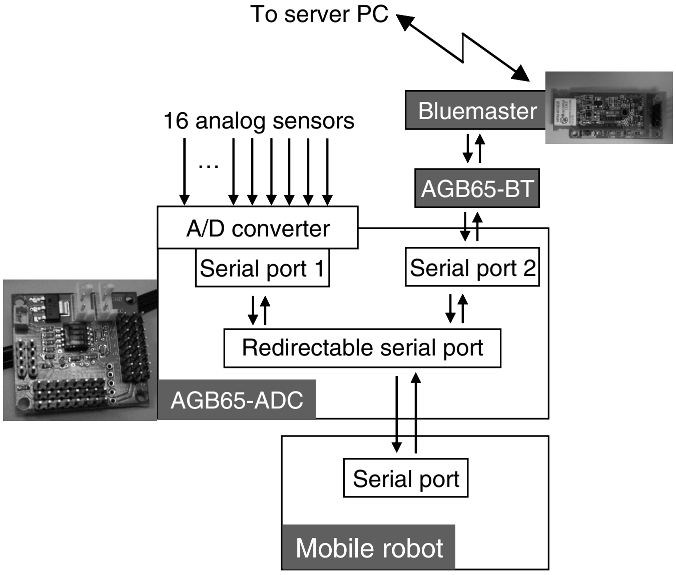

Figure 3.16. *Compact AD conversion module ADC65 with 16 channels*

3.5. Conclusions

In this chapter, a unique education system has been proposed for students pursuing mechanical engineering to learn basic mechatronics skills and techniques practically. The system is composed of three subsystems. The first subsystem is used for second year students to learn mainly input or output port operations, periodically switching LED lights ON or OFF, and a stepper motor control. The second subsystem is effective for third year students to learn AD conversion for several sensory information systems, DA conversion for DC motor control and a design of a discrete-time PI controller. Further, the third subsystem is a system of multiple mobile robots (as one of graduation studies) so that the fourth year students can learn the concept of subsumption control architecture for schooling behavior. The effectiveness of the proposed mechatronics education systems has been evaluated through experimental instructions at Tokyo University of Science, Yamaguchi, Japan.

3.6. Bibliography

[BRO 86] BROOKS R.A., "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[KIT 11] KITAHARA N., NAGATA F., OTSUKA A., "Proposal of mechatronics educational system for students", *Proceedings of the 13th IEEE Hiroshima Section Hiroshima Student Symposium*, (F)–13, p. 4, 2011.

[KIT 12] KITAHARA N., NAGATA F., OTSUKA A., SAKAKIBARA K., WATANABE K., HABIB M.K., "A proposal of experimental education system of mechatronics", *Proceedings of 17th International Symposium on Artificial Life and Robotics*, pp. 166–169, 2012.

[NAG 11] NAGATA F., YAMASHIRO T., WATANABE K., "Cooperative swarm control for multiple mobile robots using only information from PSD sensors", *Artificial Life and Robotics*, vol. 16, no. 1, pp. 116–120, 2011.

[TAN 99] TANG J., WATANABE K., KURIBAYASHI K., SHIRAISHI Y., "Autonomous control for an omni-directional mobile robot with the orthogonal-wheel assembly", *Journal of the Robotics Society of Japan*, vol. 17, no. 1, pp. 51–60, 1999.

[WAT 98] WATANABE K., SHIRAISHI Y., TZAFESTAS S.G., TANG J., FUKUDA T., "Feedback control of an omni-directional autonomous platform for mobile service robots", *Journal of Intelligent & Robotic Systems*, vol. 22. nos. 3/4, pp. 315–330, 1998.

[YAM 10] YAMASHIRO T., NAGATA F., "Cooperative control of schooling for multiple mobile robots by using only information from PSD sensors", *Proceedings of the 12th IEEE Hiroshima Section Hiroshima Student Symposium*, (F)–13, p. 4, 2010.

[YAM 11] YAMASHIRO T., NAGATA F., WATANABE K., "Cooperative swarm control for multiple mobile robots using only information from PSD sensors", *Proceedings of 16th International Symposium on Artificial Life and Robotics (AROB 16th '11)*, Oita, Japan, pp. 387–390, 27–29 January 2011.