4ℓ

Advanced Robotics, Vol. 17, No. 9, pp. 925–950 (2003)

Also available online - www.vsppub.com

Full paper

A system for self-diagnosis of an autonomous mobile robot using an internal state sensory system: fault detection and coping with the internal condition

KUNIAKI KAWABATA [1,*], SHINNOSUKE OKINA [2], TERUO FUJII [3] and HAJIME ASAMA [4]

[1] *Distributed Adaptive Robotics Research Unit, RIKEN, 2-1 Hirosawa, Wako, Saitama 351-0198, Japan*
[2] *NEC Co., 4-14-22 Shibaura, Minato-ku, Tokyo 108-8558, Japan*
[3] *IIS, University of Tokyo, Komaba Research Campus, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan*
[4] *RACE, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan*

Received 31 January 2003; accepted 27 March 2003

Abstract—Considering that intelligent robotic systems work in a real environment, it is important that they themselves have the ability to determine their own internal conditions. Therefore, we consider it necessary to pay some attention to the diagnosis of such intelligent systems and to construct a system for the self-diagnosis of an autonomous mobile robot. Autonomous mobile systems must have a self-contained diagnostic system and therefore there are restrictions to building such a system on a mobile robot. In this paper, we describe an internal state sensory system and a method for diagnosing conditions in an autonomous mobile robot. The prototype of our internal sensory system consists of voltage sensors, current sensors and encoders. We show experimental results of the diagnosis using an omnidirectional mobile robot and the developed system. Also, we propose a method that is able to cope with the internal condition using internal sensory information. We focus on the functional units in a single robot system and also examine a method in which the faulty condition is categorized into three levels. The measures taken to cope with the faulty condition are set for each level to enable the robot to continue to execute the task. We show experimental results using an omnidirectional mobile robot with a self-diagnosis system and our proposed method.

Keywords: Self-diagnosis system; autonomous robot; internal sensory system; coping measures.

1. INTRODUCTION

Most of the research work on intelligent robots has been done in which the system is ideal [1–3]. When the system becomes faulty, it is difficult to determine which

*To whom correspondence should be addressed. E-mail: kuniakik@riken.cp.jp

part is at fault and what has caused the fault. The importance of dealing with these problems has been pointed out by researchers in intelligent systems [4]. For example, in the Robocup tournament [5] problems arose because of differences between the test conditions in the laboratory and the conditions at the tournament site. Usually analyzing such problems takes a lot of time because the robots have no error detection system. Although the importance of coping with faulty situations is pointed out in the technical reports, it seems that the topic is not discussed as widely as the intelligent function. Therefore, it is important to discuss the development of a system for self-diagnosis, in which the robot is able to recognize its own condition.

Usually, an autonomous robot is a complete single system and it has different characteristics to supervised industrial robots in a controlled environment. For example, there is the possibility that a faulty condition may develop in an unknown environment and situation. Then, it becomes necessary to judge the condition utilizing an on-board sensory system. In previous work on self-diagnostic systems for robots, a fault tree diagnostic method dividing the system into multiple subsystems has been proposed [6, 7]. Cocca *et al.* [8] and Shin *et al.* [9] discussed a control method for a redundant manipulator for coping with mechanical faults at the joints. These researchers used a system that functioned at the control level to cope with the condition, but there were still undetectable parts in the system. Therefore, we have developed an internal sensory system to enable detection of faulty parts in an autonomous mobile robot. In this paper, we describe the construction of an internal state sensory system and propose a fault detection method for improving the reliability of the robotic system. Also, we applied such a system to a real omnidirectional mobile robot and attempted some basic fault detection experiments. Usually, the CPU and other processors are utilized for diagnosis and control in autonomous mobile robot systems. However, this type of fault diagnosis system does not work because the whole system is not working precisely in such a case. The system consists of a sufficient number of sensors around the modules, from which the system is constructed. Generally, when a fault occurs in a module in the system, it is not expected that the module will recover. Thus, under a faulty condition, it is either returned to the home position or it continues to execute the given task with the remaining function. Therefore, we have added some functions to our sensor system to realize intelligent behavior.

In this paper, we examine the generation of operations required to cope with faulty conditions, focusing on the function of the robotic system. The system is constructed on an omnidirectional mobile robot. We have performed experiments utilizing the mobile robot and here we show the results.

The scope of this paper is to detect its own condition and state, and cope with them for compensating mobile function by managing with what the system has. In this paper, as an example target for simple discussion, we examine a basic system for diagnosing the control system of a mobile robot (mobile function) and generation of coping behavior as a self-adaptation function. Therefore, we assume that the target task of the mobile robot is to go in a circular motion on the flat floor as a simple situation in this paper.

In Section 2, we describe the types of fault which we will discuss in this paper. In Section 3, our self-diagnosis system for the mobile robot is introduced. System configuration and coping function to the self-condition are discussed in Sections 4 and 5, respectively. In Section 6, we examine our proposed system using a real mobile robot. Finally, we conclude this paper in Section 7.

2. TYPES OF FAULTS

First, we define a fault as occurring when there is a difference between the realizable function and the required function. The realizable function is what is actually executed by the system and the required function indicates what the system should normally execute. Generally, the reason the realizable function decreases can be roughly classified into three categories as follows (Fig. 1).

(i) Fatigue (deterioration).

(ii) Noise (sudden fault).

(iii) Initial failure.

In case (i), the realizable function declines gradually over the period of time the system has been working. In case (ii), the realizable function declines suddenly. An example of (ii) is when the robot collides with an obstacle. In case (iii), the robot is already in a faulty condition before the systems starts. Case (iii) can usually be avoided by inspection and maintenance before start-up. Therefore, we shall discuss cases (i) and (ii) only in this paper. Here, we consider that general mechatronic systems consist of plural functional modules (subsystems) and the condition of each module can be detected by equipping them each with a sensor. The type of fault that occurs in such a system can be roughly classified either as a module fault or a sensor fault. For example, if one module in the system becomes faulty, it is impossible to expect the module that is connected to it to work properly. On the other hand, if a sensor becomes faulty, the condition cannot be measured accurately. Therefore, the motion of the robot is also inaccurate. Diagnosis in the field is different to that

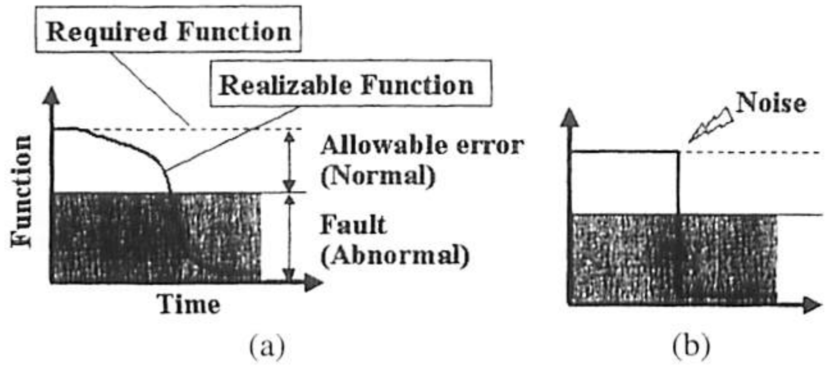


Figure 1. Types of fault. (a) Fatigue/wear. (b) Noise.

in the laboratory, because there is a possibility that a fault occurs in an unknown environment and situation. However, in general, in robot systems used for research, sensory information is utilized for controlling the system. Thus, it is difficult to judge the kind of fault and which module is faulty. Therefore, by attaching sufficient sensors to every module in an autonomous mobile robot, we are able to categorize the fault and specify the faulty module.

3. SELF-DIAGNOSIS SYSTEM FOR AN AUTONOMOUS MOBILE ROBOT

3.1. Concept

The concept of our proposed self-diagnosis system for an autonomous robot consists of three processes: internal condition sensing, diagnosis and coping with the faulty condition. This means the system not only detects the faulty module, but also makes adjustments so that it adheres to the original motion plan after the fault occurs. In this study, we define the fault as the difference between the required function and the realizable function. For example, it may occur due to fatigue, noise, etc. Usually, a system like that of an autonomous robot consists of plural modules (subsystems). Thus, to maintain the functional level of the system, utilizing probability statistics from each part of the system can be done. However, in reality there are so many parts and modules in a robotic system.

We have introduced a series of diagnostic processes to understand the current condition of the robot before, during and after working. Functional diagnosis before working is useful for changing the original control plan according to the condition of the robot and for determining how each d.o.f. of the robot can work. Diagnosis during working should monitor sudden and unexpected faults. After working, diagnosis will enable determination of whether maintenance is required or not. Also, this allows data to be collected for estimating the conditions of the robot at the next step. For diagnosing before and after working, the robot would execute a planned motion and collect internal information from each module. Thus, in these stages, warming up and cooling down motion must be done. Before and after execution of a task, the robot is activated and moves so that information can be collected. It is required, for example, to detect its condition by moving it in a two-dimensional plane using all the d.o.f. During the time the robot is working, the diagnostic system collects the information passively. In this stage, the robot monitors internal state information according to the task actions.

3.2. Diagnostic method

Diagnostic methods based on a model are typically applied to diagnosing general systems [10–12]. In this method, the diagnostic system has a model of the target system and the model is utilized to compare this with the observed performance (Fig. 2). The model-based method is suitable for flexible diagnosis and dealing with

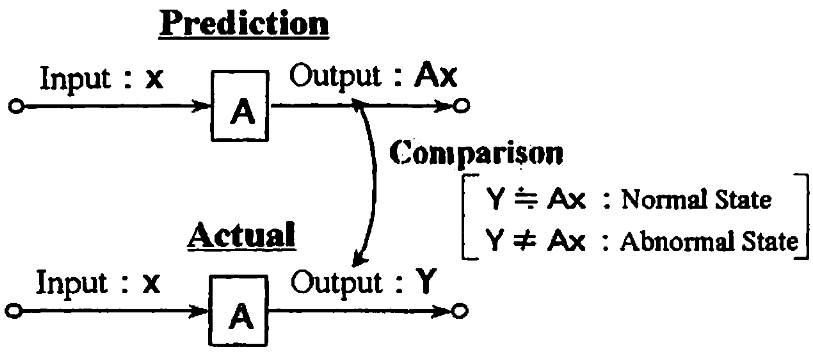


Figure 2. Model-based fault detection.

unexpected faults. In other words, the faulty part is rapidly detected by monitoring the input value and output performance at each module. In our system, we also utilize a model-based diagnostic algorithm for each module.

Figure 2 shows our algorithm for detecting faulty parts and the cause of the fault. When a mechanical fault occurs in the system, it can be detected easily because there is no consistency between the input and output values to and from the module. However, when a sensor fault occurs, it is not so easy to detect the faulty part. Therefore, we introduce an algorithm to confirm the consistency of the sensory information from each module to enable detection of the faulty part. Here, in the case of a CPU that becomes faulty, the whole system, including the control and sensory systems, cannot work and diagnose normally. Therefore, we do not treat computer faults in this study.

3.3. Fault detection algorithm

As mentioned above, the system consists of plural functional modules. When the diagnostic system detects any faults, they are classified into 'system fault', 'sensor fault' and 'combined faults'. A system fault indicates that the module (hardware) has become faulty and a sensor fault means that the sensor has become faulty. A combined fault means that both system and sensor faults occur simultaneously. For diagnosing each module, the sensory information is compared with the predicted value based on the input value. If the module is not faulty, the consistency between input and output values is confirmed with the neighboring modules. No consistency means the module is in a faulty condition. As an example, we consider a group of modules consisting of three modules A, B and C (Fig. 3). If there is no consistency between the information from sensors S1 and S2, it is possible that module B is faulty. However, using only sensors S1 and S2, it is difficult to decide whether it is a system or a sensor fault. Therefore, we check for consistency among the other sensory information to decide which is the faulty condition. If there is consistency between information from sensor S2 and S3, but inconsistency between S1 and S3, it is possible that system B is faulty. If there is no consistency between information from sensor S2 and S3, but consistency between S1 and S3, it is possible that sensor S2 is faulty. As for the previous example, it is difficult to decide whether it is

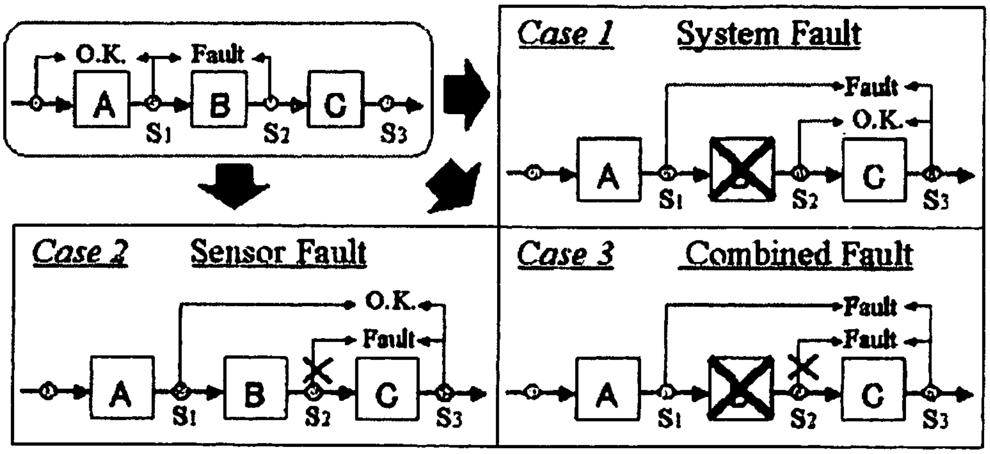


Figure 3. Fault detection algorithm.

a system or sensor fault. In this case, we can decide the sort of fault by checking the consistency between information from sensors S2 and S4 (sensor fault or combined fault). Utilizing these procedures, each module condition is classified as 'normal', 'system fault', 'sensor fault' or 'combined fault'. Also, the system can detect which part is the cause of the fault.

4. SYSTEM CONFIGURATION

4.1. Internal state sensory system

For the experimental platform, we utilized an omnidirectional mobile robot: ZEN-450 (Fig. 4), which can realize omnidirectional motion with a special driving mechanism and four wheels with free rollers [13]. The omnidirectional mobile robot: ZEN-450 was developed by the robotics group of RIKEN and has Vx-Works installed. The sensors for self-diagnosis are arranged around each module of the driving and power supply systems. Figure 5 shows an overview of the omnidirectional mobile robot with the internal state sensory system. The current sensors and the voltage sensors are installed on each electrical power line for monitoring input and output values. Also, each motor is equipped with a rotary encoder. A gyro-sensor is utilized to detect the rotational motion of the robot 's body. For measuring the wheel motion directly, a magnetic sensor is installed with a slit-processed disk on each wheel as an encoder (Fig. 6). Thus, using two encoders, the system can observe both the motor input and output values through the gears. A sensor is provided at both the input and output of the modules.

Here, we have attempted basic sensing experiments using the system. As for the power system, its condition is monitored using voltage and current intensity values. Figure 7 shows measured voltage data for the case in which there is no load and running at constant velocity. As a result, the ZEN-450 is unable to work when the battery voltage becomes 17 V. Also, the robot stops completely 5 min after the battery voltages decrease. Thus, in practice, the robot must charge its own battery

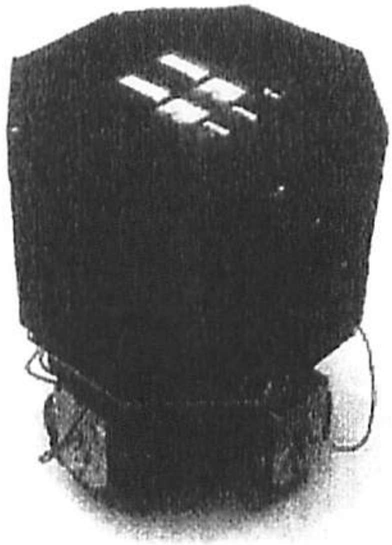


Figure 4. Omnidirectional mobile robot ZEN-450.

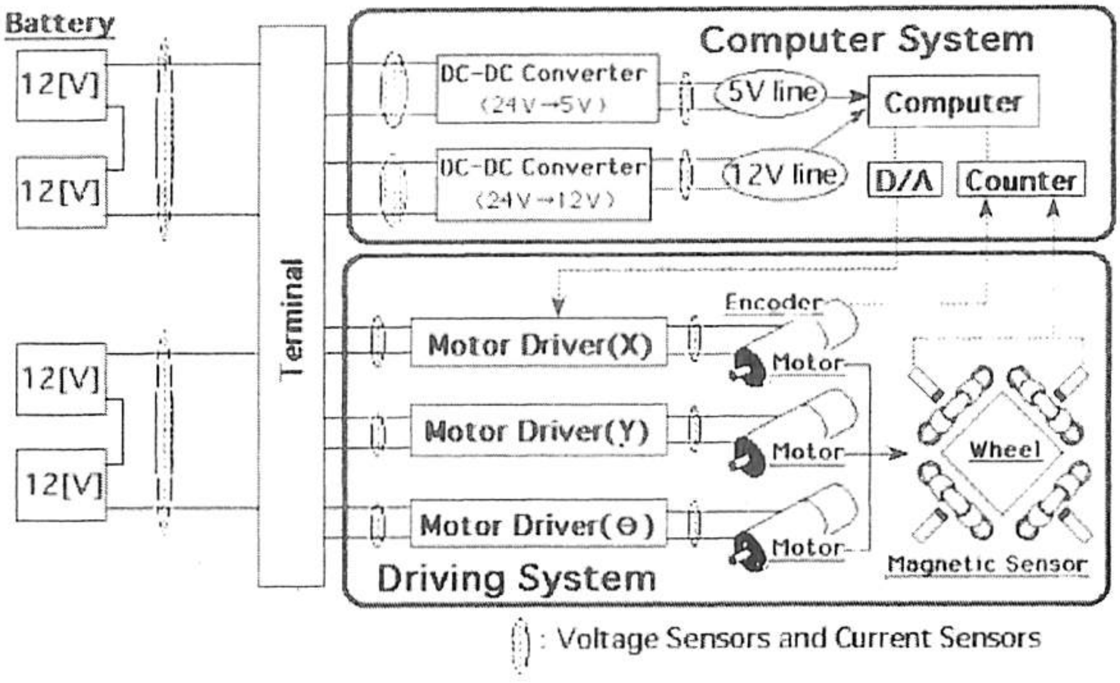


Figure 5. System overview.

when the voltage value drops below the threshold (approximately 23 V). For the driving system, a transmission fault is detected at the gears and axles by monitoring the motor rotation and the wheel rotation. When the ZEN-450 plans to go in a straight line movement, the error increases due to the effects of wheel slipping and the 'back-rushing' of the gears for the power transmission (Fig. 8). For example, if the rotational velocities of the left and right wheels are not equal, the robot's posture changes although the robot tries to move in a straight line (Fig. 9). We utilize sensory information to detect faulty parts in the power and driving systems.

Figure 6. Magnetic sensor and slit.

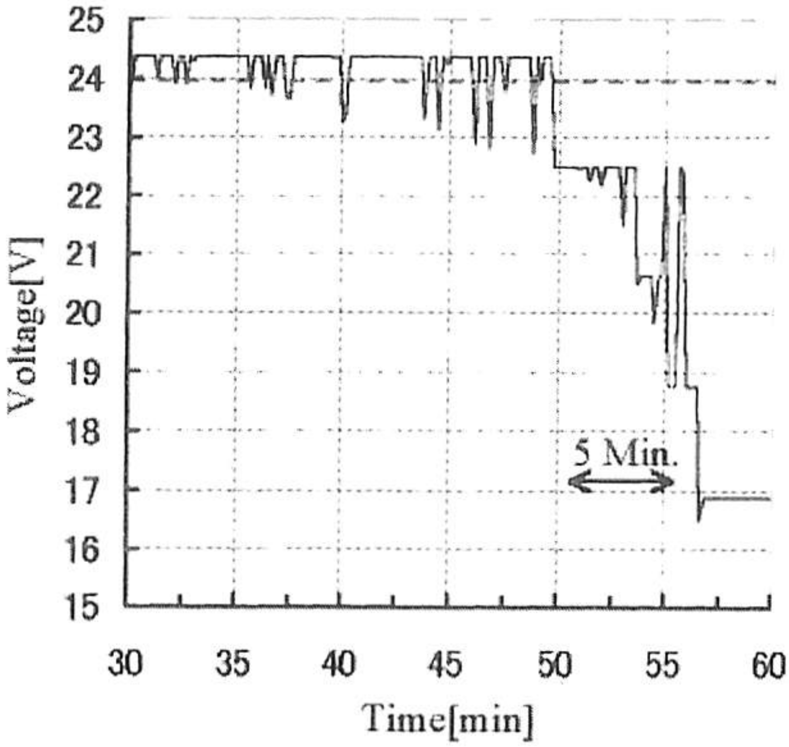


Figure 7. Experimental result: voltage level.

4.2. Diagnosis experiment

Table 1 indicates the conditions of the ZEN-450 in each situation. 'O' indicates the normal output — a state in which the output from each module is normal for the input command. 'X' indicates an abnormal state, which means the hardware modules or the equipped sensors have become faulty. For example, if a motor is faulty, the motor or the wheel that is connected to the motor would stop rotating. On the other hand, if the encoder on the motor is faulty, the wheel would rotate

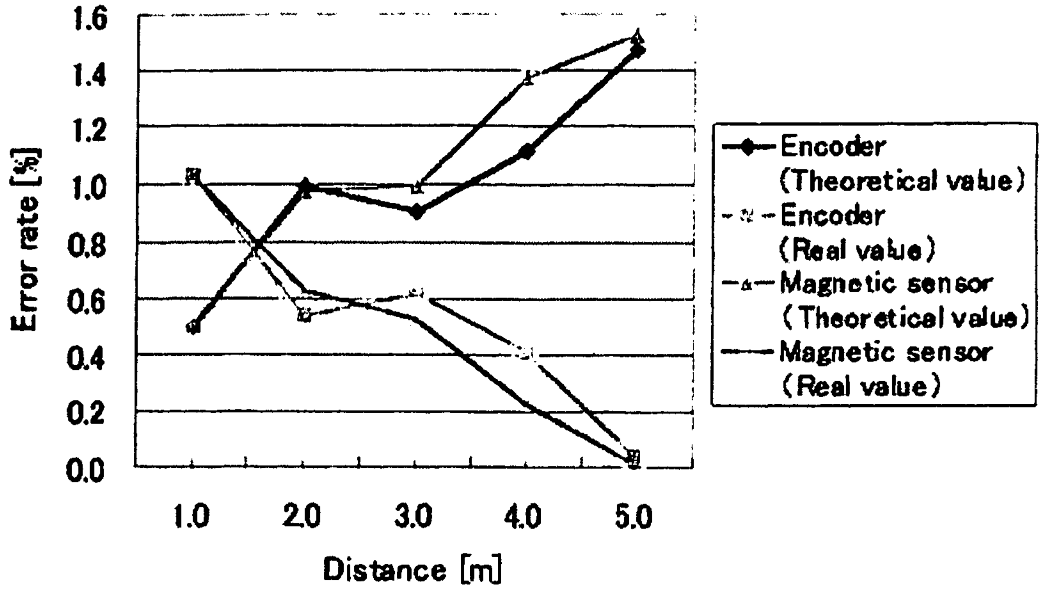


Figure 8. Experimental result: driving wheel(2).

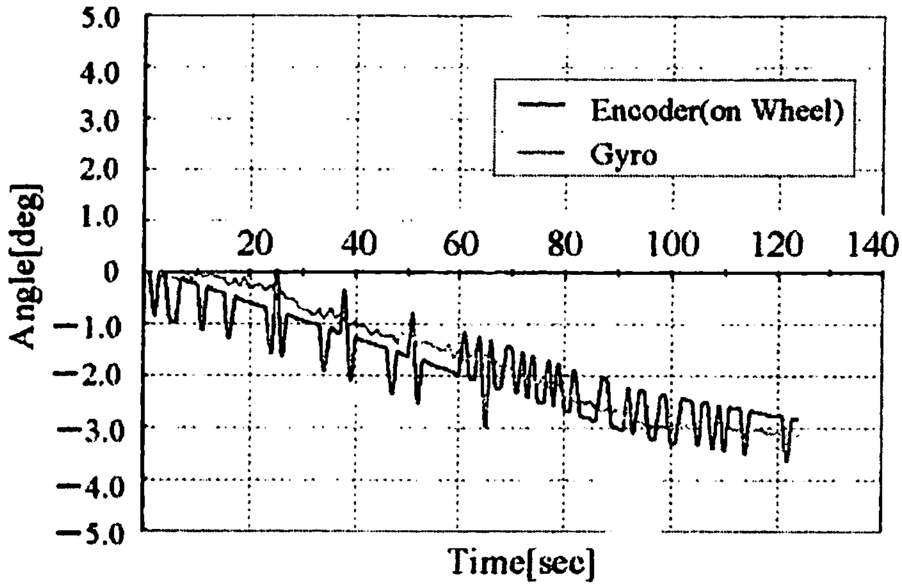


Figure 9. Experimental result: driving wheel(1).

normally, but the rotational information would not be measured. Using Table 1, it is possible to determine the robot's condition by classifying the condition of each module. For recognizing such faulty conditions, individual criteria for judging each module are needed. Here, we set the criterion for the rotational velocity of a DC motor and a wheel as in Figs 10 and 11. As we mentioned, in this paper, we cosider examining the basic system for diagnosing the mobile function. Therefore, we assume that the working environment is on a flat floor. These are determined from empirical knowledge using error values and error rates in the velocity under the normal condition. When the output value is above this threshold, the module is judged to be in a faulty condition. The gray areas in Figs 10 and 11 are the values for

Table 1.
System conditions

Sensor output	Condition (a faulty point)											
	Normal	System fault						Sensor fault				
		Wheel slip	Gear, axle fault	Motor fault	Motor driver fault	Power line fault	Low battery level	Encoder fault (on motor)	Encoder fault (on wheel)	Current sensor fault (on motor driver)	Current voltage sensor (fault (on power line))	Current voltage sensor (fault (on battery))
Current, voltage sensor (on battery)	○	○	○	○	○	○	×	○	○	○	○	×
Current, voltage sensor (on power line)	○	○	○	○	○	×	×	○	○	○	×	○
Current sensor (on motor driver)	○	○	○	○	×	×	×	○	○	×	○	○
Encoder (on motor)	○	○	○	×	×	×	×	○	×	○	○	○
Encoder (on wheel)	○	○	×	×	×	×	×	×	○	○	○	○
Gyro sensor	○	×	×	×	×	×	×	○	○	○	○	○

○ = normal output; × = abnormal output.

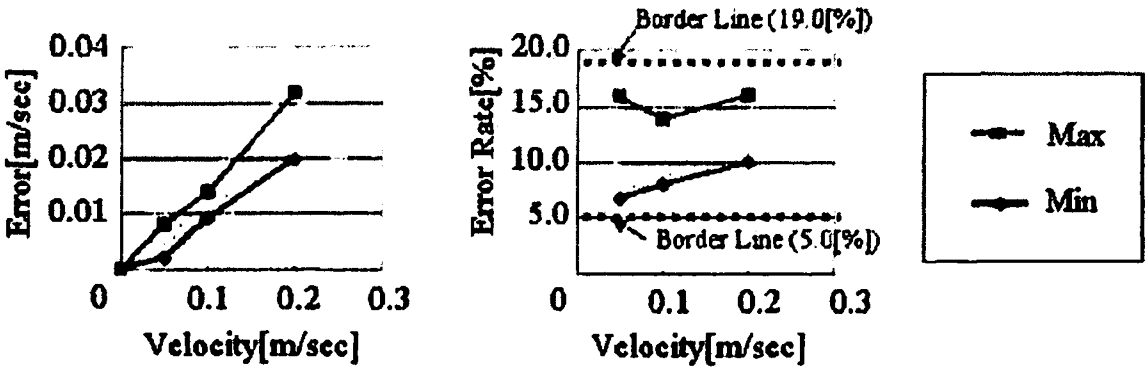


Figure 10. Permissible error range (motor).

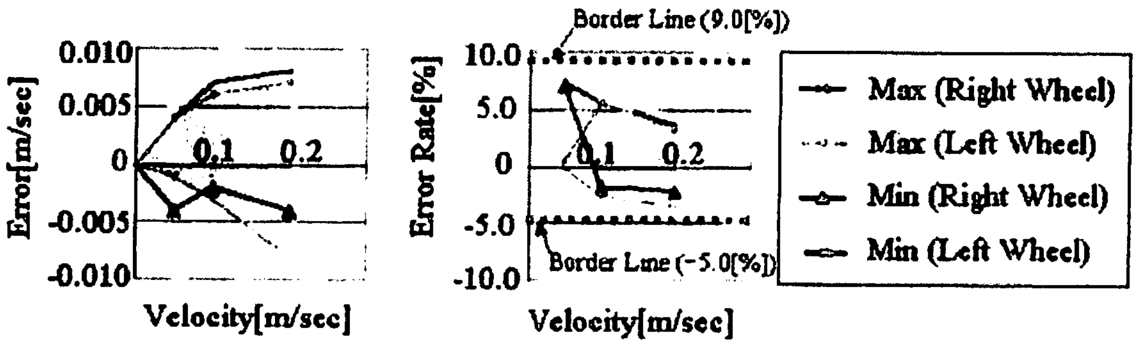


Figure 11. Permissible error range (wheel).

the experimental results and the line with symbols indicates the criterion for judging the fault. Thus, border lines (dotted lines) in the left panels indicate upper and lower limits of the permissible error range for the normal condition. When the sensory value is over this range, the system is judged to be be in an abnormal condition. In this experiment, we utiilize the criteria based on our empirical kowledge which can be observed when the robot works normally. It is our future work to detemine the criteria adaptively according to the environmental condition and task.

Using the proposed algorithm and judgment criteria, we have performed diagnostic experiments on the driving system of an omnidirectional mobile robot. The purpose of this experiment was to detect motor (system) faults and encoder (sensor) faults. Figure 12 shows an experimental result in which a single motor is driven with an intentional accident (detaching the power or information line). When the sensory output is within the permissible range, it is normal. Otherwise, it is diagnosed as being faulty. In Fig. 12, the problem is detected at point 1. After that, the system diagnoses the sort of fault (sensor fault, combined fault and system fault at points 2, 3 and 4, respectively). Then, we switch the connection state of the lines, intentionally. In such a case, the system can detect and classify its own conditions based on the proposed system.

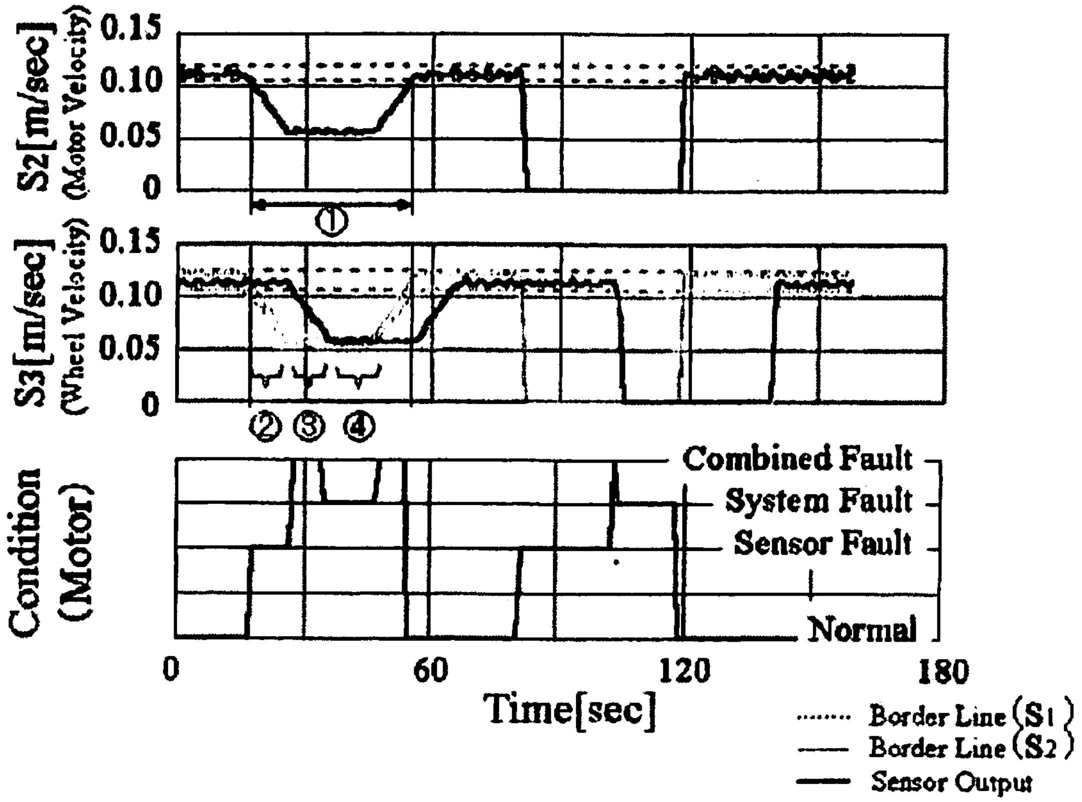


Figure 12. Experimental result.

5. COPING WITH THE CONDITION

In this section, we attempt to advance our self-diagnostic system to enable it to cope with the condition of the robot system.

5.1. Fault levels

It is not usually desirable to apply the same measures for coping with all types of fault, because there is a possibility that some of the modules that work normally are sacrificed. To avoid this, we need a method that copes with each faulty condition. Therefore, we introduce a scheme whereby the faults are classified into three levels as follows (Table 2):

(i) The system retains the normal state of a function but with revision(s) (Fault Level 1).

(ii) Although the system loses a function(s), it can recover it using another function(s) (Fault Level 2).

(iii) The system completely loses a function (Fault Level 3).

We examine the measures taken to cope with each fault level using these definitions. After the occurrence of a fault, confirmation is needed as to whether the robot can act safely or not. If there is a possibility that the robot might develop an adverse condition, it has to stop its own motion or task. If there is no such

Table 2.
Fault level

	Fault Level 1	Fault Level 2	Fault Level 3
Functional condition	decline	partial loss	complete loss
Coping method example	revision using feedback	revision using the other function	stop
Fault example	setscrew loose, bias generation	part of d.o.f. not working	out of control

possibility, the system continues executing the task and judges the fault level based on the condition of the faulty function. In Fault Level 1, the system does not fail in the stationary condition. Therefore, the system retains the function, but with revisions. For example, for the case in which the wheel velocity does not reach the reference value due to mechanical trouble, the actual reference value is increased to a value such that the original reference value is obtained. When the posture of the robot is changed due to a drift in the motor while the robot is running, the feedback needs to be revised. In Fault Level 2, some functions in the system are independent and the system can recover the function using other functions. If necessary, the system should revert to its base or home position to avoid Fault Level 3 developing. When the robot system is in Fault Level 3, the system must be stopped because it cannot function correctly. If functional compensation is done at Fault Level 2, recovery of the function is not applied to continue the task in every case. For example, in the case of transportation, it is important to reach the destination, but in the case of inspection of a power plant, both posture and position need to be controlled simultaneously. Focusing on the means of the function, the latter case is already Fault Level 3. Thus, we apply different measures for coping with each fault level to realize effective and reliable task execution in each case.

5.2. Classification of Fault Level

Generally, most of the working environments for autonomous mobile robots are unknown or uncertain. Therefore, shifts in the fault level occur while the robot copes with a faulty condition. Indices required for judging the fault level also need to be determined. If our proposed algorithm, which performs a detailed diagnosis, is used while the robot executes a task, the robot must be controlled using an open loop scheme. This means that it is not applied when control of the robot is based on a feedback value. Here, we focus on each functional unit of the robot. While the robot executes the task, it is important to judge the condition of the functional unit rather than the reason for the fault. We propose a method for classifying the faulty condition into fault levels.

Figure 13 shows an example system for diagnosing the function. It consists of a D/A converter, a motor driver and a motor (wheel). Using a reference value V_{ref},

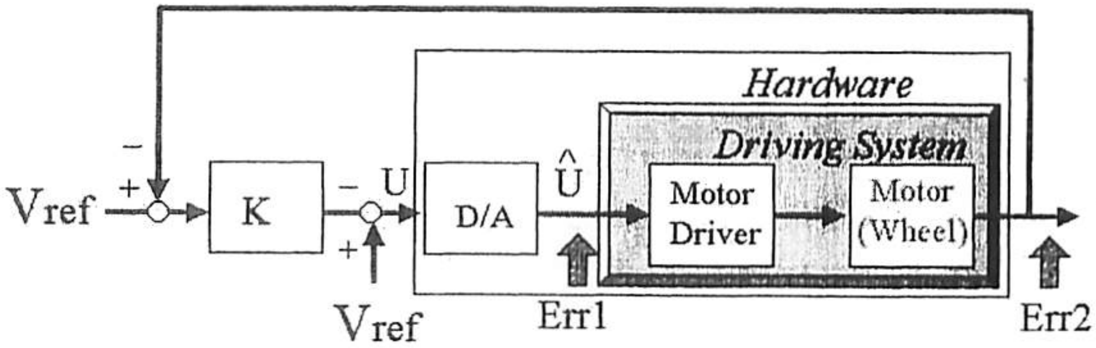


Figure 13. A target system.

the measured values V, U, Err1 and Err2 are derived from the following equations.

$$U = V_{\mathrm{ref}} - K(V - V_{\mathrm{ref}}), \tag{1}$$

$$\mathrm{Err1} = \begin{cases} \left| \dfrac{\widehat{U} - V_{\mathrm{ref}}}{V_{\mathrm{ref}}} \right| & (V_{\mathrm{ref}} \neq 0) \\ 0 & (V_{\mathrm{ref}} = 0), \end{cases} \tag{2}$$

$$\mathrm{Err2} = \begin{cases} \left| \dfrac{V - V_{\mathrm{ref}}}{V_{\mathrm{ref}}} \right| & (V_{\mathrm{ref}} \neq 0) \\ 0 & (V_{\mathrm{ref}} = 0). \end{cases} \tag{3}$$

Here, U is the input value to the module. K and $\widehat{U}$ are a positive value and a D/A value corresponding to the value of U, respectively. Err1 and Err2 indicate the error rate values of the input and output. If there is no problem in continuing to execute a task after a fault has occurred, the value of Err1 is measured and a judgement is made as to whether the feedback value is normal or not. When the output is reduced by the faults, the feedback value increases to maintain the output at the reference value. We determine this to be Fault Level 1 if this value is above the threshold and the function can be recovered by feedback. Moreover, if a fault that cannot be recovered using feedback occurs in the system, the system output decreases. In such a case, the system measures the value of Err2 and judges whether the output is correct or not. If the value is less than the criterion, the system judges it to be Fault Level 2 because the function cannot work normally. The faults are classified by using this algorithm for each function of the system. The criteria of the fault are established utilizing experimental data.

First, we make observations about the value of Err1. Figure 14 shows experimental results of V, $\widehat{U}$ and Err1 when the ZEN-450 runs in a straight line at 0.1 m/s. We reduce the sensor value to introduce an intentional faulty condition using the software. Although the velocity of the wheel keeps the reference value, $\widehat{U}$ is increased. Also, the value of Err1 is increased because of the function being faulty, but the feedback control law revises the functional effect. Here, we set the threshold value of Err1 to 0.2 (20% of the reference value) based on the results of running experiments. The feedback value is limited by this because it will otherwise burden the robot's hardware if it is above the criterion. Next, we observe the behavior

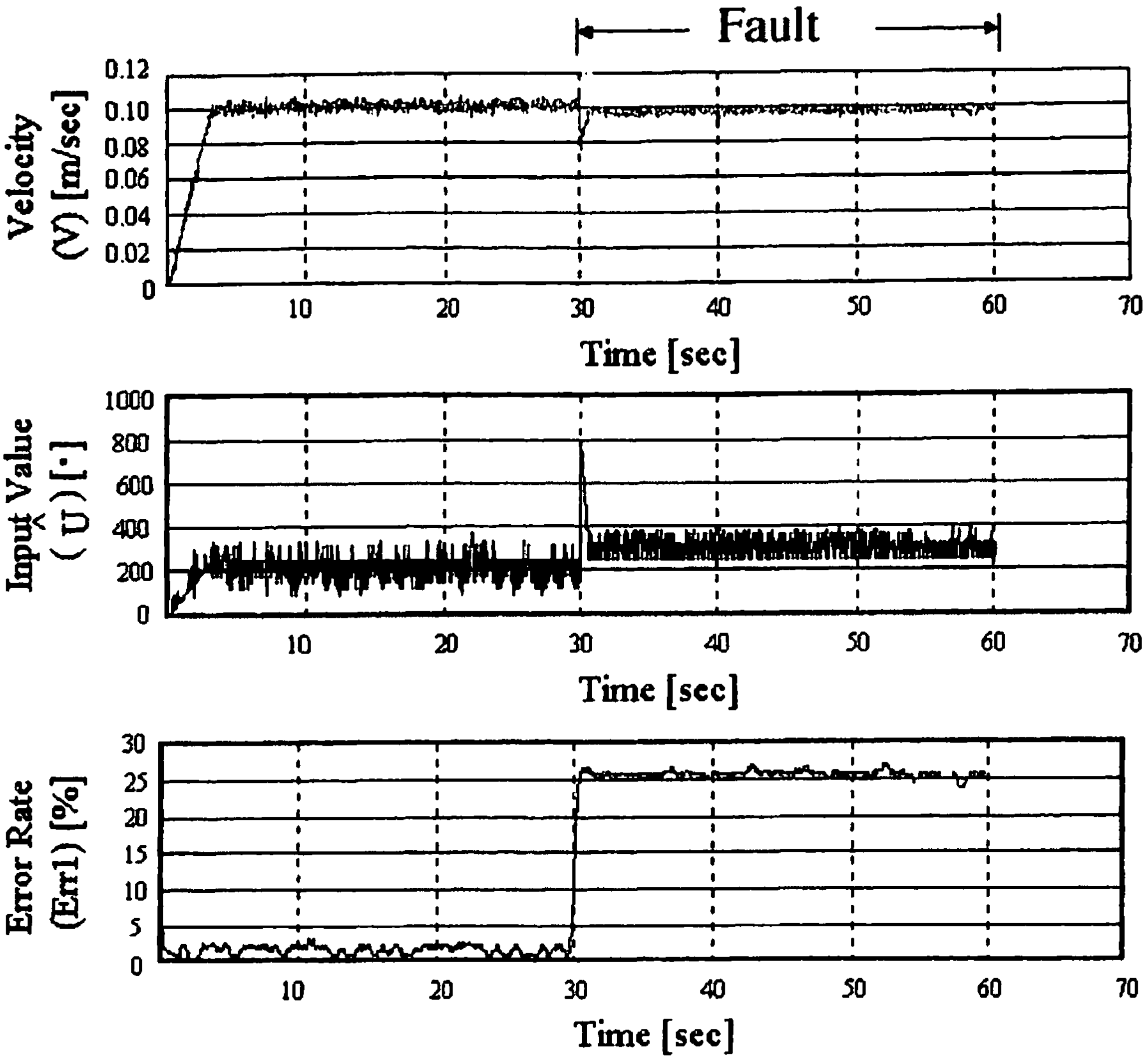


Figure 14. Experimental result for Err1.

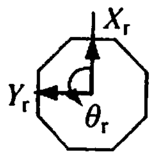
of the value of Err2. When Errl is not an extraordinary value, the function is in a normal condition and the value of Err2 becomes approximately zero. However, if Err1 is above the upper limit, the feedback control is limited and the value of Err2 increases. Therefore, we also classify faulty conditions using Err2, because if the value of Err2 is above the criterion, the function cannot work anymore. Here, we set the threshold value of Err2 to 0.2 (20% of the reference value) utilizing the result of the running experiments. When Errl is above the threshold, the function moves to Fault Level 2. Using this method, the functional fault can be classified and the condition of the system can be judged.

5.3. Functional compensation

Our system (an autonomous mobile robot), which consists of multiple functional modules, can, by itself, compensate for a faulty function using other modules. For example, the ZEN-450 has 2 translational d.o.f. and one rotational d.o.f. This kind

Table 3.
Required function under Fault Level 2

Faulty function	Required motion						
	X_r	Y_r	θ_r	$X_r \cdot Y_r$	$X_r \cdot \theta_r$	$Y_r \cdot \theta_r$	$X_r \cdot Y_r \cdot \theta_r$
x-axis	$-\alpha$ $\downarrow$ Y_r $\downarrow$ $+\alpha$	—	—	$-\beta$ $\downarrow$ Y_r $\downarrow$ $+\beta$	$-\alpha$ $\downarrow$ $Y_r \cdot \theta_r$ $\downarrow$ $+\alpha$	—	×
y-axis	—	$+\alpha$ $\downarrow$ X_r $\downarrow$ $-\alpha$	—	$+\beta$ $\downarrow$ X_r $\downarrow$ $-\beta$	—	$+\alpha$ $\downarrow$ $X_r \cdot \theta_r$ $\downarrow$ $-\alpha$	×
Rotation	—	—	×	—	×	×	×



$\alpha = \pi/2$ (rad); $\beta = \tan^{-1} \frac{X_r}{-Y_r}$ (rad).

of system, which consists of functions connected in parallel, can, in some cases, revise a function using other functions. Here, we discuss the method by which the system recovers a lost function(s) using a normal function(s) for the case of Fault Level 2. Table 3 shows examples of the means used for coping in the case of Fault Level 2. A '−' means that there is no need to cope with the fault; 'X' indicates the inability to cope with a faulty condition using other functions. For example, if the robot loses one translational mobile function, the robot can retain the faulty function by rotating and using the other translational mobile function. After this, the robot returns to its posture by rotational motion. In this case, the robot can continue to execute the task. However, if the rotational function of the robot becomes faulty, the ZEN-450 cannot execute the task.

6. EXPERIMENTS

Here, we report on experiments that confirm that our proposed method works effectively. The task is to execute object transportation to a destination using the autonomous mobile robot. Figure 15 shows the reference trajectory for the robot. The working area is limited to 3 m × 3 m on a flat floor. Figures 16–19 show experimental results for the case of normal operation. No fault occurs during the task. Figures 20–23 show the locus of each axis of the robot when an intentional fault is introduced. At point 'A' in Fig. 22, we disconnected the power line for the Y_r axis, artificially introducing a faulty condition. The values of Err1 and Err2 go above the threshold at the moment the fault is detected. Figure 24 shows the trajectory of

1	X_r	+2.5	[m]
2	Y_r	+2.5	[m]
3	θ_r	+270	[deg]
4	Y_r	−1.5	[m]
5	X_r	+1.5	[m]
6	θ_r	−270	[deg]
7	X_r	−1.5	[m]
8	Y_r	−1.5	[m]

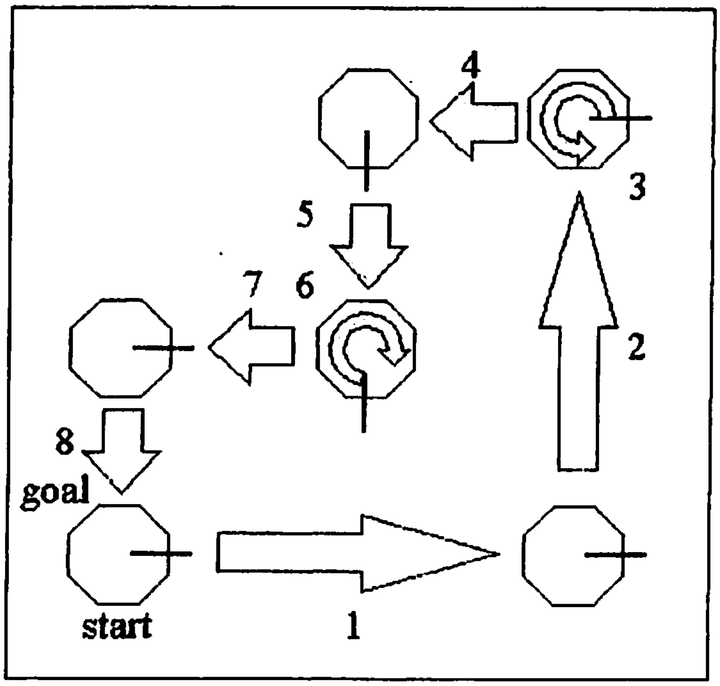


Figure 15. Planned robot's motion for a specified task.

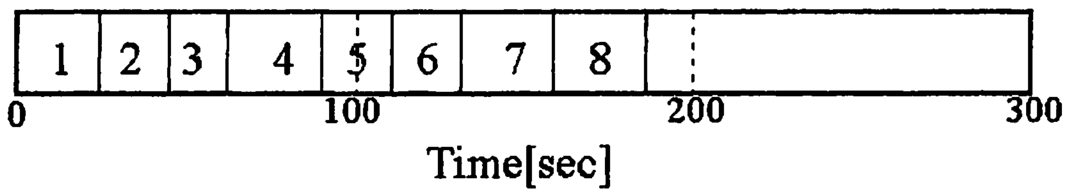


Figure 16. Robot's motion during the task (normal condition).

the robot in the case of the normal condition using dead-reckoning. As the robot loses its mobility function to move in the Y_r direction (Fault Level 2), it attempts to execute the task using the mobility function in the X_r direction and the rotational function θ_r (Fig. 25). X_r and θ_r function normally, and the system attempts to cope after fault detection. Actually, the robot turns through 90° (motion 2a) using θ_r and moves along the planned trajectory using the X_r mobile axis (motion 2b). After motion 2b, the robot turns −90° (motion 2c) to regain its posture. Whenever movement in the direction of the Y_r mobile axis is required (motion 4, 8), the robot utilizes the X_r axis to execute the task. Finally, at the destination, there is a 0.5 m positional error from the reference trajectory. This is the why we performed this experiment using dead-reckoning for motion control of the robot. Figure 26 shows the trajectory of the robot with a faulty encoder for the X_r axis during motion 2. In this case, the robot can also regulate the motion plan and arrive at the destination.

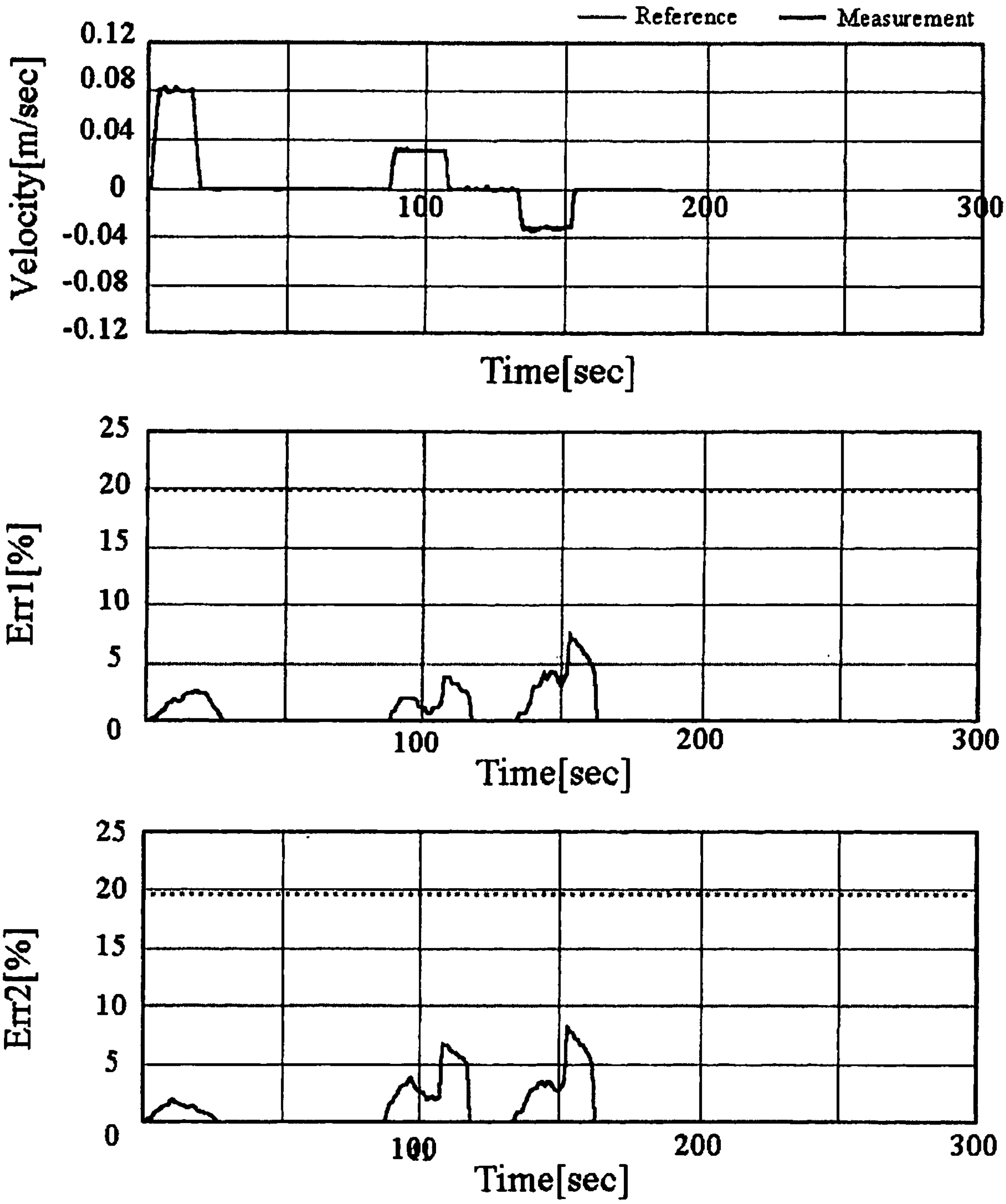


Figure 17. Velocity, Err1 and Err2 of the X_r axis during the task (normal condition).

Considering that the robot does not utilize any external sensors or information, the task can be considered to have been almost achieved. This result shows that our proposed system can work effectively in an autonomous robotic system where functional faults occur.

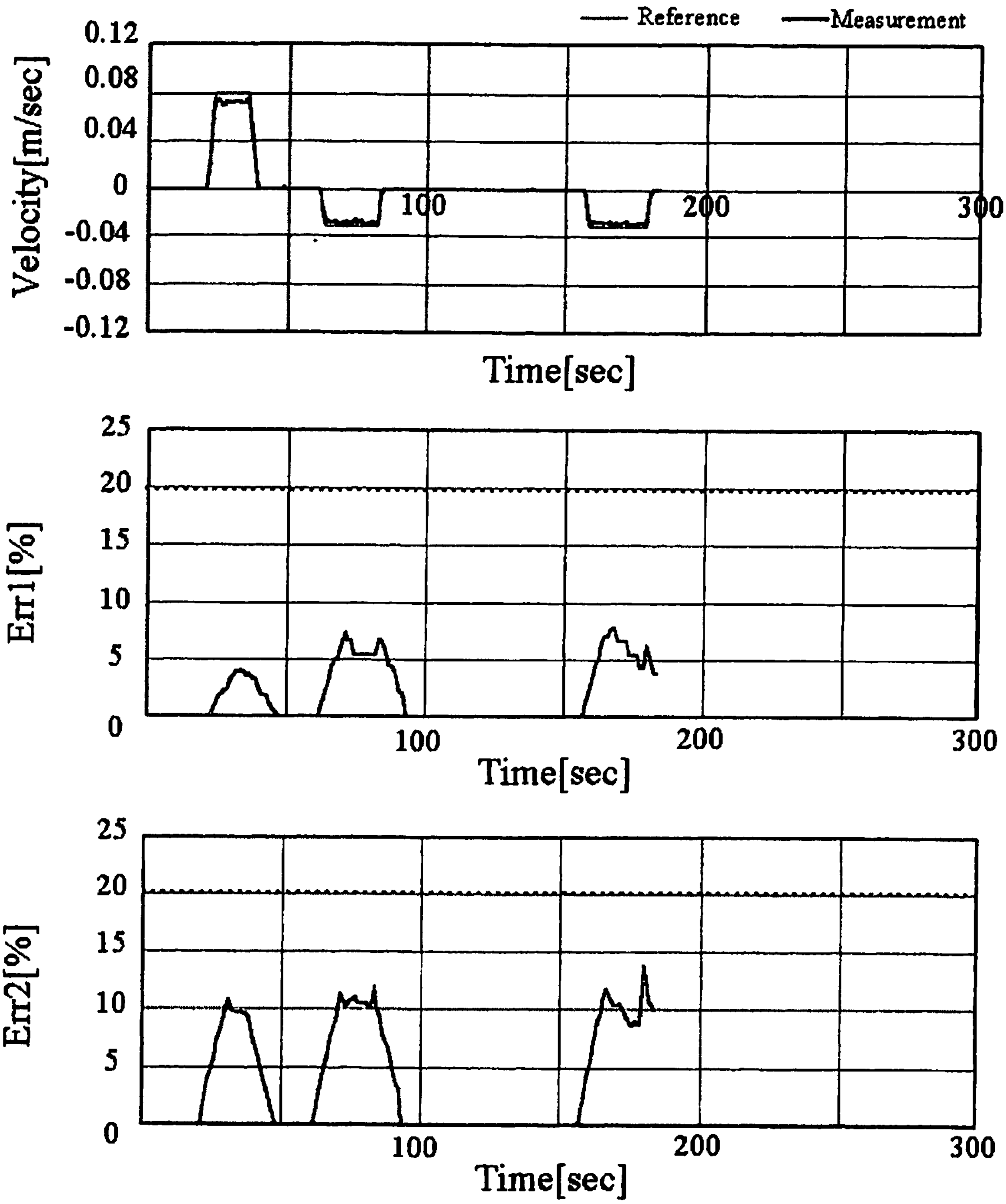


Figure 18. Velocity, Err1 and Err2 of the Y_r axis during the task (normal condition).

7. CONCLUSION

Considering that robots work in a real environment, it is very important to have a self-diagnosis function. A motion planning system that utilizes information about the internal condition is one of the necessary intelligent factors required for autonomous robotic systems. In this paper, we examine and discuss our self-diagnostic system and internal state sensory system of a robot. We defined the faults

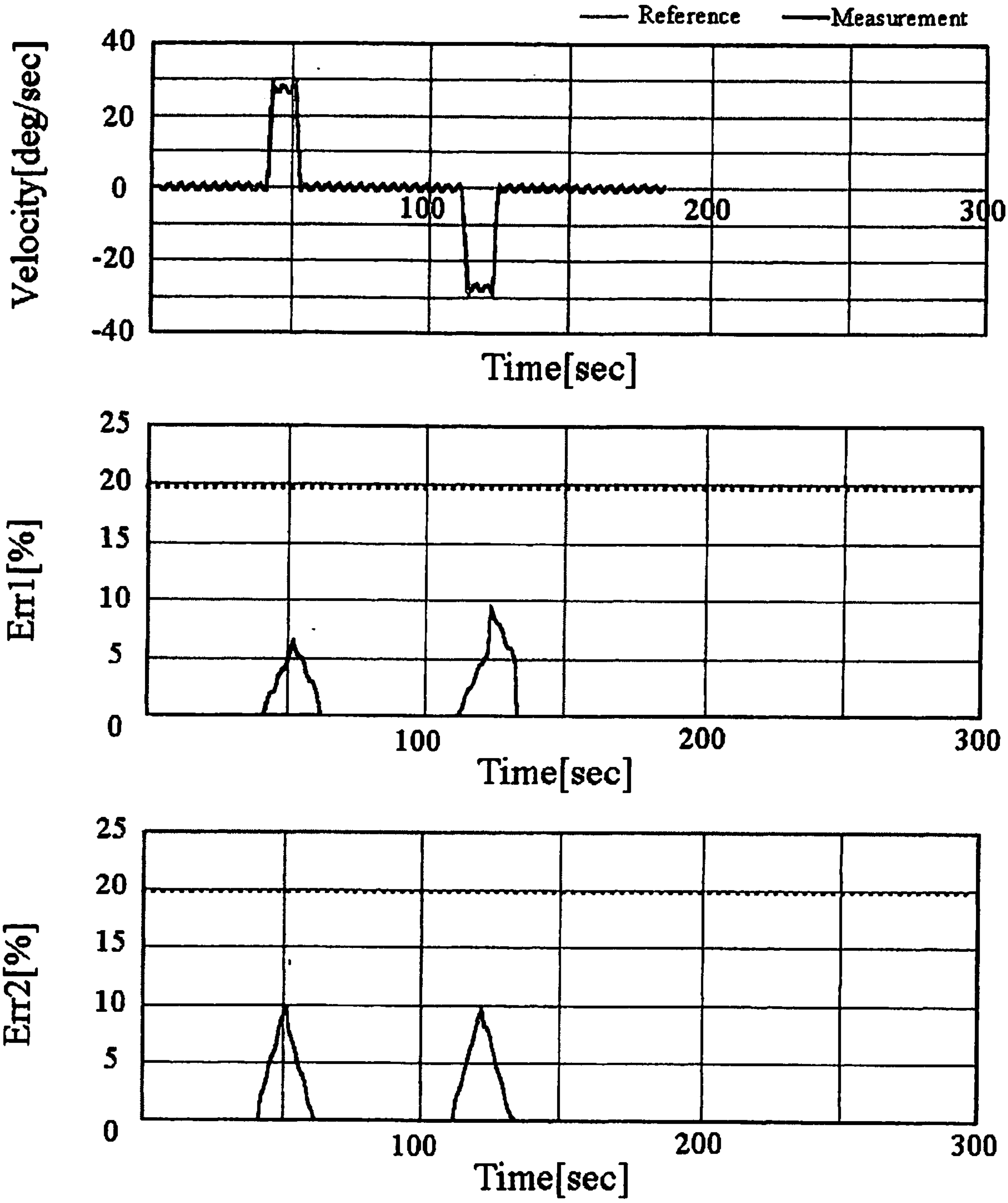


Figure 19. Velocity, Err1 and Err2 of the rotation axis during the task (normal condition).

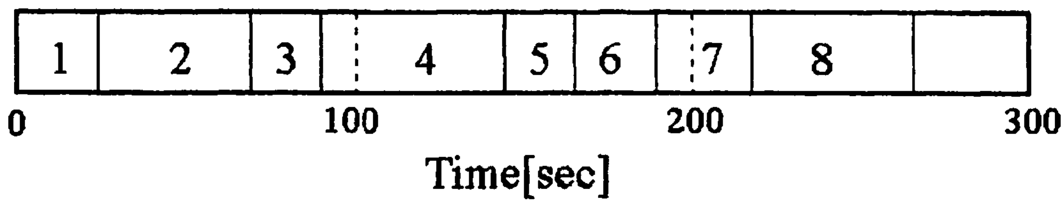


Figure 20. Robot's motion during the task (Y_r axis at fault).

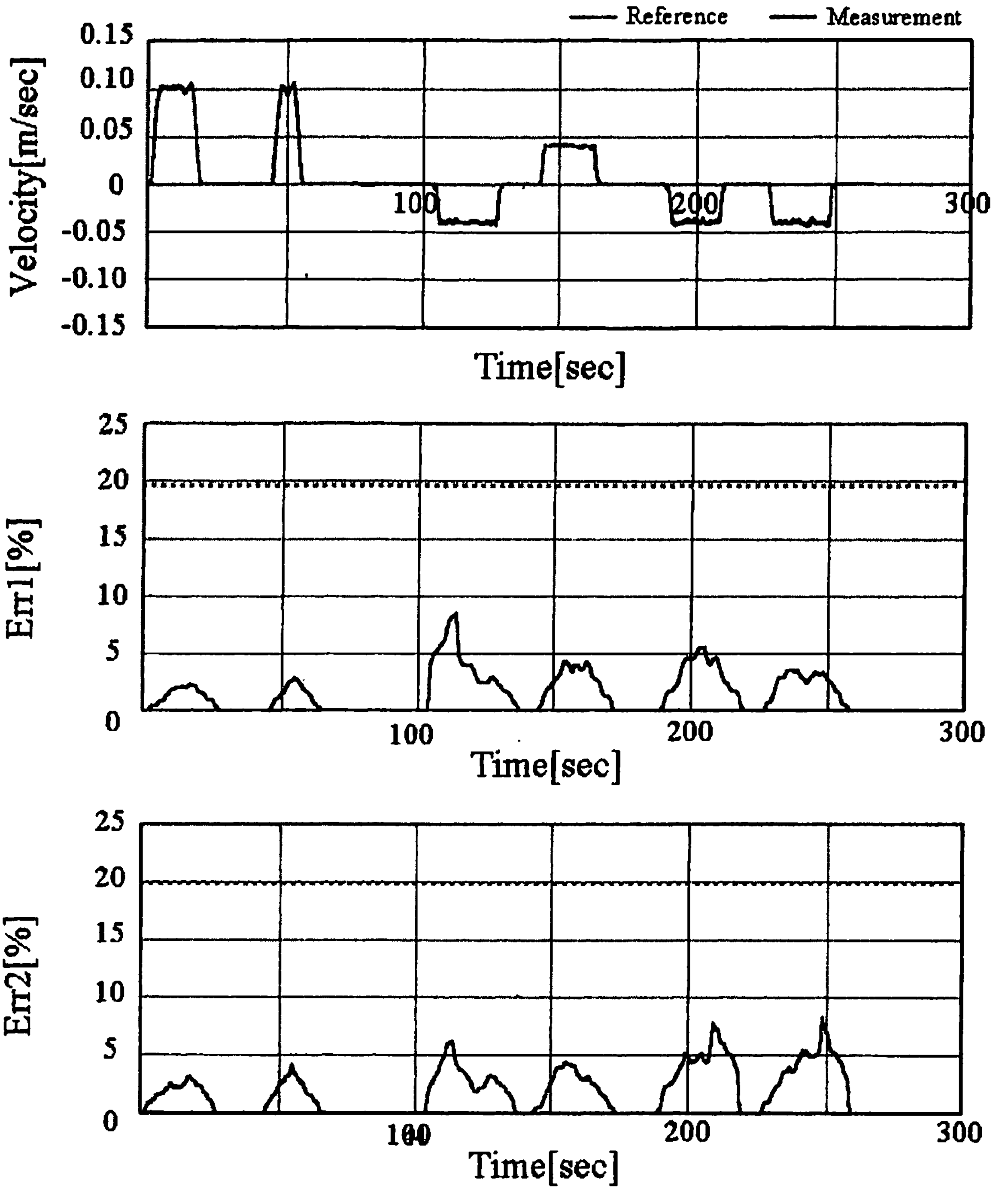


Figure 21. Velocity, Err1 and Err2 of the X_r axis during the task (Y_r axis at fault).

and also classified them into three categories. A model-based algorithm is utilized to diagnose each module in the system. Also, additional sensors were installed on an omnidirectional mobile robot ZEN-450 and an internal sensory system that uses the sensors was constructed. As examples, we examined diagnostic experiments on the robot's driving and power supply systems. For efficient diagnosis, we classified the diagnostic process into the following three stages: before, during and after operation. Also, we focussed on the functions of the autonomous robot

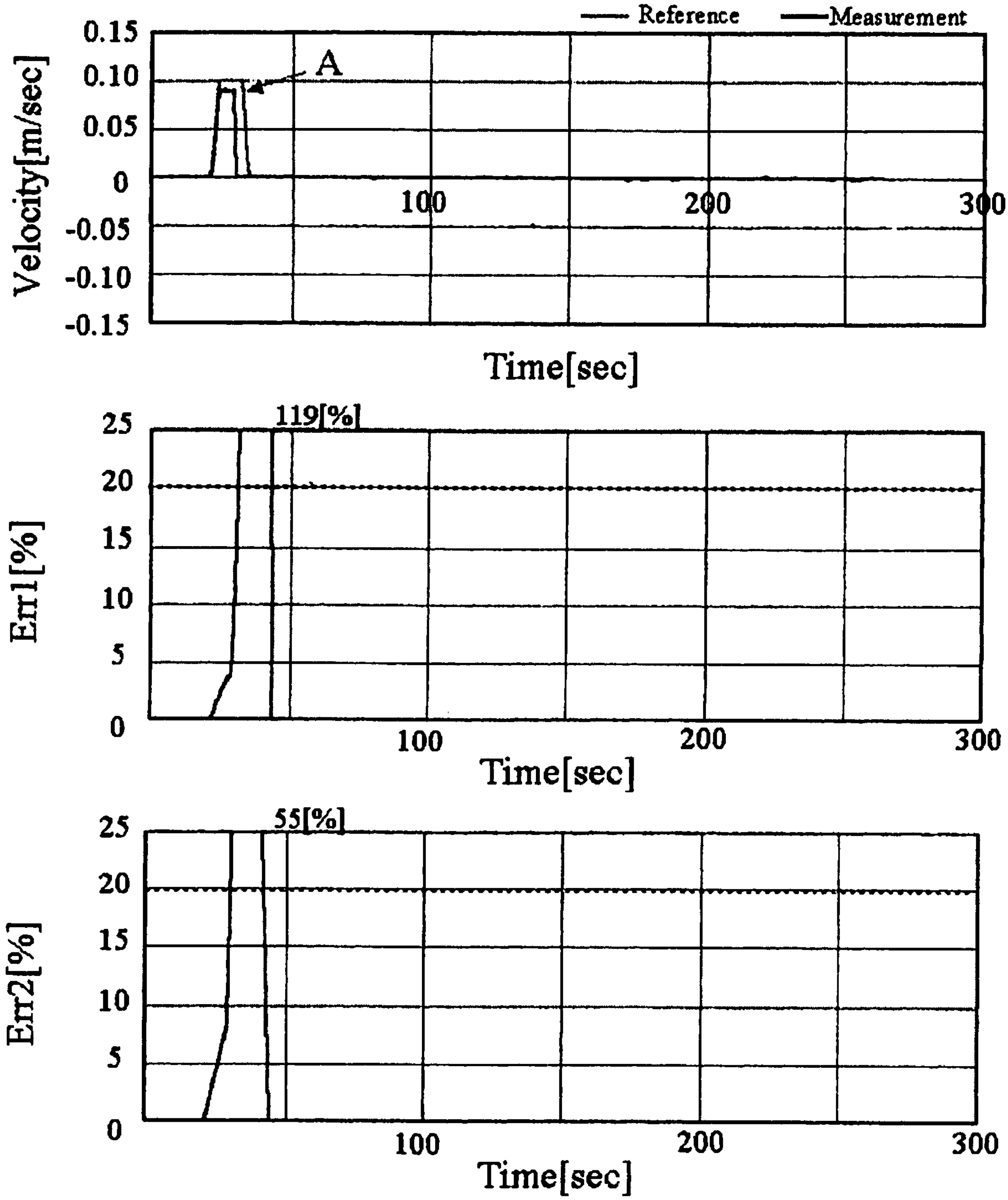


Figure 22. Velocity, Err1 and Err2 of the Y_r axis during the task (Y_r axis at fault).

and introduced fault levels for classifying the internal condition. Utilizing these levels, we divided the faulty condition into three stages and also set the measures used to cope at each level. We attempted experiments using our proposed system and a real mobile robot. From the result, we confirmed that our method works effectively.

In our future work we intend to add external sensors to our present system and to advance our self-diagnosis system for extending adaptivity of the mobile robot.

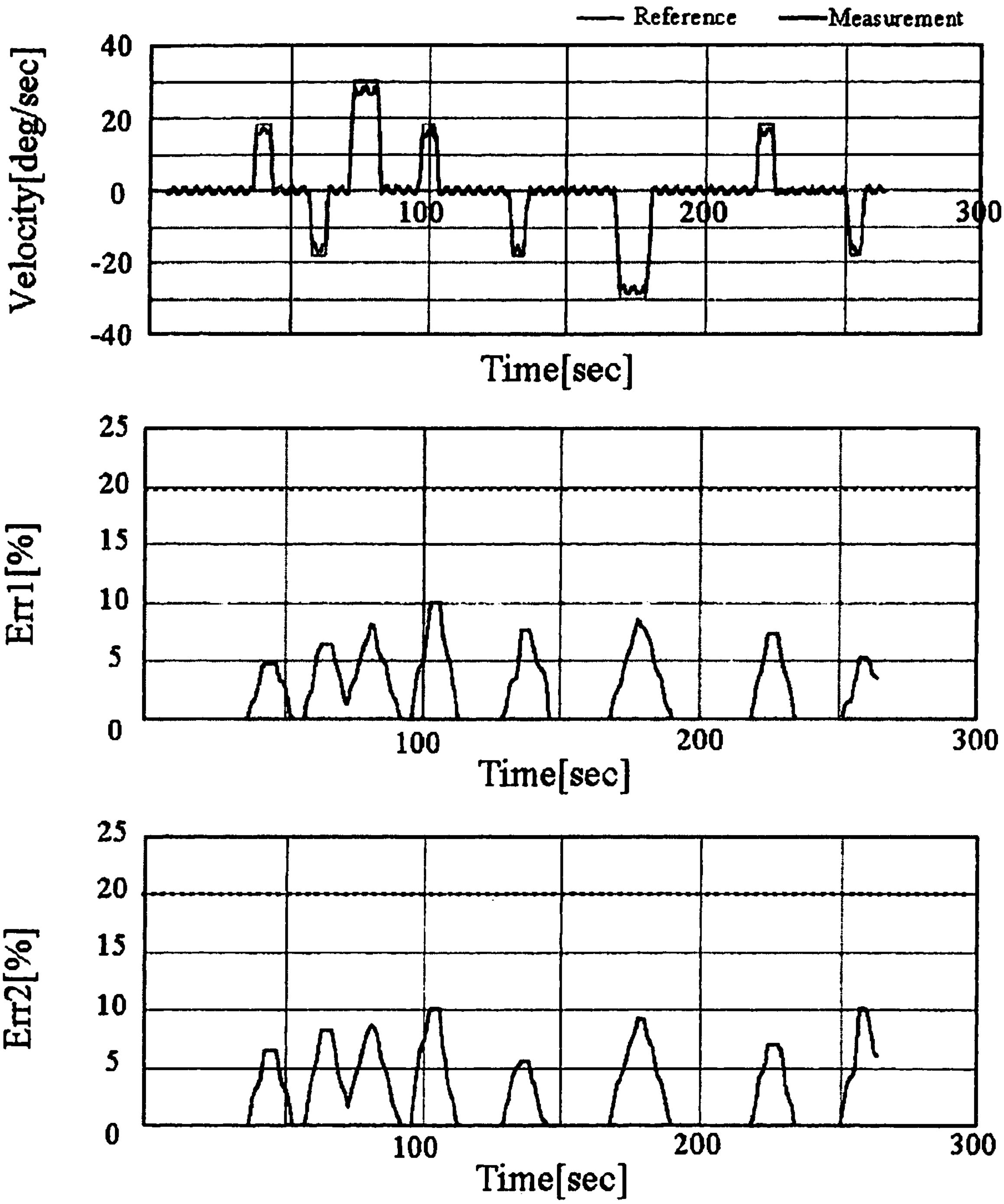


Figure 23. Velocity, Err1 and Err2 of the rotation axis during the task (Y_r axis at fault).

In detail, we must discuss how to detemine the criterion for judging the system condition autonomously when the situation or environment is changed during when the robot is running. Futhermore, a motion planning method based on considering the self-condition or motion in the past record should be developed.

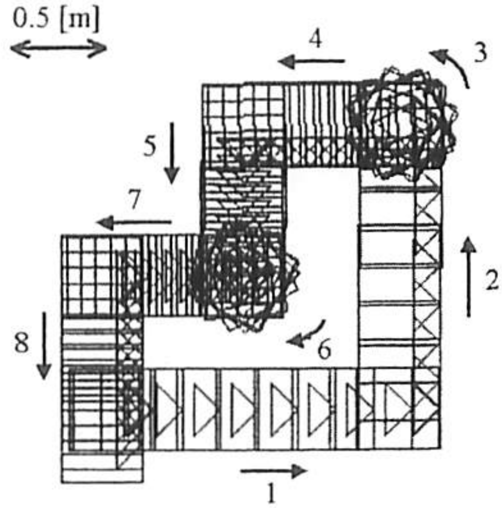


Figure 24. The robot's path under the normal condition.

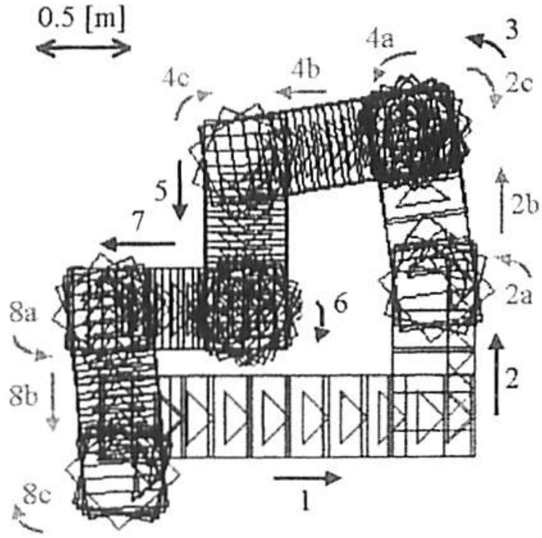


Figure 25. The robot's path under the faulty condition (Y_r axis at fault).

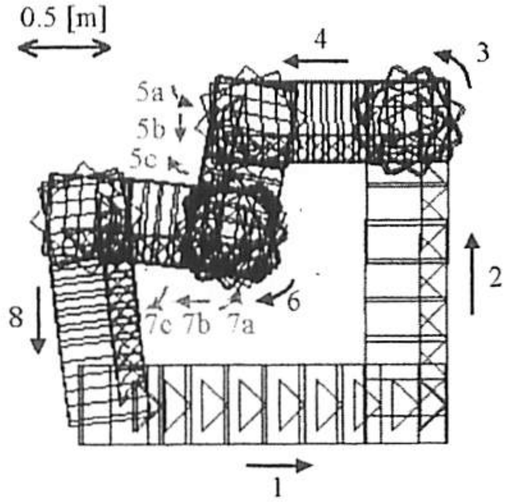


Figure 26. The robot's path under the faulty condition (X_r axis at fault).

REFERENCES

1. H. Inoue, Priority research program on intelligent robotics, *J. Robotics Soc. Japan* **16** (5), 578–585 (1998).
2. R. C. Arkin, *Behavior-Based Robotics*. MIT Press, Cambridge, MA (1998).
3. R. A. Brooks, Intelligent without Reason, *Artif. Intell.* **47**, 139–159 (1991).
4. Special Issue: Intelligent Robotics, *J. SICE* **35** (4), 305–322 (1996).
5. H. Kitano, J. Siekmann and J. G. Carbonell (Eds) *Robocup 97 Robot Soccer World Cup I*. Springer Verlag, Berlin (1998).
6. M. L. Visinsky, I. D. Walker and J. R. Cavallaro, Layered dynamic fault detection and tolerance for robots, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, Vol. 2, pp. 180–187 (1993).
7. S. I. Roumeliotis, G. S. Sukhatme and G. A. Bekey, Sensor fault detection and identification in a mobile robot, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, BC, Vol. 2, pp. 1383–1388 (1998).
8. C. Cocca, D. Cox and D. Tesar, Failure recovery in redundant serial manipulators using nonlinear programming, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, Vol. 2, pp. 855–860 (1999).
9. J.-H. Shin and J.-J. Lee, Fault detection and robust fault recovery control for robot manipulators with actuator failures, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, Vol. 2, pp. 861–866 (1999).
10. Z. Vukic, H. Ozbolt and D. Pavlekovic, Improving fault handling in marine vehicle course-keeping systems, *IEEE Robotics Automat. Mag.* (June), 39–52 (1999).
11. M. Takai, T. Fujii and T. Ura, A model based diagnosis system for autonomous underwater vehicles using artificial neural network, in: *Proc. 9th Int. Symp. on Unmanned Untethered Submersible Technology*, Durham, NC, pp. 243–252 (1995).
12. R. Iserman and D. Fussel, Supervision and diagnosis of technical systems, in: *Workshop on 5th Int. Conf. on Control, Automation, Robotics and Vision*, Mandarin, Singapore (1998).
13. H. Asama, M. Sato, L. Bogoni, H. Kaetsu and I. Endo, Development of an omni-directional mobile robot with 3 DOF decoupling drive mechanism, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Vol. 2, pp. 1925–1930 (1995).
14. S. Okina, K. Kawabata, T. Fujii, Y. Kunii, H. Asama and I. Endo, Self-diagnosis system of an autonomous mobile robot using sensory information, *J. Robotics Mechatronics* **12** (2), 72–77 (2000).
15. S. Okina, K. Kawabata, T. Fujii, Y. Kunii, H. Asama and I. Endo, A study of self-diagnosis system for an autonomous mobile robot, *Advanced Robotics* **14** (5), 339–341 (2000).
16. K. Kawabata, S. Okina, T. Fujii and H. Asama, A study of self-diagnosis system for an autonomous mobile robot: coping with self-condition using internal sensory information, in: *Proc. 27th Ann. Conf. of the IEEE Industrial Electronics Society*, Denver, CO, pp. 381–386 (2001).

ABOUT THE AUTHOR

Kuniaki Kawabata received the BE, ME and PhD degrees in Electrical Engineering from Hosei University in 1992, 1994, and 1997, respectively. He joined the Biochemical Systems Laboratory at RIKEN (The Institute of Physical and Chemical Research) as a Special Postdoctral Researcher from 1997 to 2000. In 2000, he joined the Advanced Engineering Center at RIKEN as a research scientist. In 2002, he joined the Distributed Adaptive Robotics Research Unit at RIKEN. His research interests cover distributed autonomous robotic systems, emergent systems and adaptive mechanisms. He is a member of RSJ, SICE, IEEJ, JSAI and

IEEE.

Shinnosuke Okina received the BE and ME degrees in Electrical Engineering from Chuo University in 1999 and 2001, respectively. He was interested in research on the self-diagnosis of autonomous mobile robots during his university and graduated school period. From 2001, he joined NEC Corp. as a software engineer in the middleware division.

Teruo Fujii received the BE, ME and PhD degrees in Naval Architecture and Ocean Engineering from the University of Tokyo in 1988, 1990 and 1993, respectively. He served as an Associate Professor of Globe Engineering at the Institute of Industrial Science (IIS), University of Tokyo from 1993 to 1995. He was with the Biochemical Systems Laboratory at the Institute of Physical and Chemical Research from 1995 to 1999. In 1999, he joined the IIS again where he is currently an Associate Professor at the Underwater Technology Research Center. His research interests cover microbiochemical systems. He is a member of the AAAS, ACS and IEEE.

Hajime Asama received the MS and DS degrees in Engineering from the Department of Precision Machinery Engineering, University of Tokyo, in 1984 and 1989, respectively. In 1986, he joined the Institute of Physical and Chemical Research and became the Professor of Research into Artifacts, Center for Engineering, University of Tokyo. He received the JSME Robotics and Mechatronics Division Robotics and Mechatronics Award in 1995, JSME Robotics and Mechatronics Division Robotics and Mechatronics Academic Achievement Award in 2000, JIDPO Good Design Award in the New Frontier Design Category in 2002, etc. He also received RoboCup-97 Engineering Challenge Award and RoboCup-98 Japan Open JSAI award as a member of the UTTORI United Team. He was an Editor of *Distributed Autonomous Robotics Systems*, vols 2 and 5, which were published by Springer-Verlag, Tokyo. He is a member of the IEEE, RSJ, JSME, SICE, New York Academy of Science, etc. His main interests are distributed autonomous robotic systems, cooperation of multiple autonomous mobile robots, emergent robotic systems and intelligent data carrier systems.