

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Model-Based Fault Detection and Isolation for a Powered Wheelchair

Masafumi Hashimoto, Fumihiro Itaba and Kazuhiko Takahashi
*Doshisha University
Japan*

1. Introduction

In mobile robotics and vehicle automation, the demand for fault detection and isolation (FDI) of sensors, actuators, and system components is growing significantly to assure system reliability and safety (Carlson, 2004; Petterson, 2005). FDI techniques are broadly classified into hardware redundancy and analytical redundancy approaches (Simani et al., 2003). The typical hardware redundancy approach for FDI is based on majority voting logic; outputs from redundant sensors are compared with each other, and the sensor whose output does not match the others is determined to be the faulty one. The major problem with hardware redundancy is the extra equipment required.

Unlike hardware redundancy, analytical redundancy requires no additional hardware components, and its algorithm can be implemented using software on a computer. Analytical redundancy makes use of mathematical models of the system under investigation, and it is therefore often referred to as a model-based method.

The typical model-based FDI is based on Bayesian filtering. The traditional FDI method uses the multiple-model adaptive estimation (MMAE) algorithm (*i.e.*, a bank of Kalman filters) (Magill, 1965); its application to FDI of mechanical failures (flat tires) and internal sensor faults has been presented (Roumeliotis et al., 1998a & 1998b). The MMAE algorithm is a noninteracting multiple-model algorithm. Therefore, it is not suitable for situations where system faults occur abruptly. To make the MMAE algorithm a better fit for such situations, various ad hoc techniques have been investigated (Maybeck & Hanlon, 1998; Goel et al., 2000). To cope with the weakness of the MMAE-based approach, the interacting multiple-model (IMM) estimator (Blom & Bar-shalom, 1988; Zhang & Li, 1998) was applied to sensor FDI in mobile robots (Hashimoto et al., 2001 & 2003). Another typical model-based approach for FDI is the use of particle filters (Verma et al., 2004; Duan et al., 2006); this approach is suitable for FDI in nonlinear systems or systems with multimodal distribution or non-Gaussian noise. However, the computational cost increases rapidly with the number of state dimensions, even though there are means for increasing efficiency.

The major problems encountered with the model-based method are due to the use of imperfect models. In practical systems, model plant mismatches always exist; unexpected faults and disturbances also occur. These conditions may cause false alarms or missed alarms in FDI. Thus, model-based FDI needs to be made more robust (Simani et al., 2003).

In our previous paper, we presented model-based FDI for mobile robots and powered wheelchairs (Hashimoto et al., 2001; 2003 & 2007); external sensors and actuators were assumed to be always fault free, and only the faults of internal sensors were handled. The IMM estimator and Kalman filter were applied to FDI of the internal sensors.

In this chapter, we present model-based FDI for a powered wheelchair handling faults of both the internal and external sensors, as well as actuators. This chapter is organized as follows: Section 2 overviews our experimental wheelchair. Sections 3 and 4 present the FDI method for internal sensors and actuators, and Section 5 presents the FDI method for external sensors. In Section 6, we describe experiments conducted in a typical indoor environment to evaluate the FDI method, followed by our conclusions.

2. Experimental Wheelchair

Figure 1 shows the powered wheelchair used in the experiments; it has wheel units on the left and right sides to achieve differential steering. The wheel units each consist of a wheel motor and a wheel resolver. The wheel resolver measures the wheel velocity. A yaw-rate gyro is attached to the chassis of the wheelchair to sense the turn velocity. A laser range sensor (LRS) is mounted at the front side of the wheelchair to sense environments. It takes a laser scan image, which is represented by a sequence of distance samples in the horizontal half plain. The angular resolution of the LRS is 0.5 [deg], and one scan image contains 361 distance samples.

We handle the faults of three internal sensors (two wheel resolvers and one gyro), one external sensor (LRS) and two wheel motors.

3. Detection and Isolation of Internal Sensor and Wheel Motor Hard Faults

3.1 Internal Sensor FDI

First, we consider the hard faults of the wheel resolver and the gyro—the sensor fails completely, and the sensor output is stuck at a constant value.

A dynamic model related to the left and right wheel units is derived by



Fig. 1. Experimental wheelchair.

$$v_i(t+1) = A_i v_i(t) + B_i u_i(t) + \Delta v_i(t) \quad (1)$$

where $i = \text{left (wheel) and right (wheel)}$. v_i is the wheel velocity and Δv_i is the disturbance. u_i is the input of the wheel unit. A_i and B_i are constants.

We assume that the wheelchair moves stably at an almost constant velocity through fault-tolerant operation, even though sensor faults have occurred. The dynamic model related to the turning motion of the wheelchair is then assumed to be

$$\dot{\phi}(t+1) = \dot{\phi}(t) + \Delta \dot{\phi}(t) \quad (2)$$

where $\dot{\phi}$ is the turning velocity and $\Delta \dot{\phi}$ is the disturbance.

When the sensor output of the hard fault is assumed to be stuck at zero, the measurement model related to the wheel unit is

$$\text{Fault-free model } (m_0): z_i(t) = v_i(t) + \Delta z_{i0}(t) \quad (3)$$

$$\text{Hard fault model } (m_1): z_i(t) = 0 + \Delta z_{i1}(t) \quad (4)$$

where z_i is the output of the wheel resolver. Δz_{i0} and Δz_{i1} are the sensor noises.

Because the wheel resolver is assumed to be stuck at zero on the hard fault, the measurement is modeled by zeroing out the kinematic function, as in Eq. (4). If it is stuck at a nonzero value on the hard fault, the measurement is modeled by $z_i(t) = C + \Delta z_{i1}(t)$, where C is the stuck value.

The gyro output is also modeled by

$$\text{Fault-free model } (m_0): z_{gyro}(t) = \dot{\phi}(t) + \Delta z_{gyro0}(t) \quad (5)$$

$$\text{Hard fault model } (m_1): z_{gyro}(t) = 0 + \Delta z_{gyro1}(t) \quad (6)$$

We assume that the disturbances and the sensor noises are both zero means and white Gaussian sequences, and we estimate the mode probability based on Eqs. (1)–(6) via the IMM estimator (Zhang & Li, 1998; Hashimoto et al., 2001). The mode probability provides an indication of FDI of the internal sensor; for example, we consider FDI of the gyro and denote the mode probability estimates of fault free and hard fault by μ_0 and μ_1 , respectively. The fault decision is made based on simple logic: If $\mu_0 > \mu_1$, then the gyro is fault free; else, a hard fault occurs.

The stop condition of the wheelchair results in wheel resolver FDI making an incorrect fault decision, because it makes the sensor output zero. In addition, the forward motion and stop condition of the wheelchair make it difficult to discriminate between fault-free and hard fault conditions of the gyro, because in both these conditions, the gyro output is almost zero. To cope with these problems, we switch the following model sets, according to the motion of the wheelchair:

$$\left. \begin{aligned} MG_1 &= \{m_0\} \\ MG_2 &= \{m_0, m_1\} \end{aligned} \right\} \quad (7)$$

In FDI for the wheel resolver, when the wheelchair is considered to stop (*i.e.*, $u_{left} \approx 0$ and $u_{right} \approx 0$), model set MG_1 is selected; else, MG_2 is selected. In FDI for the gyro, when the wheelchair stops (*i.e.*, $u_{left} \approx 0$ and $u_{right} \approx 0$) or moves straight (*i.e.*, $u_{left} \approx u_{right}$), model set MG_1 is selected; else, MG_2 is selected. The model-set-switching-based FDI algorithm is formulated based on a variable-structure-interacting-multimodel (VSIMM) estimator (Li, 2000; Hashimoto et al., 2003).

3.2 Fault Isolation of the Wheel Resolver and Wheel Motor

We consider a hard fault of the wheel motor as well as that of the wheel resolver. For a hard fault of the wheel motor, wheel rotation might stop completely. The hard fault of the wheel motor and that of the wheel resolver make the wheel-resolver output zero; thus, the IMM-based FDI mentioned in 3.1 cannot identify the faulty component (the wheel resolver or the wheel motor). In this subsection, when the IMM-based FDI detects the fault of the wheel unit, we identify the faulty component.

A hard fault of the wheel motor and that of the wheel resolver cause a difference in the turning velocity of the wheelchair. When the hard fault of the wheel motor occurs, the turning velocity sensed by the gyro is almost the same as that calculated by the wheel resolver. On the other hand, when a hard fault in the wheel resolver occurs, the turning velocity sensed by the gyro is much different from that calculated by the wheel resolver. We therefore perform fault identification based on the following logic:

$$\text{If } \left| z_{gyro} - \left(\frac{z_{right} - z_{left}}{b} \right) \right| < h \quad (8)$$

then the wheel motor is faulty;
else, the wheel resolver is faulty.

where b is the tread length of the wheelchair. h is the threshold; in our experiment, we set h as 0.01 [rad/s].

In principle, when a gyro fault occurs, it is impossible to discriminate between faults of the wheel resolver and the motor. For simultaneous faults of the wheel resolver and the motor, our FDI decides that the wheel resolver is faulty.

4. Detection and Isolation of Soft Faults of the Internal Sensor

4.1 Scan Matching Based Velocity Estimate

In this section, we handle soft faults of the internal sensors that appear as changes in the sensor gain. A change in the sensor gain is caused by not only sensor faults but also various system component failures and environmental interactions. For example, a flat tire, wheel slippage, and soft wheel motor faults change the gain of the wheel resolver; therefore, their conditions are considered to be soft faults of the wheel resolver.

In principle, it is necessary to determine the accurate linear and turning velocity of the wheelchair to detect soft faults of the internal sensors. We make use of laser scan matching with the fault-free LRS to estimate the velocity of the wheelchair. Our laser scan matching is based on typical point-to-point scan matching using an iterative closest point (ICP) algorithm (Besl & McKay, 1992; Lu & Milios, 1997). In ICP-algorithm-based scan matching,

the distance samples in the new laser scan are matched with those in the previous scan, so that the sum of squared distance can be minimized, and the velocity of the wheelchair is estimated.

Before scan matching, we preprocess the scan image to remove erroneous distance samples (Diosi & Kleeman, 2007). This preprocessing consists of noise reduction, segmentation, and interpolation. The noise reduction is based on a median filter, and it replaces outliers of distance samples with suitable samples. The segmentation prevents incorrect interpolation between two disjoint distances. The interpolation allows estimation of distance samples at every angular resolution of 0.5 [deg] of the LRS.

We determine the linear and turning velocities, $\dot{x} = (\dot{x}, \dot{y}, \dot{\phi})^T$, of the wheelchair by the weighted least-squares method; the cost function is

$$J = \sum_{i=1}^{361} w_i \{q_j - (R p_i + T)\}^2 \quad (9)$$

Here, $p_i = (p_{ix}, p_{iy})^T$, where $i = 1, 2, \dots, 361$, is the distance sample in the current scan, while $q_j = (q_{jx}, q_{jy})^T$, where $j = 1, 2, \dots, 361$, is that in the previous scan. Each sample p_i corresponds with the minimum distance sample q_j of all samples in the previous scan. w_i is the weight. R and T are the rotational matrix and the translational vector, respectively, given by

$$R = \begin{pmatrix} \cos \tau \dot{\phi} & -\sin \tau \dot{\phi} \\ \sin \tau \dot{\phi} & \cos \tau \dot{\phi} \end{pmatrix} \text{ and } T = \begin{pmatrix} \tau \dot{x} \\ \tau \dot{y} \end{pmatrix}. \quad \tau \text{ is the scanning period of the LRS; in our experimental}$$

system, $\tau = 0.1$ [s].

The correspondence errors of distance samples of two successive scans affect the velocity estimate. To reduce the effect, we determine the weight w_i according to the errors between the correspondence points:

$$w_i = \frac{1}{1 + \frac{d_{ij}}{C}} \quad (10)$$

where d_{ij} is the distance error between the new and previous scans. C is a constant; in our experiments, $C = 0.1$.

From Eq. (9), the iterative least-squares method updates the wheelchair velocity $\dot{x}^{(m-1)}$ as follows:

$$\dot{x}^{(m)} = \dot{x}^{(m-1)} + (H^T W H)^{-1} H^T W (q - p) \quad (11)$$

where $p = (p_1^T, p_2^T, \dots, p_{361}^T)^T$, $q = (q_1^T, q_2^T, \dots, q_{361}^T)^T$, $W = \text{diag}(w_1, w_2, \dots, w_{361})$, and $H = \partial p / \partial \dot{x}$. In our experiments, the maximum iterative number of the velocity update is 20. The velocity estimate calculated by the iterative least-squares method is noisy, and it is smoothed using Kalman filter. One disadvantage of the ICP algorithm is that it converges slowly. To accelerate the convergence, we use dead-reckoning information as the initial velocity $\dot{x}^{(0)}$.

4.2 FDI Algorithm

A soft fault appears as a change in the sensor gain, which deviates it from the fault-free condition; thus, we detect a soft fault by estimating the sensor gain. Sensor gain α is defined by

$$\alpha = \frac{\text{Actual output of the sensor}}{\text{Sensor output in the fault - free condition}} \quad (12)$$

From this definition, when a sensor is fault free, the sensor gain equals unity.

We estimate the sensor gain based on the wheelchair velocity estimated by laser scan matching. We assume that the state model of the sensor gain is given by

$$\alpha_i(t+1) = \alpha_i(t) + \Delta\alpha_i(t) \quad (13)$$

where $i = \text{left (wheel), right (wheel), and gyro}$.

From the inverse kinematics of the wheelchair velocity, the measurement model is derived:

$$\begin{pmatrix} z_{\text{left}}(t) \\ z_{\text{right}}(t) \\ z_{\text{gyro}}(t) \end{pmatrix} = \begin{pmatrix} \alpha_{\text{left}}(t) \left(\sqrt{\dot{x}^{(M)2}(t) + \dot{y}^{(M)2}(t)} + \frac{b\dot{\phi}^{(M)}(t)}{2} \right) \\ \alpha_{\text{right}}(t) \left(\sqrt{\dot{x}^{(M)2}(t) + \dot{y}^{(M)2}(t)} - \frac{b\dot{\phi}^{(M)}(t)}{2} \right) \\ \alpha_{\text{gyro}}(t)\dot{\phi}^{(M)}(t) \end{pmatrix} + \begin{pmatrix} \Delta z_{\text{left}}(t) \\ \Delta z_{\text{right}}(t) \\ \Delta z_{\text{gyro}}(t) \end{pmatrix} \quad (14)$$

where $(\dot{x}^{(M)}, \dot{y}^{(M)})$ and $\dot{\phi}^{(M)}$ are the linear and turning velocities of the wheelchair, which are estimated by laser scan matching.

The sensor gain is estimated based on Eqs. (13) and (14) via Kalman filter. A small change in the gain estimate always occurs due to wheel slippage, sensor noise, and so on. Thus, if the estimate of the sensor gain is far away from one (e.g., 0.7), the sensor is determined to be faulty. The stop condition of the wheelchair yields an incorrect estimate of the sensor gain, because it makes the right-handed side of Eq. (14) small. When the wheelchair moves straight, the turning velocity is very small, making it impossible to estimate the sensor gain. To better estimate the sensor gain, the following heuristic rules are incorporated into the fault decision making:

Rule 1: If the control inputs of the two drive wheels are very small (i.e., $u_{\text{left}} \approx 0$ and $u_{\text{right}} \approx 0$), the gain estimates of all internal sensors in the previous cycle are held for the current cycle.

Rule 2: If the control inputs of the two drive wheels are almost the same (i.e., $u_{\text{left}} \approx u_{\text{right}}$), the gain estimate of the gyro in the previous cycle is held for the current cycle.

Because a hard sensor fault leads to $\alpha = 0$, we can detect the hard fault by estimating the sensor gain. However, the time delay in the hard fault detection by the sensor gain estimate is longer than that in the IMM-based hard fault detection. For this reason, we exploit the IMM estimator for hard fault detection.

5. Fault Detection of the LRS

5.1 Fault Detection Algorithm

An LRS fault can be detected based on an error related to scan matching (Hashimoto et al., 2008a); if the current scan cannot match the previous scan, the LRS is determined to be faulty. For fault detection, we define the following cost function:

$$J_{LRS} = \sum_{i=1}^{361} \{q_i - (\mathbf{R}^{(M)} \mathbf{p}_i + \mathbf{T}^{(M)})\}^2 \quad (15)$$

where $\mathbf{R}^{(M)}$ and $\mathbf{T}^{(M)}$ are the rotational matrix and the translational vector, respectively, which contain the velocity estimate $\dot{\mathbf{x}}^{(M)}$ from laser scan matching.

A fault is determined when $J_{LRS} > h_{LRS}$, where h_{LRS} is a threshold. In detecting an LRS fault, Eq. (9) removes the correspondence points of large errors in two successive scan points. For this reason, we use Eq. (15) for detecting LRS faults. Noisy distance samples of the LRS may cause false detection, even though the LRS is fault free. To cope with this problem, the LRS is determined to be faulty, when $J_{LRS} > h_{LRS}$ is always satisfied in several successive scans.

In featureless environments (*e.g.*, in long corridors), the corresponding pair between distance samples in two successive laser scans does not necessarily represent the same physical point in the environment, and the estimate of the wheel velocity becomes inaccurate, even though the LRS is fault free. To cope with this problem, we evaluate the distance samples by means of the Nieto approach (Nieto et al., 2007) and diagnose the LRS fault only in featured environments.

In dynamic environments, moving objects and backgrounds occluded by the objects cause J_{LRS} to be large, and false detections may occur. To cope with this problem, we detect the moving objects by the occupancy grid method (Hashimoto et al., 2006). As shown in Fig. 2, a grid map is represented in a reference coordinate frame Σ_w ($O_w; X_w Y_w$). The LRS measurements are mapped in Σ_w and marked on the grid map. For each new cell of occupation in the current scan, the corresponding cell in the previous scan is checked. If that cell is marked, the cell in the current scan is considered a stationary object cell; else, it is a moving object cell. An object consisting of many moving object cells is determined as a moving object, and we remove the distance samples related to that moving object and backgrounds occluded by it when calculating the cost function J_{LRS} . In our experiments, the cell size is set as 0.3 [m] \times 0.3 [m]. For mapping the laser image on the grid map, we use the wheelchair velocity estimated with laser scan matching.

5.2 Selection of Initial Velocity

In scan matching, we use dead-reckoning information as the initial velocity $\dot{\mathbf{x}}^{(0)}$ to accelerate the convergence. A fault of an internal sensor thus causes J_{LRS} to be large, even though the LRS is fault free. To cope with this problem, we define the following cost function:

$$J_{DR} = \sum_{i=1}^{361} \{q_i - (\mathbf{R}^{(0)} \mathbf{p}_i + \mathbf{T}^{(0)})\}^2 \quad (16)$$

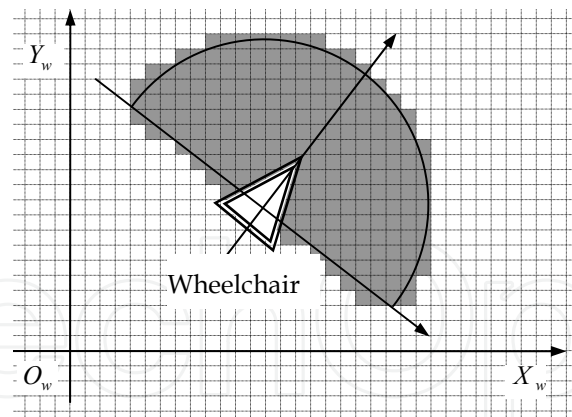


Fig. 2. Grid map; gray cells show the field of view of the LRS.

where $R^{(0)}$ and $T^{(0)}$ are the rotational matrix and the translational vector, respectively, which contain the initial velocity $\dot{x}^{(0)}$ resulting from dead reckoning.

If $J_{DR} < h_{DR}$, where h_{DR} is a threshold, the internal sensor is assumed to be fault free, and we use dead-reckoning information as the initial velocity for scan matching. However, if $J_{DR} \geq h_{DR}$, the internal sensor is assumed to be faulty, and the velocity estimate in the previous scan is used as the initial velocity for scan matching in the new scan.

When abrupt faults occur in the LRS, our algorithm detects the fault. However, incipient faults (a slow degradation of LRS performance), which may occur in the LRS in the real world, would allow scan matching, but estimate the wheelchair velocity inaccurately, causing incorrect fault detection. To cope with this problem, an alternative approach (Sundvall & Jensfelt, 2005) can be considered; the internal sensors estimate the wheelchair velocity by dead reckoning. The LRS also estimates it by scan matching. We thus can consider that the wheelchair is equipped with two different velocity providers (dead reckoning and laser scan matcher). Velocity estimates by fault-free providers yield similar values; whereas those of fault-free and faulty providers are inconsistent. This approach might be suitable for various fault patterns. However, this indicates only that something is going wrong in the two velocity providers; it cannot identify which provider is failing. If a wheelchair is equipped with three or more velocity providers, we can perform majority-voting-logic-based FDI (Hashimoto et al., 2008b). The velocity estimates from the providers are compared, and the provider whose velocity estimate does not match the others is determined to be the faulty one.

6. Experimental Results

6.1 Hard Fault of the Wheel Unit

The wheelchair was moved in a typical indoor environment. We conducted experiments in the following three scenarios:

Scenario 1: All the internal sensors and LRS are fault free.

Scenario 2: A hard fault of the left wheel resolver occurs at 8 [s].

Scenario 3: A hard fault of the left wheel motor occurs at 8 [s].

Figures 3–5 show the experimental results; subfigures (a) and (b) show the velocity of the left and right wheels, respectively. Subfigure (c) shows the turning velocity of the

wheelchair; the bold and broken lines indicate the results calculated by the wheel resolvers and sensed by the gyro, respectively. Subfigures (d)–(f) show the mode probability of the left wheel unit, the right wheel unit, and the gyro, respectively; the bold and broken lines indicate the fault-free and hard fault modes, respectively.

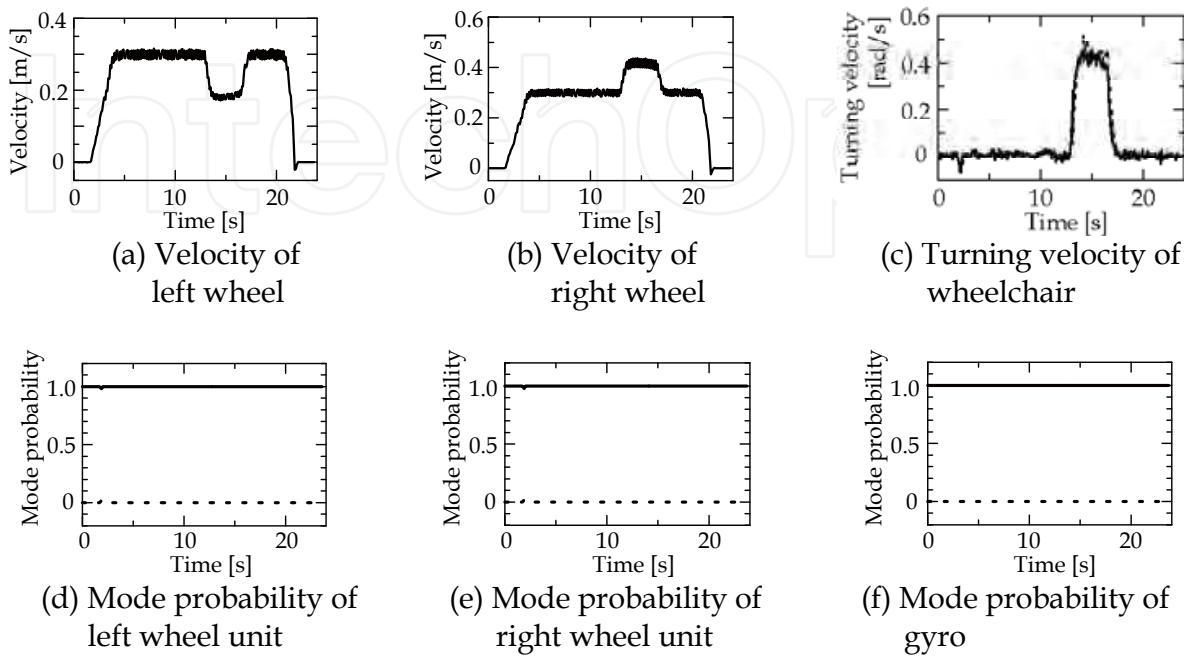


Fig. 3. Experimental result (Scenario 1).

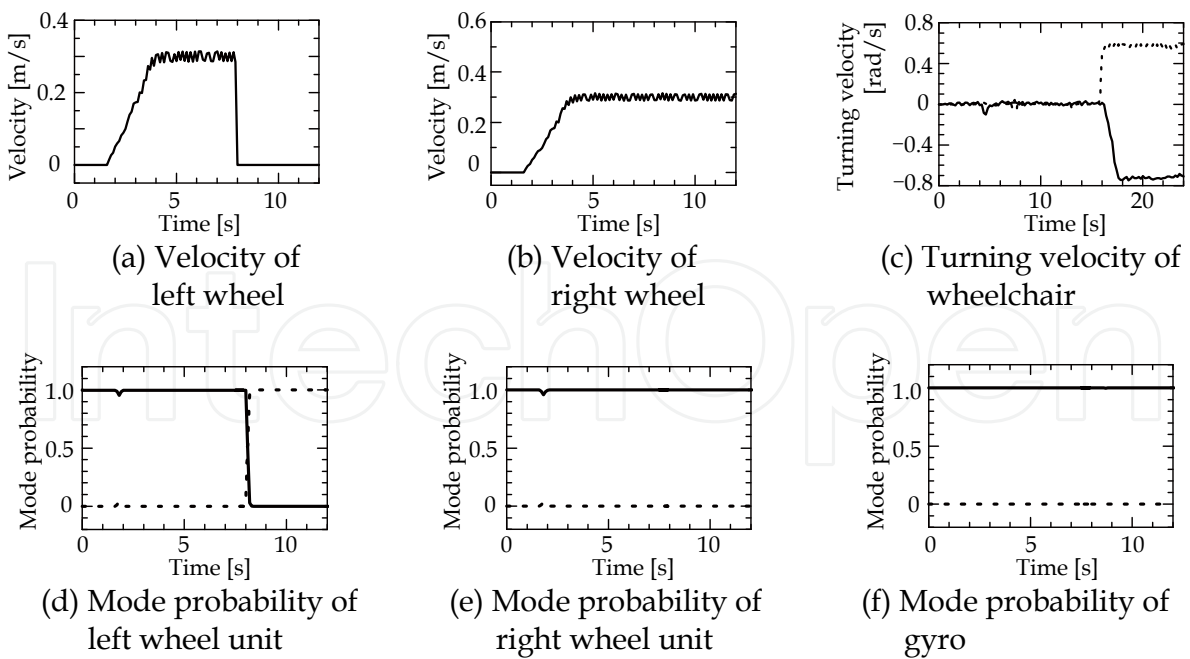


Fig. 4. Experimental result (Scenario 2).

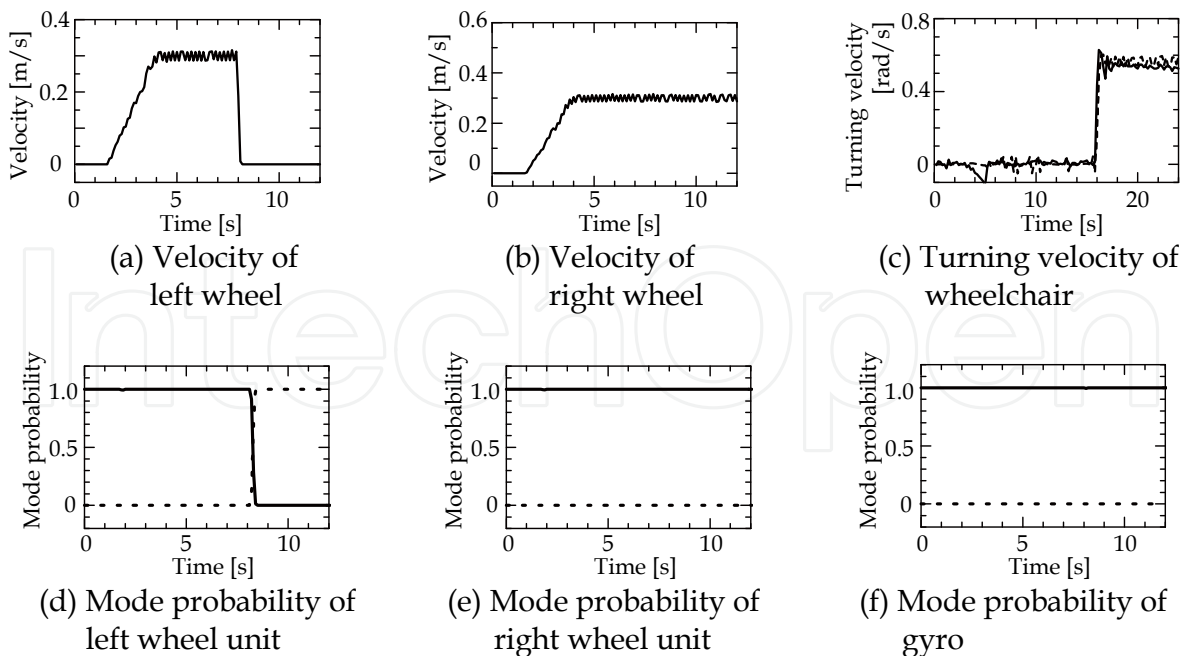


Fig. 5. Experimental result (Scenario 3).

It is clear from subfigure (d) that in scenarios 2 and 3, we can detect a hard fault of the left wheel unit exactly, based on the mode probability. In scenario 2, the turning velocity sensed by the gyro is much different from that calculated by the wheel resolvers. On the other hand, the turning velocity sensed by the gyro is almost the same as that calculated by the wheel resolvers in scenario 3. This difference enables identification of the component (resolver or motor) that fails.

6.2 Soft Fault of Internal and External Sensors

The wheelchair was moved in a typical indoor environment. Figure 6 shows the velocity profile of the wheelchair. Two people walked around in the environment. We set the thresholds related to scan matching at $h_{LRS} = h_{DR} = 15$ [m²]. We conducted experiments in the following four scenarios:

Scenario 4: All the internal sensors and LRS are fault free.

Scenarios 5 and 6: The sensor gain of the right wheel resolver changes abruptly to 0.5 (scenario 5) and 0.8 (scenario 6), respectively, from 1.0 at 21 [s].

Scenarios 7: The sensor gain of the LRS changes abruptly to 0.8 from 1.0 at 30 [s].

Figures 7–10 show the experimental results. Because in scenarios 4, 5, and 6, the cost function J_{LRS} is always less than the threshold h_{LRS} , the LRS is determined to be fault free. Then, we estimate the sensor gain of the internal sensor, and then, based on the estimate, we detect a soft fault of the right wheel resolver, as shown in Figs. 8(c) and 9(c). In scenario 7 (Fig. 10), the cost function J_{DR} is larger than the threshold h_{DR} ; the internal sensor is assumed to be faulty, and the velocity estimate in the previous scan is used as the initial velocity for laser scan matching in the new scan. We also check the value of the cost function J_{LRS} . Because $J_{LRS} > h_{LRS}$, the LRS is determined to be faulty.

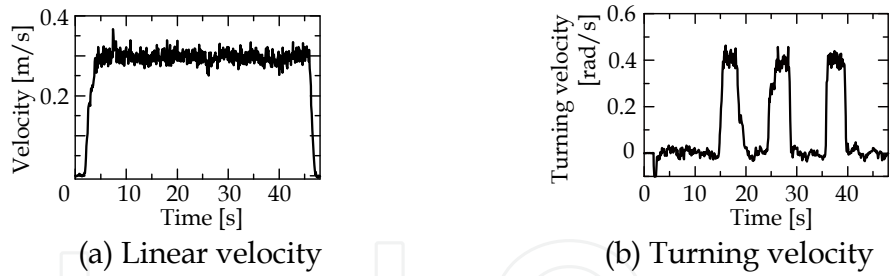


Fig. 6. Profile of wheelchair velocity.

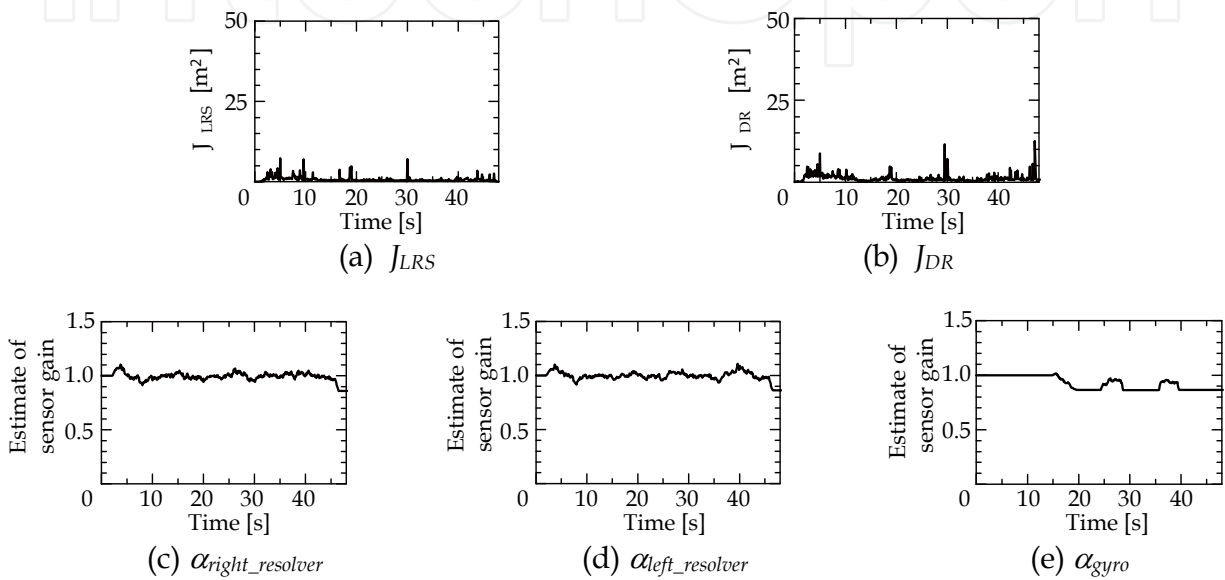


Fig. 7. Experimental result (Scenario 4).

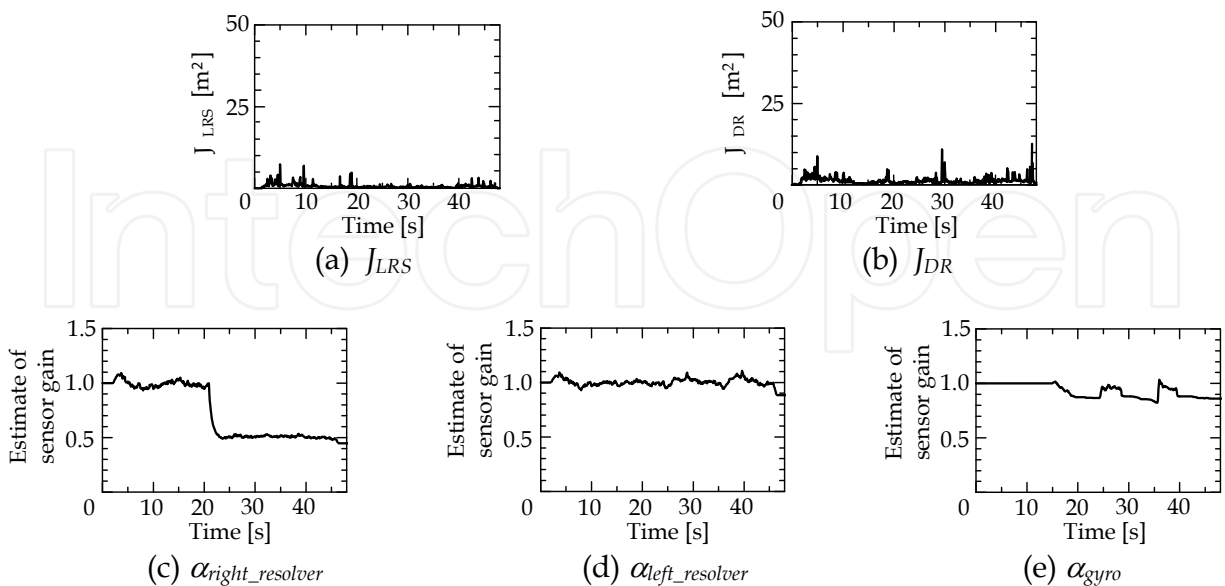


Fig. 8. Experimental result (Scenario 5).

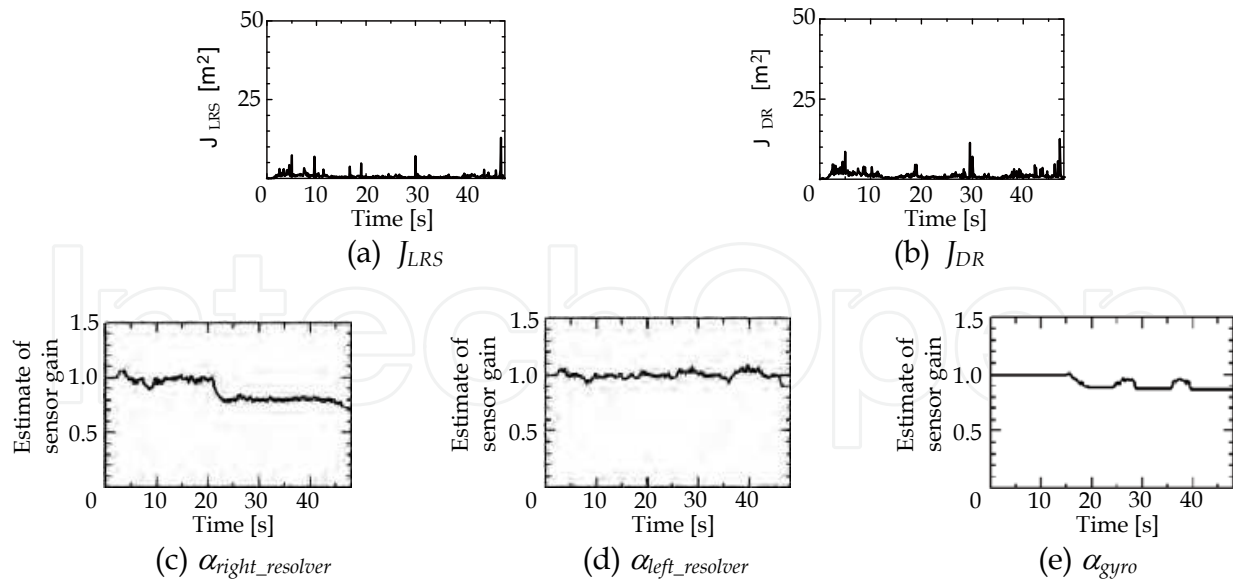


Fig. 9. Experimental result (Scenario 6).

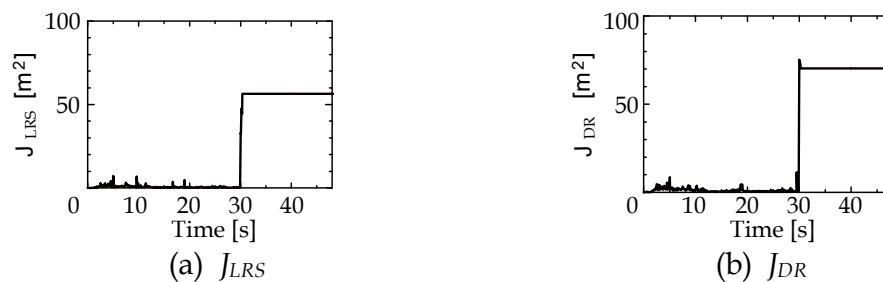


Fig. 10. Experimental result (Scenario 7).

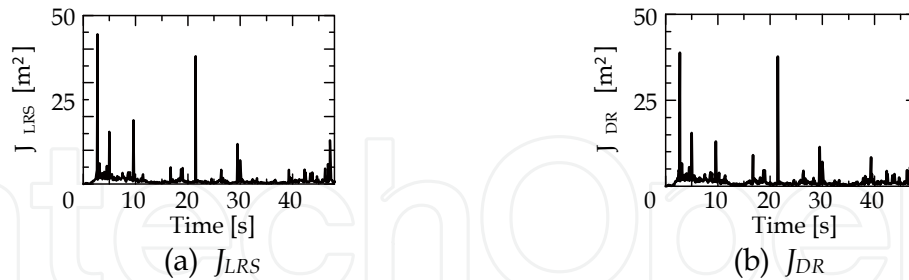


Fig. 11. Experimental result (without moving-object detector).

For comparison, we conducted experiments using the FDI system without the moving-object detector, based on the occupancy grid method. Figure 11 shows the results (the values of the cost functions) in a fault-free condition of both the internal sensors and LRS. The FDI system without the moving-object detector causes false detection of the LRS fault because the values of the cost functions sometimes become larger than the thresholds. On the other hand, as shown in Fig. 7(a) and (b), our FDI system with the moving-object detector, provides the correct detection of the sensor state.

7. Conclusions

This chapter presented a model-based FDI for wheelchair sensors and actuators. Hard faults of the internal sensors and the wheel motors were detected based on the IMM estimator. Fault isolation of the wheel resolvers and the motors was achieved based on the information of the fault-free gyro. A soft fault of the internal sensors was diagnosed based on the velocity estimate of the wheelchair from laser scan matching, using the fault-free LRS. The LRS fault was detected based on errors related to scan matching.

Abrupt faults of the LRS can be detected by our algorithm. However, incipient faults (a slow degradation of LRS performance), which can occur in the real world, allows scan matching, but estimates the wheelchair velocity inaccurately, and causes incorrect LRS fault detection. Our research effort is directed toward FDI for sensors and actuators in various fault patterns.

8. Acknowledgments

This work was partially supported by the Academic Frontier Research Project on “New Frontier of Biomedical Engineering Research” of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

9. References

- Besl, P. J. & McKay, N. D. (1992). A Method of Registration of 3-D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256.
- Blom, H. A. P. & Bar-Shalom, Y. (1988). The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficient, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 8, pp. 780-783.
- Carlson, J. (2004). *Technical Report for the Safety Security Rescue Research Center, ICRA 2004 Robot Fault Diagnosis Workshop*, <http://www.csee.usf.edu/~jcarlsol/papers/FaultDiagnosisWorkshopReport.pdf>.
- Diosi, A. & Kleeman, L. (2007). Fast Laser Scan Matching Using Polar Coordinates, *The International Journal of Robotics Research*, Vol. 26, No.10, pp.1125-1153.
- Duan, Z.; Cai, Z. & Yu, J. (2006). Adaptive Particle Filter for Unknown Fault Detection of Wheeled Mobile Robots, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1312-1315.
- Goel, P.; Dedeoglu, G.; Roumeliotis, S.I. & Sukhatme, G.S. (2000). Fault Detection and Identification in a Mobile Robot Using Multiple-Model Estimation and Neural Network, *Proceedings of 2000 IEEE International Conference on Robotics and Automation*, pp.2302-2307.
- Li, X.R. (2000). Engineer's Guide to Variable-Structure Multiple-model Estimation for Tracking, In: *Multitarget-Multisensor Tracking: Applications and Advances volume III*, Bar-Shalom, Y. & Blair, W.D. (Ed.) Artech House, pp.499-567.
- Lu, F. & Milios E. (1997). Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans, *Journal of Intelligent and Robotic Systems*, Vol.20, pp.249-275.

- Hashimoto, M.; Kawashima, H.; Nakagami, T. & Oba, F. (2001). Sensor Fault Detection and Identification in Dead-Reckoning System of Mobile Robot: Interacting Multiple Model Approach, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Hashimoto, M.; Kawashima, H. & Oba, F. (2003). A Multi-Model Based Fault Detection and Diagnosis of Internal Sensor for Mobile Robot, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Hashimoto, M.; Ogata, S.; Oba, F. & Murayama, T. (2006). A Laser Based Multi-Target Tracking for Mobile Robot, *Intelligent Autonomous Systems 9*, pp.135-144.
- Hashimoto, M.; Watanabe, S. & Takahashi, K. (2007). Sensor Fault Detection and Isolation for a Powered Wheelchair, *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*.
- Hashimoto, M.; Itaba, F. & Takahashi, K. (2008a). Fault Diagnosis of Internal and External Sensors with Scan Matching for Mobile Robot, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp.3096-3101.
- Hashimoto, M.; Kitade, R.; Itaba, F. & Takahashi, K. (2008b). Voting Based Fault Isolation of In-Vehicle Multi-Sensors, *Proceedings of International Conference on Instrumentation, Control and Information Technology*.
- Magill, D.T. (1965). Optimal Adaptive Estimation of Sampled Stochastic Processes, *IEEE Transactions on Automatic Control*, Vol. AC-10, pp. 434-439.
- Maybeck, P. S. & Hanlon, P. D. (1998). Performance Enhancement of a Multiple Model Adaptive Estimator, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 31, pp. 1240-1254.
- Nieto, J.; Bailey, T. & Nebot, E. (2007). Recursive Scan-Matching SLAM, *Robotics and Autonomous Systems*, Vol. 55, pp. 39-49.
- Petterson, O. (2005). Execution Monitoring in Robotics: A Survey, *Robotics and Autonomous Systems*, Vol.53, pp.73-88.
- Roumeliotis, S. I. ; Sukhatime, G. S. & Bekey, G.A. (1998a). Fault Detection and Identification in a Mobile Robot using Multiple-Model Estimation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2223-2228.
- Roumeliotis, S. I.; Sukhatime, G. S. & Bekey, G.A. (1998b). Sensor Fault Detection and Identification in a Mobile Robot," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1383-1388.
- Simani, S.; Fantuzzi, C. & Patton, R.J. (2003). *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, Springer.
- Sundvall, P. & Jensfelt, P. (2005). Fault Detection for Mobile Robots Using Redundant Positioning Systems, *Proceedings of IEEE International Conference on Robotics and Automation*.
- Verma, V.; Gordon, G.; Simmons, R. & Thrun, S. (2004). Real-Time Fault Diagnosis, *IEEE Robotics and Automation Magazine*, pp. 56-66.
- Zhang, Y. & Li, X.R. (1998). Detection and Diagnosis of Sensor and Actuator Failures Using IMM Estimator, *IEEE Transactions on Aerospace and Electronic Systems*, Vol.34. No.4, pp. 1293-1313.



Mechatronic Systems Applications

Edited by Annalisa Milella Donato Di Paola and Grazia Cicirelli

ISBN 978-953-307-040-7

Hard cover, 352 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

Mechatronics, the synergistic blend of mechanics, electronics, and computer science, has evolved over the past twenty five years, leading to a novel stage of engineering design. By integrating the best design practices with the most advanced technologies, mechatronics aims at realizing high-quality products, guaranteeing at the same time a substantial reduction of time and costs of manufacturing. Mechatronic systems are manifold and range from machine components, motion generators, and power producing machines to more complex devices, such as robotic systems and transportation vehicles. With its twenty chapters, which collect contributions from many researchers worldwide, this book provides an excellent survey of recent work in the field of mechatronics with applications in various fields, like robotics, medical and assistive technology, human-machine interaction, unmanned vehicles, manufacturing, and education. We would like to thank all the authors who have invested a great deal of time to write such interesting chapters, which we are sure will be valuable to the readers. Chapters 1 to 6 deal with applications of mechatronics for the development of robotic systems. Medical and assistive technologies and human-machine interaction systems are the topic of chapters 7 to 13. Chapters 14 and 15 concern mechatronic systems for autonomous vehicles. Chapters 16-19 deal with mechatronics in manufacturing contexts. Chapter 20 concludes the book, describing a method for the installation of mechatronics education in schools.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Masafumi Hashimoto, Fumihiro Itaba and Kazuhiko Takahashi (2010). Model-Based Fault Detection and Isolation for a Powered Wheelchair, *Mechatronic Systems Applications*, Annalisa Milella Donato Di Paola and Grazia Cicirelli (Ed.), ISBN: 978-953-307-040-7, InTech, Available from:

<http://www.intechopen.com/books/mechatronic-systems-applications/model-based-fault-detection-and-isolation-for-a-powered-wheelchair>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

www.intechopen.com

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821

IntechOpen

IntechOpen

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen