Multi-localization Methods for Fault Diagnosis in Autonomous Mobile Robots Systems

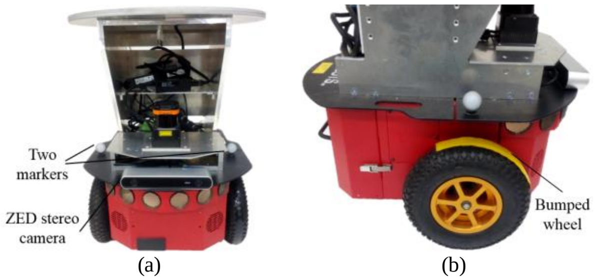
Xiaojun Lu, Angela Faragasso, Yonghoon Ji, Hitoshi Kono, Atsushi Yamashita, and Hajime Asama

***Abstract*— Autonomous mobile robots have been widely employed for many applications in indoor and outdoor environments. Most of these robots have to operate in environments where human intervention is expensive, slow, unreliable or even impossible. It is therefore essential for robots to monitor their behavior to diagnose and address faults before they result in catastrophic failures. In this paper we introduce a new approach to diagnose faults of autonomous mobile robots systems. The proposed methodology firstly computes the poses of the robot by using the onboard stereo camera, the wheels' encoders and the commanded velocities, respectively. Then, the residuals between each pair of the localization methods are used to evaluate the occurrence of faults. Experimental tests, in ideal fault free cases, have been carried out to find a reference threshold for each residual. A bool value is assigned to each residual by comparing it with the respective threshold. The bool values of all residuals are then combined and used to detect and isolate a fault in the robotic system. The pose of ground truth, obtained from a motion capture system, is used here to evaluate the errors of the poses obtained from three other ways and validate their accuracy. Our approach can potentially detect and identify different faults of the robots system. Experimental tests have shown its effectiveness in determine fault on the robot's wheel.**

***Index Terms*— Fault diagnosis, autonomous mobile robot, odometry, visual odometry.**

I . INTRODUCTION

Over the last century, the applications of mobile robots have been extensively expanded and there have been significant activities in the area of robot's reliability and fault tolerance. In order to fulfilling many kinds of tasks, perceiving and operating in unknown environments, mobile robots are provided with advanced sensory mechanisms [1]. Currently, autonomous mobile robots are widely employed in different applications, including indoor and outdoor environments [2]. For instance, many service robots are used in structured environments, e.g. vacuum cleaning for house [3]. On the other hand, in outdoor applications robots have to navigate and accomplish tasks in unstructured and dynamic environments [4]. Fault tolerance is extremely important in remote and hazardous environments, such as planetary exploration, searching and rescuing, mine mapping, nuclear waste cleanup and demining, as it is impossible and expensive for humans to intervene the robot. In these scenarios, failure can have disastrous consequences, although reliability and safety are important issues in almost all applications [5]. Moreover, most current robots are not reliable enough to show the same performance of work as they can in laboratories or indoors [6]. Different faults may occur in the robots system, which results in the failure of completing the planned tasks. For instance, the robots used in decommissioning process, i.e. Fukushima Daiichi Nuclear Power Station, often break down, which makes the decommissioning process very slow and costly [7]. A fault may generate relatively dramatic effects such as a significant decrease in the robot's performance, an aborted mission, damage or crash of the robot or even human's injuries. Detecting and identifying a fault can help a robot to autonomously assess a situation and take appropriate measures; e.g. if the type and severity of the fault is not crucial, the robot could continue its operation uninterrupted and notify the supervisor about the event. Thus, to avoid dangerous situations, it is essential to promptly diagnose the appearance of a fault in mobile robots systems.



(a) (b)

Fig. 1: Pioneer 3-DX robot. a) The ZED stereo camera and the motion capture's markers. b) System fault: a bumped wheel.

In this paper, we propose a novel methodology for fault detection and identification in autonomous mobile robots which uses the poses of the robot obtained from three methods, such as visual odometry, odometry, and control command. These localization results for the robot are compared with each other to generate the respective residuals. Small residuals between each two of the three poses are usual due to sensor noises; however, large residuals can potentially indicate the occurrence of a fault. In order to avoid false detection, which means a fault is detected when there are no faults, evaluation tests in fault-free cases have been carried out to obtain the thresholds, which are then used to determine the appearance of a fault. A motion capture system tracks the movements of the robot, which are used as a ground truth data. The proposed approach has been tested using the Pioneer 3-DX, a commercially available mobile robot, as shown in Fig. 1a). Experimental tests demonstrate that our approach can successfully detect and isolate faults of the robots system. The paper is organized as follows: Section II reviews the current state of the art in techniques for mobile robot's fault diagnosis. The developed methodology is presented in Section III. The experimental setup and results are discussed in Section IV. Conclusions and future work are presented in Section V.

II . BACKGROUND

Based on the importance of the robustness and safe operation of autonomous mobile robots, many researchers have been exploring solutions for robot's fault detection and

*Xiaojun Lu, Angela Faragasso, Yonghoon Ji, Hitoshi Kono, Atsushi Yamashita and Hajime Asama are with Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan (email: luxiaojun, faragasso, ji, kono, yamashita, asama@robot.t.u-tokyo.ac.jp).

fault tolerance [8]. Faults may occur on any subsystems of the robot, e.g. power supply, locomotion, communication and sensors. For autonomous mobile robots, it is extremely important to detect and prevent faults which occur during navigation and locomotion [5]. A fault can be defined as an unexpected change in the system which causes unacceptable deviation from the normal operation [9]. Generally, it is possible to identify two main methodologies for robot's fault detection: model-based and data-driven methods [10]. In the model-based fault detection technique, models of the faults are established to detect the occurrence of faults. This model may be mathematical or knowledge based. Multiple models based method is most commonly used in sensor fault diagnosis for autonomous mobile robots [11]. Roumeliotis et al. [12], present a Multiple Model Adaptive Estimation (MMAE) based technique for sensor failure detection and identification for mobile robots. In particular, each estimator is a Kalman filter enclosing a specific failure model which contributes to the system output at a given time. A particular feature of this solution is the use of measurements from several sensors asynchronously and at not-constant frequencies. However, the accuracy of MAME method decreases in situations where the system frequently undergoes inconsistent changes, such as system failures [20]. To overcome the inaccuracy of models for dynamic systems, some researchers investigate Bayesian approaches to fault diagnosis. For instance, Verma et al. [21] applied Bayesian method to dynamic state estimation, but it has computational and representational disadvantages. Currently, in the domain of autonomous mobile robots, accurate analytical models are often not feasible due to uncertainties in the environments, noises of the sensors, and malfunctioning of the actuators. Most of the approaches for robot's fault diagnosis are based on data-driven methods, which contrary to the model-based methods, do not require a priori knowledge about the system behavior. These techniques, only need a large amount of historical data to extract a knowledge base which is used as a reference of the fault diagnosis system. Neural networks have been widely applied to detect faults of mobile robots when using data-driven methods [13]. For instance, in the work proposed by Skoundrianos et al. [22], the multiple local model neural networks are trained to capture the input-output relationship for robot fault detection. In this work, authors focus on wheel's fault detection of a robot, using the voltage to the motor driving the wheel and the speed of the wheel as the input and output, respectively. In order to overcome the computational disadvantage of multiple local model neural networks, Christensen et al. [14], established a single neural network which is trained when the robot is operating normally and then re-trained when the robot is operating under different faults. Moreover, even if data-driven methods present high accuracy for the detection of a fault, it cannot localize where the fault occurred and cannot identify different types of faults. It has been shown that data-driven methods are not practical for autonomous mobile robots since they require the priori analysis of many failure cases and data that are too costly, and sometimes not available [15].

The approach proposed in this paper employs two main stages, e.g. residual generation and decision making, but unlike previous conventional methods, in our solution it is not needed to know physical parameters of the robot, which are generally difficult to obtain for complex mobile robot systems. Moreover, our approach presents a versatile and cheap solution, as it only uses wheel's encoders and onboard camera, which are commonly embedded in mobile robots system, to detect and isolate faults.

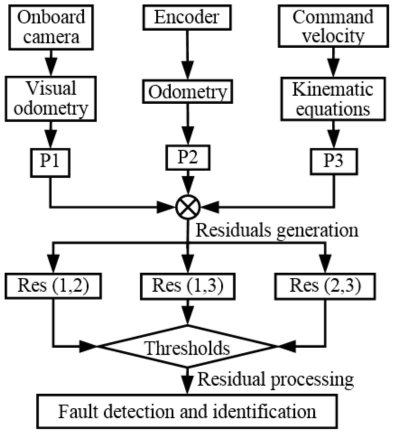


Fig. 2: Schematic representation of the fault diagnosis method. P1, P2 and P3 are the estimated poses of the robot based on each scheme.

Ⅲ. METHODOLOGY

Several sensors can provide real-time information about the robot's location or displacement. Our method computes the pose of the robot by using the command velocities, the wheels' encoders and a stereo camera. Then, residuals are generated by comparing each two estimated poses calculated from the three methods. Ideal fault free tests have been conducted to find reference thresholds. The detection of a fault is determined by comparing the residuals to the chosen thresholds. A bool value is assigned to each residual, e.g. if the residual is below the threshold its bool value is set to 0, otherwise it is set to 1, and combine them into a fault decision table. A general overview of the steps needed to detect and identify the faults is shown in the flow diagram in Fig. 2. The following subsections will explain the different methods for computing the poses of the robot in more details.

A. Pose Computed from the Commanded Velocities

Through using kinematics equations, the pose P_d of the robot can be computed from the commanded velocities (v, ω) at time t. This pose is represented by the vector:

$$\boldsymbol{p}_{d,t} = [x_{wr,t}\ z_{wr,t}\ \theta_{wr,t}]^{\mathrm{T}}; \qquad (1)$$

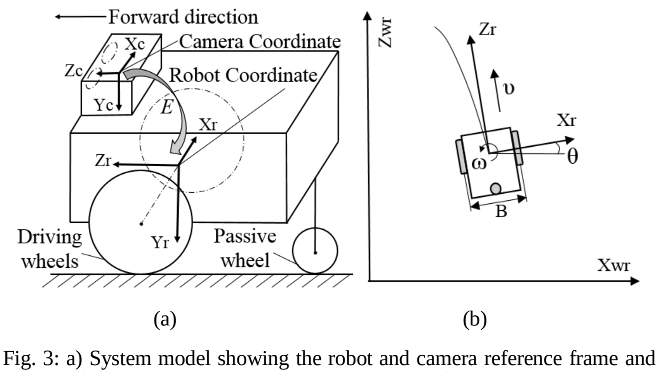


Fig. 3: a) System model showing the robot and camera reference frame and the transformation matrix *E*. b) The movement of robot on plane ground showing the coordinates systems.

where ($x_{wr,t}$ $z_{wr,t}$) are the 2D Cartesian coordinates and $\theta_{wr,t}$ is the heading angel of the robot in the world coordinate frame as illustrated in Fig. 3b). The pose of the robot can be estimated by starting from a known position and integrating the commanded velocities over time. For simplicity, the robot's world coordinate frame here is assumed to be the same with robot's coordinate frame at $t = 0$ time. Hence, the initial pose of the robot is $\boldsymbol{p}_{d,0} = [0\ 0\ 0]^{\mathrm{T}}$ and the update rule for $\boldsymbol{p}_{d,t}$ can be expressed as follow:

$$\boldsymbol{p}_{d,t} = \boldsymbol{p}_{d,0} + \begin{bmatrix} \int_0^t v * \sin\theta \ \ d_t \\ \int_0^t v * \cos\theta \ \ d_t \\ \int_0^t \omega \ d_t \end{bmatrix}; \qquad (2)$$

where v is the linear velocity and ω is angular velocity, which are provided by the control command.

B. Pose Computed by Odometry

Odometry is the use of data from sensors, such as wheel encoders, to retrieve the position of the robot[16]. The pose P_e of the robot computed from encoder data at kth time step is represented by the vector:

$$\boldsymbol{p}_{e,k} = [x_{wr,k}\ z_{wr,k}\ \theta_{wr,k}]^{\mathrm{T}}; \qquad (3)$$

Where ($x_{wr,k}$ $z_{wr,k}$) are the 2D Cartesian coordinates and $\theta_{wr,k}$ is the heading angle of the robot as illustrated in Fig. 3b). The pose can also be estimated by starting from a known position and integrating the robot's movements, e.g. summing the incremental travel distances, over time. For simplicity, the world coordinate frame here is assumed to be the same with robot's coordinate frame at $k = 0$ time step. So, the initial pose is $\boldsymbol{p}_{e,0} = [0\ 0\ 0]^{\mathrm{T}}$. The update of incremental travel distance (Δx_{k+1} Δz_{k+1} $\Delta\theta_{k+1}$) can be expressed as:

$$\Delta x_{k+1} = \Delta s \cos(\theta_{wr,k} + \tfrac{\Delta\theta_{k+1}}{2}); \qquad (4)$$

$$\Delta z_{k+1} = \Delta s \sin(\theta_{wr,k} + \tfrac{\Delta\theta_{k+1}}{2}); \qquad (5)$$

$$\Delta\theta_{k+1} = \tfrac{\Delta s_r - \Delta s_l}{B}; \qquad (6)$$

$$\Delta s = \tfrac{\Delta s_r + \Delta s_l}{2}; \qquad (7)$$

where (Δx_{k+1} Δz_{k+1} $\Delta\theta_{k+1}$) is the path traveled in the last sampling interval; Δs_r, Δs_l are the traveled distances of the right and left wheel obtained from the encoders' reading; The constant B represents the distance between the two driving wheels of the robot. Thus, the update rule for the robot's position can be expressed as follow:

$$\boldsymbol{p}_{e,k+1} = \boldsymbol{p}_{e,k} + \Delta\boldsymbol{p}_e; \qquad (8)$$

where $\Delta\boldsymbol{p}_e$ has the following form:

$$\Delta\boldsymbol{p}_e = \begin{bmatrix} \Delta x_{k+1}\cos(\theta_{wr,k} + \tfrac{\Delta\theta_{k+1}}{2}) - \Delta z_{k+1}\sin(\theta_{wr,k} + \tfrac{\Delta\theta_{k+1}}{2}) \\ \Delta x_{k+1}\sin(\theta_{wr,k} + \tfrac{\Delta\theta_{k+1}}{2}) + \Delta z_{k+1}\cos(\theta_{wr,k} + \tfrac{\Delta\theta_{k+1}}{2}) \\ \Delta\theta_{k+1} \end{bmatrix}. \qquad (9)$$

C. Pose Computed by Visual Odometry

The movements of the robot can be also monitored by using other sensing modalities, such as images from an on-board camera. Visual odometry is the process of determining the position and orientation of a robot by analyzing the associated sequential camera images [17]. The pose P_i of the camera which is estimated by visual odometry method using images at kth time step is represented by the vector:

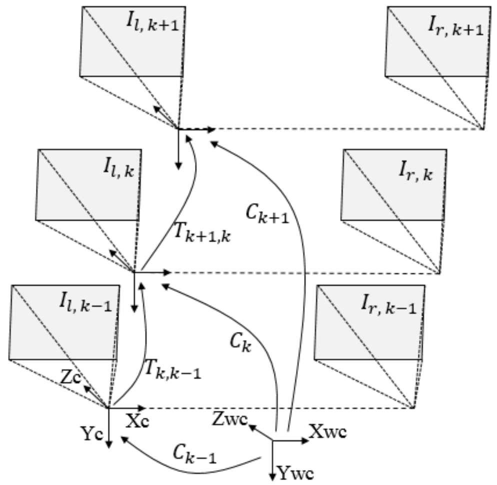


Fig. 4: Visual odometry method: an illustration of robot's pose estimation

$$\boldsymbol{p}_{i,k} = [x_{wc,k}\ y_{wc,k}\ z_{wc,k}\ \alpha_{wc,k}\ \theta_{wc,k}\ \gamma_{wc,k}]^{\mathrm{T}}; \qquad (10)$$

where ($x_{wc,k}$ $y_{wc,k}$ $z_{wc,k}$) are the 3D Cartesian coordinates of the camera and ($\alpha_{wc,k}$ $\theta_{wc,k}$ $\gamma_{wc,k}$) is the orientation of the camera, in the camera world coordinate frame, as illustrated in Fig. 4. For simplicity, the camera world coordinate frame here is assumed to be the same of camera coordinate frame at $k = 0$ time step. Thus, the initial camera's pose is $\boldsymbol{p}_{i,0} = [0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$. Two camera positions at adjacent time steps k-1 and k are related by the rigid body transformation $T_{k,k\text{-}1} \in \mathbb{R}^{4\times4}$, of the following form:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} \qquad (11)$$

where $R_{k,k\text{-}1} \in SO(3)$ is the rotation matrix, and $t_{k,k\text{-}1} \in \mathbb{R}^{3\times1}$ is the translation vector. $R_{k,k\text{-}1}$ and $t_{k,k\text{-}1}$ are extracted from the essential matrix M[18], as shown in **Algorithm1**. The set $T_{1:k} = \{T_{1,0}, \ldots, T_{k,k\text{-}1}\}$ contains all subsequent motions. The set of camera poses $C_{0:k} = \{C_0, \ldots, C_k\}$ contains the transformations of the camera with respect to the initial camera coordinate frame at $k = 0$ time step. C_k is computed by concatenating all the transformations $T_{1:k}$, and, therefore $C_k = C_{k\text{-}1} * T_{k,k\text{-}1}$. C_k also can be represented as [$x_{wc,k}$ $y_{wc,k}$ $z_{wc,k}$ $\alpha_{wc,k}$ $\theta_{wc,k}$ $\gamma_{wc,k}$]$^{\mathrm{T}}$, which contains the location and orientation of the camera at kth time step. Thus, the visual odometry method estimates $\boldsymbol{p}_{i,k} = [x_{wc,k}\ y_{wc,k}\ z_{wc,k}\ \alpha_{wc,k}\ \theta_{wc,k}\ \gamma_{wc,k}]^{\mathrm{T}}$ from C_k. If the robot moves on a plane ground, then $y_{wc,k} = \alpha_{wc,k} = \gamma_{wc,k} = 0$. Hence, the pose P_i of the camera can be simplified to $\boldsymbol{p}_{i,k} = [x_{wc,k}\ z_{wc,k}\ \theta_{wc,k}]^{\mathrm{T}}$, which also represents the robot pose as the camera is fixed on the robot.

Algorithm1: Visual odometry method

```
Input: Stereo images $I_l$ and $I_r$
Output: Robot's pose $\boldsymbol{p}_{i,k}$
Initialize $\boldsymbol{p}_{i,0} = C_0 = [0\ 0\ 0\ 0\ 0\ 0]^{\mathrm{T}}$
While $I_l, I_r \neq \emptyset$ do
    Undistort stereo image pairs $I_{l,k-1}$, $I_{l,k}$ and $I_{r,k-1}$, $I_{r,k}$
    Extract and match features between $I_{l,k-1}$ and $I_{l,k}$
    Compute $M = R * [R^{\mathrm{T}} * t]_{\times}$
    Decompose $M$ into $R_{k,k-1}$ and $t_{k,k-1}$
    Form $T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$
    Update $\boldsymbol{p}_{i,k} = C_k = C_{k-1} * T_{k,k-1}$
End while
```

D. Pose obtained from Motion Capture System

Motion capture, also known as Mo-cap, is the process of recording the movements of objects in the environments. Thanks to their accuracy, motion capture systems are currently used in many applications, such as entertainment, sports, medical applications, and for validation of computer vision and robotics methods [19]. The Mo-cap system, used in this work, is a "marker-based" optical motion capture system, which uses ten infrared cameras to view a scene from a variety of angles. Thus, just like binocular vision, which allows humans to see the world in three dimensions, Mo-cap technologies are able to reconstruct 3D dimensional space by using two or more cameras. Reflective markers, which are easy to be tracked, are placed on the robot and used as reference for the Mo-cap system, as shown in Fig. 1a). Recording the positions of these markers throughout the range of the motion, allows us to track the movements of the robot in real-time, which is then used as ground truth. The pose P_m of the robot computed by the motion capture system at k^{th} time step is represented by the vector:

$$\boldsymbol{p}_{m,k} = [x_{wr,k}\ z_{wr,k}\ \theta_{wr,k}]^{\mathrm{T}}; \tag{12}$$

where ($x_{wr,k}$ $z_{wr,k}$) are the 2D Cartesian coordinates and $\theta_{wr,k}$ is the heading angel of the robot in world reference frame, as illustrated in Fig. 3b).

E. Fault Diagnosis: Evaluation of the Residuals

In our method, we define the poses of the robot obtained from visual odometry method, odometry and the commanded velocities as P1, P2 and P3 respectively. Fig. 2 shows a schematic representation of the convention used in our approach. In order to compare the robot's poses obtained from different methods, the poses need to be mapped to the same coordinate frame. A transformation matrix E, is used here to map the camera world coordinate frame XwcYwcZwc into the robot world coordinate frame XwrYwrZwr, as shown in Fig. 3a). Only the pose P1, which is computed by visual odometry need to be mapped to the robot's reference system. Hence, the equations of the residuals $Res(i, j)$ of the robot poses calculated from different methods are defined below:

$$Res(i,j) = P_j - P_i * W; \tag{13}$$

where $W \in \mathbb{R}^{3\times3}$ is equal to E if $i = 1$, or to the identity matrix $I \in \mathbb{R}^{3\times3}$ otherwise.

In order to avoid false fault diagnosis, a bool value $BRes(i, j)$ of $Res(i, j)$ is computed by comparing the residuals to their thresholds, which have been computed in ideal fault free tests. The residuals between two poses is computed in both, x and z directions. If the residuals in both directions are below the thresholds, the bool value $BRes(i, j)$ of the residuals between two poses is set to 0, otherwise is set to 1. Small residuals are in general caused by system or sensor noise, while large residual can be potentially representing a fault or several faults in the robot's system. With the aim of determining faults on the wheels and encoders, a novel methodology is proposed in this paper, assuming that there are no other faults on the system. Table 1 shows the derived fault isolation decision table. For instance, robot's wheel fault will cause large discrepancy between P1 and both, P2 and P3, however the discrepancy between P2 and P3 will be small. As a result, *BRes*(1,2) and *BRes*(1,3) will be equal to 1 but *BRes*(1,3) will be 0. Hence, it is possible to detect the occurrence of a fault on the robot's wheel as shown in the second case of Table1.

Table 1: Fault isolation decision table

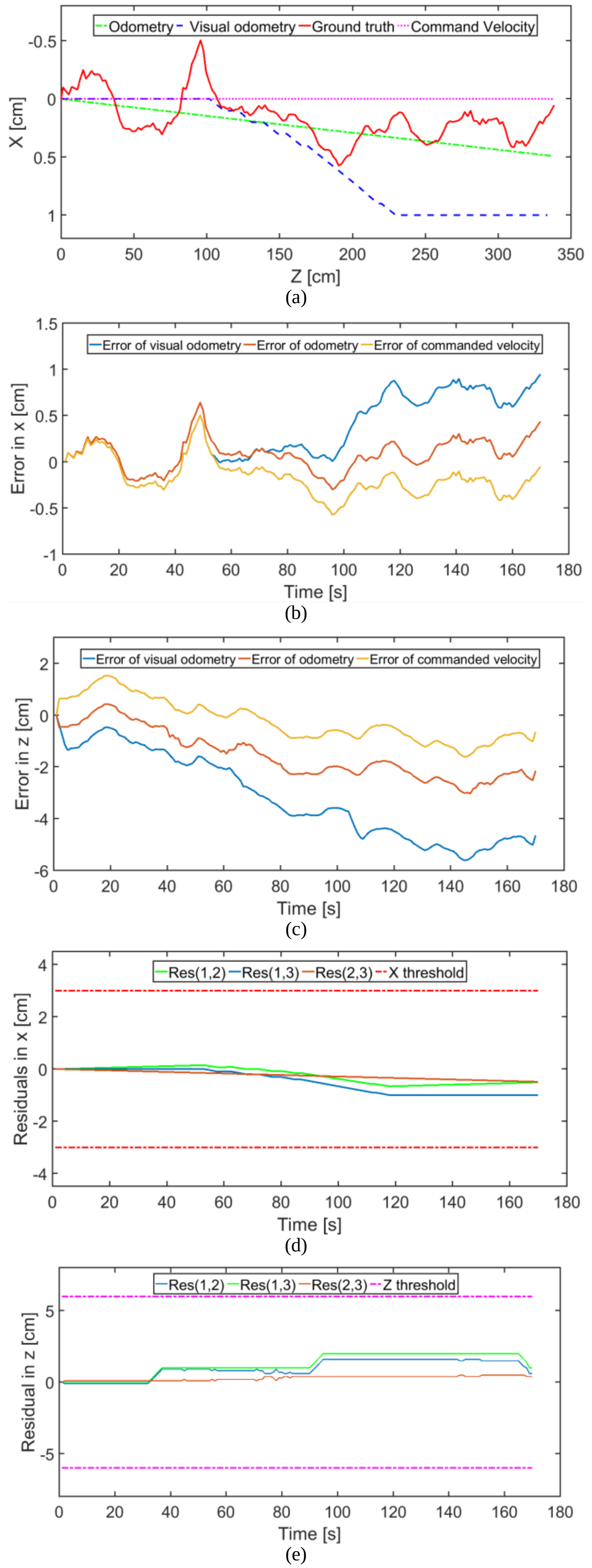
Case	*BRes*(1,2)	*BRes*(1,3)	*BRes*(2,3)	State
1	0	0	0	No fault
2	1	1	0	Wheel fault
3	1	0	1	Encoder fault
4	0	1	1	Motor fault

IV. Experimental Results

The Pioneer 3-DX mobile robot equipped with the ZED stereo camera, as shown in Fig. 1, has been used to conduct experimental tests and validate the proposed fault detection and identification method. All tests are performed on the plane ground in our laboratory room. The motion capture system (Motion Analysis. Corp.) is used here to obtain the ground truth pose of the robot.

During the ideal fault-free tests, the robot has been remote controlled to move in a straight path and synchronously tracked by the motion capture system. At the same time data from the rotary encoders and the images from the on-board camera have been recorded with sample frequency of 1Hz. The experimental results for one of the fault free tests are shown in Fig. 5. The trajectories of the robot obtained from the motion capture system and the one computed by the visual odometry and odometry methods, as well as the commanded velocities are shown in Fig. 5a). From the results it is clear that the trajectories generated by the three methods follow the trend of the ground truth, as there is no fault on the system. The error between the three poses and the ground truth data, in both x and z directions over time are shown in Fig. 5b) and Fig. 5c). In this test the robot travelled 340cm. The errors of the different methods are within 1cm in x direction and within 6cm in z direction. Experimental results show that, the poses obtained from odometry and the commanded velocities are better than the pose computed by visual odometry. This is due to two reasons: the first reason is that all the tests are conducted on the plane ground, which makes the work condition very ideal for using odometry and the commanded velocities; while the second reason is that the chosen scene does not contain enough features to be tracked by the image processing algorithm. The thresholds for residuals between each pair of the poses, in both x and z directions, are shown in Fig. 5d) and Fig. 5e). The thresholds are set five times of peak residuals in x and z directions separately. However, due to the visual odometry noise and the drift affecting the odometry method, the value of the residual grows over time.

In order to evaluate that a fault can be detected and identified when large residuals appear, we manually bumped one of the wheel, as shown in Fig. 1b). The experimental results of the test are shown in Fig. 6. During the experiment, the robot is remote-controlled to move in a straight path, but as the right bumped wheel moves longer than the left wheel, the robot will only turn left. As a result, the trajectories obtained through the motion capture system and the visual odometry have large discrepancy when compared to the ones computed by the odometry method and the commanded velocities, as shown in Fig. 6a). The residuals of the robot poses obtained from the three methods both in x and z direction are shown in Fig. 6b) and Fig 6c), respectively. From



(a)

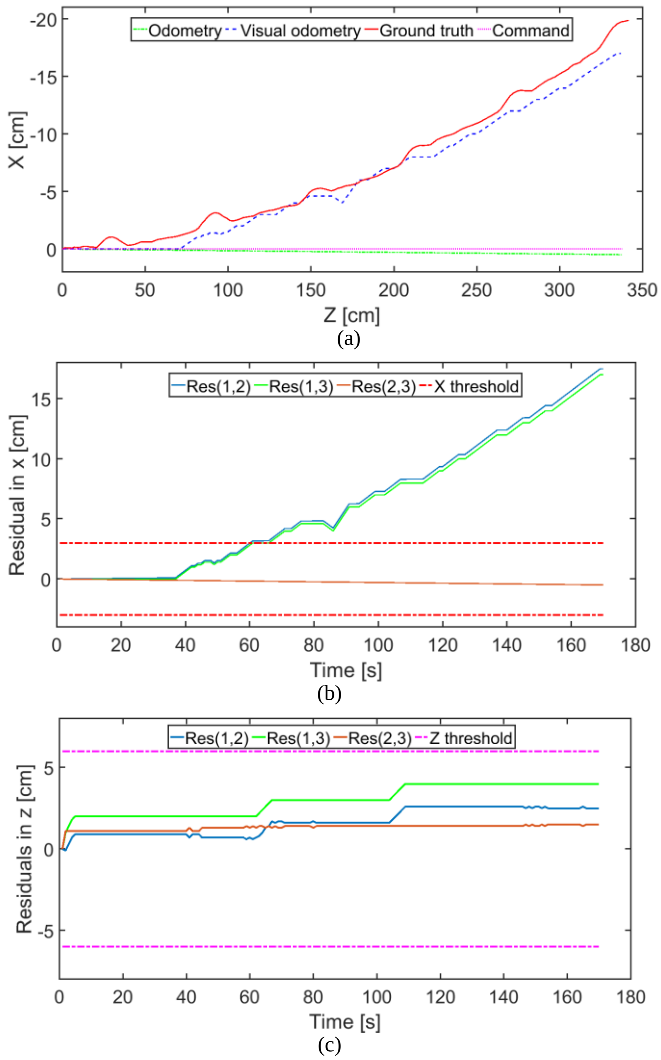
(b)

(c)

(d)

(e)

Fig. 5: Fault free tests results: a) Robot's trajectories computed from the three methods and by the motion capture system. Error between the computed poses and the ground truth in x direction b) and z direction c). Thresholds of the residual between each pair of three poses in x direction d) and z directions e).



(a)

(b)

(c)

Fig. 6: Fault diagnosis tests results: a) Robot's poses computed from the three methods and tracked by the motion capture system. Residuals of the three poses in x direction b) and z directions c).

these results, it is clear that *Res*(1,2) and *Res*(1,3) are above the threshold in x direction and only *Res*(2,3) is below the thresholds in both, x and z directions. Hence, according to methodology described in Section III, *BRes*(1,2) and *BRes*(1,3) are set to 1, while *BRes*(2,3) is set to 0. It has to be noticed that in this case the fault affects the pose of the robot mainly in the x direction, i.e. the residuals in the z direction are all within the thresholds. The detection of the fault is derived by checking the fault isolation decision table, shown in Table1. By using this table, our decision making approach successfully determines the presence of a fault in the wheel. This preliminary result demonstrates the ability of our method to successfully detect and identify a specified fault on the robot's system.

V. Conclusions and Future Works

This paper presents a novel approach for mobile robot fault detection and isolation. The method firstly utilizes the poses computed from visual odometry, odometry and the commanded velocities to generate residuals and then compare the residuals to the thresholds to diagnosis faults. Here, we have explained our approach for the creation of a fault detection and identification method, which to the best of authors' knowledge, is the first use of visual odometry method for mobile robot's fault diagnosis. Moreover, our method uses only low cost sensors, e.g. encoders and commercial digital camera, which are commonly available in mobile robots and have low hardware requirements. Hence, our solution can be easily adopted in different applications employing mobile robots. The experimental results show that, using our method, it is possible to monitoring the movements of the robots and successfully diagnose wheels' fault. In future work we will evaluate the capability of the developed method to diagnose encoders fault and identify in which part of the system, wheel or encoder is subjected to failure. The main problem with this method is that visual odometry and odometry errors will grow over time, thus, the discrepancy between the three poses will be unbounded. Therefore, it is not suitable to use a fixed threshold. In future work we will also perform analytical studies and evaluation of the optimal thresholds for the residuals using an adaptive threshold generation scheme.

Acknowledgment

This work was in part funded by Tough Robotics Challenge, ImPACT Program of the Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

References

[1] Z.-x. Cai, H.-g. He, and H. Chen, "Some issues for mobile robots navigation under unknown environments," Control and Decision, vol. 17, no. 4, pp. 385–390, 2002.

[2] K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg, "Indoor and outdoor localization for fast mobile robots," in Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 3. IEEE, 2004, pp. 2185–2190.

[3] K. M. Hasan, K. J. Reza et al., "Path planning algorithm development for autonomous vacuum cleaner robots," in Informatics, Electronics & Vision (ICIEV), 2014 International Conference on. IEEE, 2014, 1–6.

[4] N. Pouliot and S. Montambault, "Geometric design of the linescout, a teleoperated robot for power line inspection and maintenance," in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008, pp. 3970–3977.

[5] D. Stavrou, D. G. Eliades, C. G. Panayiotou, and M. M. Polycarpou,"Fault detection for service mobile robots using model-based method," Autonomous Robots, vol. 40, no. 2, pp. 383–394, 2016.

[6] D. Crestani, K. Godary-Dejean, and L. Lapierre, "Enhancing fault tolerance of autonomous mobile robots," Robotics and Autonomous Systems, vol. 68, pp. 140–155, 2015.

[7] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima et al., "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," Journal of Field Robotics, vol. 1, no. 30, pp. 44–63, 2012.

[8] M. Hashimoto, H. Kawashima, and F. Oba, "A multi-model based fault detection and diagnosis of internal sensors for mobile robot," in Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, vol. 4. IEEE, 2003, pp. 3787–3792.

[9] R. Isermann, Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Science & Business Media, 2006.

[10] F. Baghernezhad and K. Khorasani, "Computationally intelligent strategies for robust fault detection, isolation, and identification of mobile robots," Neurocomputing, vol. 171, pp. 335 – 346, 2016.

[11] Gertler, J. J. (1988). Survey of model-based failure detection and isolation in complex plants. IEEE Control Systems Magazine, 8, 3–11.

[12] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Sensor fault detection and identification in a mobile robot," in Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190), vol. 3, Oct 1998, pp. 1383–1388 vol.3.

[13] E. N. Skoundrianos and S. G. Tzafestas, "Finding fault - fault diagnosis on the wheels of a mobile robot using local model neural networks," IEEE Robotics Automation Magazine, vol. 11, no. 3, pp. 83–90, Sept 2004.

[14] A. L. Christensen, R. O'Grady, M. Birattari, and M. Dorigo, "Fault detection in autonomous robots based on fault injection and learning," Autonomous Robots, vol. 24, no. 1, pp. 49–67, 2008.

[15] M. Luo, D. Wang, M. Pham, C. Low, J. Zhang, D. Zhang, and Y. Zhao, "Model-based fault diagnosis/prognosis for wheeled mobile robots: a review," in Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE. IEEE, 2005, pp. 6–pp.

[16] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to autonomous mobile robots.

[17] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 1, June 2004, pp. I–652–I–659 Vol.1.

[18] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge university press, 2003.

[19] David Noonan, Peter Mountney, Daniel Elson, Ara Darzi and Guang-Zhong Yang. A Stereoscopic Fibroscope for Camera Motion and 3D Depth Recovery During Minimally Invasive Surgery. In proc ICRA 2009 , pp. 4463-4468.

[20] M. Hashimoto, H. Kawashima, T. Nakagami, and F. Oba, "Sensor fault detection and identification in dead-reckoning system of mobile robot: interacting multiple model approach," International Conference on Intelligent Robots and Systems, pp.1321-1326, 2001.

[21] Terra, M., & Tinos, R. (2001). Fault detection and isolation in robotic manipulators via neural networks: a comparison among three architectures for residual analysis. Journal of Robotic Systems, 18(7), 357–374.

[22] Verma, V., Gordon, G., Simmons, R., & Thrun, S. (2004). Real-time fault diagnosis [robot fault diagnosis]. IEEE Robotics & Automation Magazine, 11(2), 56–66. doi:10.1109/MRA.2004.1310942.