Multiple model adaptive estimation for blocked wheel fault detection on mobile robots

Mahmoud Almasri[1], Nicolas Tricot[1], and Roland Lenain[1]
[1]Irstea, Aubière, France
email: First name (dot) Last name @ Irstea.fr

Abstract

Nowadays, robots become increasingly more autonomous, which gives more importance to the Fault Detection and Isolation (FDI) task. In this article, major existing faults classification is specified. Faults are classified with respect to their time dependency, their source and their effect on system model. After that, mobile robotics-suitable FDI methods are classified into four main categories: material redundancy based, knowledge based, data based and model based approaches. Then, Extended Kalman Filter (EKF) and Multiple model Adaptive Estimation (MMAE) are explained and applied in a simulation to detect and isolate efficiently four wheel block faults, after studying briefly how wheel block faults affect the robot model. The average detection and isolation rate in the presented simulation is in order of 90%.

1 Introduction

1.1 Fault Detection and Isolation

Fault Detection and Isolation (FDI) is a crucial task to ensure greater autonomy of mobile robots. Fault can occurs at any time during the robot operation. It may prevent the robot from achieving its goal, and it may damage the equipment.

FDI consists at least on two stages:

- Fault detection: When something is wrong, the first step is to know that a fault has occurred. This operation is the fault detection.
- Fault isolation: Finding out what the source of a fault is, namely determining what the faulty component is..

A third stage may be added, fault accommodation. It consists on adapting the system so it still can achieve its goal despite the presence of fault.

1.2 Faults categorization

With respect to their time dependency, faults can be classified as (Fig. 1) [12]:

- Abrupt fault: sudden step appears on the signal.
- Drift fault: The signal deviates during time.
- Intermittent fault: The fault appears in an interval of time and ten disappears.

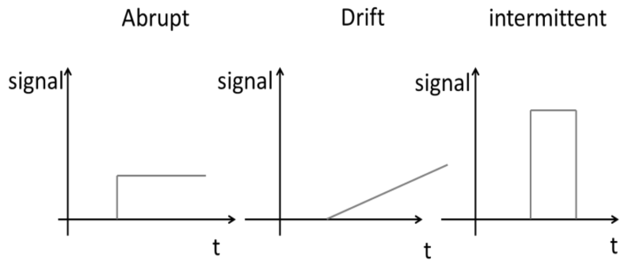

Fig. 1. Fault classification regarding time dependency

Furthermore, depending on their source, faults can be classified as:

- Sensor faults: e.g. IMU, GPS, odometers;
- External faults: e.g. invisible or negative obstacle, slip;
- Mechanical Faults: e.g. Blocked wheel, deflated wheel, suspension fault;
- Actuator fault: e.g. Motor fault.

Moreover, faults can be classified regarding their effect on the model of the system as additive or multiplicative faults. Considering the state space system in (1):

$$\dot{X}(t) = A\,X(t) + B\,u(t) + f_i \qquad (1)$$
$$Y(t) = C\,X(t) + f_o$$

Where f_i and f_o are additive faults on the input and output respectively. Multiplicative faults appear as a modification of the matrices A, B and C.

1.3 FDI methods overview

In order to be applied to mobile robots, FDI method has to respect three essential constraints:

- Online: the FDI method must diagnose the robot while he is doing his job.
- Real-time application: FDI method must not interrupt the operation of the robot.
- Cope with nonlinear models: The robot kinematic and dynamic models may have some degrees of nonlinearity. FDI method must have the capacity to deal with such models.

Many FDI methods exist in the literature, see [1] for a review. These methods can be classified essentially into four main categories:

- Material redundancy approaches: this is the most basic approach. It consists on adding redundant sensors to measure same variables. The comparison between their outputs leads to detection and isolation of faults.

- Knowledge-based approaches: in this type of methods, we should know the behavior of the system in each mode (normal or faulty). Then, the FDI is done by predicting the current mode of the system at each time. Main methods in this category are Particle filter and its variants [2-4].
- Data-based approaches: Artificial Intelligence [5-6] is a generic tool that can solve many classification or estimation problems. It can be used to treat the output of other methods or directly on the measurements to predict the actual mode [1].
- Model-based approaches: the main idea of these approaches is to use the mathematical model of the system for the FDI process. Relying on this model, we can estimate the variables and then, a comparison between these variables and measurements produces residues. The residues treatment leads to detection and isolation of faults. Many methods exist in this category, such as Extended Kalman Fileter (EKF) [1] and Multiple Model Adaptive Estimation (MMAE) [7-8].

MMAE is efficient and robust FDI technique that can deal with additive and multiplicative faults knowing their architectures. EKF is a good estimator for nonlinear models of first order. In the next sections, we will explain the principle of the Extended Kalman Filter (EKF) and the Multiple Model Adaptive Estimation (MMAE). And after, we will present a simulation of the MMAE on a mobile robot, where each of its parallel models is implemented using an EKF.

2 Methods theory

2.1 EKF

The EKF is an extension of the Kalman filter (KF) designed to deal with first order nonlinear systems. Its algorithm is similar to that of KF [7], but it does additional step of linearization. Given the following system model:

$$x_k = f(x_{k-1}, y_{k-1}) + w_{k-1} \qquad (2)$$
$$z_k = h(x_k) + v_k$$

It calculates the Jacobians (equation (3)) of state and output matrices and uses it in the KF algorithm.

$$A_k = \frac{\partial f}{\partial x}\bigg|_{\hat{x}_{k|k}, u_k} ; H_k = \frac{\partial h}{\partial x}\bigg|_{\hat{x}_{k|k}} \qquad (3)$$

In this paper, the Mahalanobis distance is used as a residue, It determines how well measured data fit predicted ones. It is calculated as in equation (4):

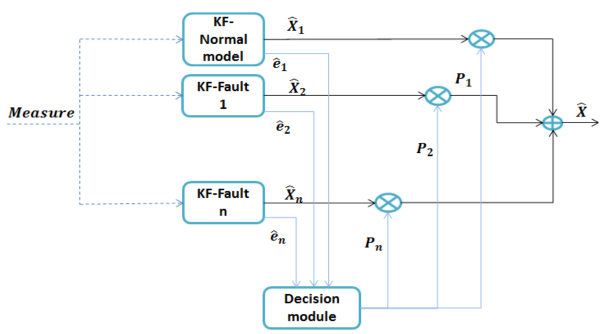

Fig. 2. MMAE basic architecture

$$r = ys^{-1}y^T \qquad (4)$$

Where y is the simple residue vector (predicted minus measured data) and s is the output covariance matrix.

EKF can cope efficiently with nonlinear models of first order. However, its performance degrades if the noise distribution on the system is very different from Gaussian one, or if the degree of nonlinearity is bigger than one.

2.2 Proposed MMAE scheme

Multiple Model Adaptive Estimation (MMAE) can be used in fault diagnosis to detect and isolate additive or multiplicative faults knowing their structures. It is robust and adapted to these types of faults. It can be used also to design fault tolerant control. In [7] and [8], MMAE approach is used to detect and compensate sensor and actuator faults in aircraft flight control systems. Surface control actuators and sensors (IMU) faults are successfully isolated and accommodated in real time.

Its principle is to run a bank of filters in parallel. Each filter implements a model matching one mode of the system, i.e. normal system, or system with particular fault. The outputs of these filters are then treated by a decision module to determine the actual mode of the system and to produce the final state estimate of the variables. Fig. 2 illustrates the basic structure of MMAE. A bank of Kalman Filters (KF - can be any version of KF) is adopted. Every KF outputs the state estimate x_i and the residue e_i. These two variables enter a decision module that generate a vector of probabilities p, where p_i is the probability that the actual mode of the system is mode i. For the FDI process, we will be interested in the model estimation for each mode and the decision.

As explained in the section 1, many faults have an additive or multiplicative effect on the model. In [10-11], a parameter estimation based FDI is designed for quadrotor faults. Many algorithms exist in the literature [9]. However, Least Square Estimation (LSE) [12] is widely used in this domain due to its generality and simplicity. The main idea of LSE is to find the parameters that reduce a cost function; this last is based on square error.

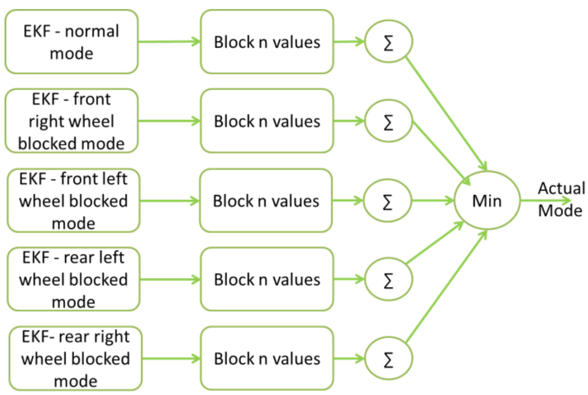

Fig. 3. Adopted MMAE scheme

The decision module takes the residue of each filter as an input, and then treats it to find the system mode. We propose to monitor the sum on a fixed size sliding window for the residue vector of each filter. A comparison between these sums leads to the identification of the actual system mode. Fig. 3 shows the complete proposed FDI scheme.

3 MMAE simulation on wheel block faults

3.1 Robot model and simulation environment

A simulation in Gazebo simulator under ROS is done to prove the theory. The Jaguar 4x4 wheel robot platform (Fig. 4) is used in this simulation. It is a skid steering light weight mobile robot, equipped with inertial measurement sensor (IMU), GPS and four odometers. The control signal is produced thanks to a joystick, to move the robot in relatively slow speed (linear speed lower than 2 m/s and angular speed lower than 1.5 rad/s).
The kinematic model in the absolute 2D frame is given by the Newtonian equations:

$$\begin{aligned} x_{t+1} &= x_t + v.\cos(\theta).dt \\ y_{t+1} &= y_t + v.\sin(\theta).dt \\ \theta_{t+1} &= \theta_t + w.dt \end{aligned} \quad (5)$$

Where x, y and θ are the position and the orientation of the robot in the absolute 2D frame. v and w are linear and angular velocity. They represent the input signal.

3.2 Fault modeling

Wheel block faults are studied in this paper. Considering the representation of Fig. 5, with F_i is active force produced by the wheel I and R_i are reactive force resulting of wheels friction. Newton's law on the forces on the normal robot gives this equation:

$$\sum_i (\vec{F_i} + \vec{R_i}) = \vec{F} + \vec{R} = m.\overrightarrow{a_n} \quad (6)$$

With $\overrightarrow{a_n}$ is the acceleration vector of the normal robot.
If one wheel is blocked, then the force produced by this wheel is set to zero, and the friction value is increased. The modified equation after projection on X is:

$$\alpha_1 F + \alpha_2 R = m.a_b \quad (7)$$

With a_b is the acceleration vector of the robot with blocked wheel.
By subtracting (6) and the projection of (7):

$$F(\alpha_1 - 1) + R(\alpha_2 - 1) = m(a_b - a_n) \quad (8)$$

By arranging this equation and integrating with respect to time:

$$v_b = \frac{\int (F(\alpha_1 - 1) + R(\alpha_2 - 1))dt}{m} + v_n \quad (9)$$

Without loss of generality, supposing constant inputs v and w, and ignoring the wheel slip, active and reactive forces become constants. So, we can rewrite the equation (9) as:

$$v_b = \beta_v + v_n \quad (10)$$

Where v_b is the velocity vector of the robot having a blocked wheel and v_n is the velocity vector of the normal robot. It corresponds normally to the control signal v, and β_v is the fault parameter.
Furthermore, starting from the equation (11):

$$L(\vec{F}_1 + \vec{F}_4 - \vec{F}_2 - \vec{F}_3) = L(2\vec{F}_r - 2\vec{F}_l) = \vec{M} = I.\ddot{\vec{\theta}} \quad (11)$$

Fig. 4. Jaguar 4x4 wheel mobile robotic platform

Where $\vec{M}$, I and $\ddot{\vec{\theta}}$ are the resulting torque, robot's inertial matrix and the angular acceleration. Following the same reasoning, we obtain after projections:

$$w_b = L\frac{\int (F(\alpha_3 - 1) + R(\alpha_4 - 1))dt}{I} + w_n \quad (12)$$

Thus, we can rewrite this equation as:

$$w_b = \beta_w + w_n \quad (13)$$

Where w_b is the angular velocity of the robot having a blocked wheel, w_n is the angular velocity of the normal robot and β_w is the fault parameter. Equation (9) indicates how this fault can affect the value of the angular velocity. In addition, considering a control signal $(v, w) = (constant, 0)$. In this case, because one wheel is blocked, the sum of forces on each side of the robot is not even. So, the torque is a constant different than zero and a parameter β_{v2} will appear in the equation. Therefore, by integrating these parameters in equation (5), the model of the robot having a blocked wheel has this form:

$$\begin{aligned} \boldsymbol{x_{t+1}} &= \boldsymbol{x_t + v.(1 + \beta_v).\cos(\theta).dt} \\ \boldsymbol{y_{t+1}} &= \boldsymbol{y_t + v.(1 + \beta_v).\sin(\theta).dt} \\ \boldsymbol{\theta_{t+1}} &= \boldsymbol{\theta_t + (w + w.\beta_w + v.\beta_{v2}).dt} \end{aligned} \quad (14)$$

This model can be used ideally supposing that the robot moves on a homogeneous land.

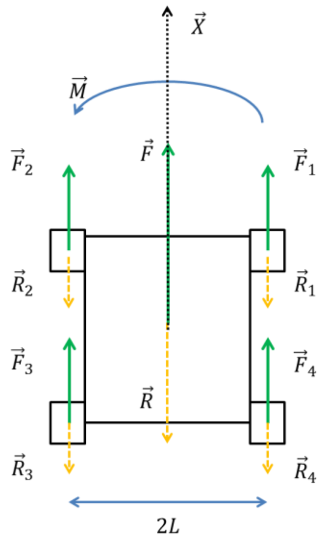

Fig. 5. Forces and torque

3.3 Application and results

Now, we have the fault architecture. The next step is to estimate fault parameters, or the betas. Thus, by running a simulation of the robot on a flat ground, we have created a database for each considered mode, i.e. normal mode and front right, front right, rear left and rear right wheel block modes. The collected data are the command signal and the position of the robot (x, y and θ) provided by the pose provider. This later is a data fusion of information coming from wheel encoders, Inertial Measurement Central (IMU) and GPS. To obtain a representative database, the command signal given to the robot during registering period must cover all the possible values i.e. v must vary from $-v_{max}$ to $+v_{max}$ and w must vary from $-w_{max}$ to $+w_{max}$. Each of these databases contains the position and the command signal of the robot; with 50 Hz sample frequency and nearly 60 seconds length. Then, LSE algorithm is applied on these databases to find the betas matching each mode of the robot. For this particular simulation, parameters are presented in the Table 1.

Table 1. Estimated parameters

	normal	Front right blocked	Front left blocked	rear left blocked	rear right blocked
β_v	-0.1	-0.2	-0.2	-0.2	-0.2
β_{v2}	0	-0.5	0.4	0.6	-0.6
β_w	-0.3	-0.3	-0.37	-0.6	-0.6

Once we have these parameters, we can implement the models matching each mode in an EKF to identify the system mode, as in Fig. 3.

Further, simulation has been done to test the performance of the proposed scheme. The response of the diagnosis scheme in two cases is presented in Fig. 6 and Fig. 7. In both simulations, the robot was initially normal. At iteration 500 a fault is injected; rear right wheel block fault (b_rr) for Fig. 6 and front right wheel block fault (b_fr) for Fig. 7. The robot returns to it normal state at iteration 1900.

We can see that they are no false alarm (i.e. the diagnosis method detects a fault while the system is normal) in both cases, and good isolation most of the time. However, little false fault isolation appears at the beginning of the fault and while returning to the normal mode. Furthermore, the isolation time is less than 1s. On the other hand, the filter detects the normal mode after several seconds of its presence. This transition time depends on the size of the residuals blocking window i.e. the number n of blocked values as in Fig. 3. In fact, the rate of good diagnosis depends on the behavior of the robot when the change of mode occurs. If a right wheel block fault occurs while the robot is turning to the right, it may be detected faster if it was moving straight forward. Furthermore, the MMAE switching between modes could be faster for higher speed.

The good detection rate is high (in order of 90% for long simulations). The algorithm takes some seconds during transitions to stabilize. But it is robust during steady states periods.

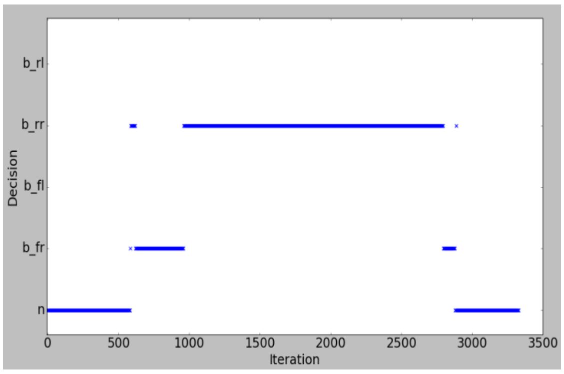

Fig. 6. Response with b_rr fault

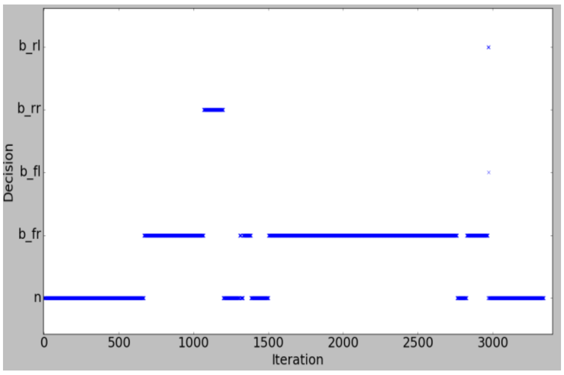

Fig. 7. Response with b_fr fault

4 Conclusion and perspectives

MMAE is a good approach to isolate additive or multiplicative faults knowing their structures. An MMAE based FDI method is presented in paper. This method operates on high level data, i.e. position obtained after sensor fusion. A simulation is explained in this paper. It proves that the performance and the time efficiency of this FDI method is good in the simulation case. It isolates four wheel block faults in real time. Operating on the same data level, an EKF method cannot detect these faults efficiently [1]. Furthermore, performance of this MMAE-based FDI method can be enhanced more by studying more deeply the fault model.

However, this method suffers from some limitations. It is limited to faults with known structures, having nearly constant additive or multiplicative effect on the model. Moreover, even with such faults (known structure), because of the limited calculation capacity on board, we cannot run a big number of filters in parallel. This means, we cannot use this method to isolate a large number of faults. This leads us to the conclusion of the recent research [1], that one and only one FDI method is not enough to diagnose a big list of mobile robotics faults.

Yet, in order to diagnose a large set of mobile robotics faults in real time, we need a hybrid FDI method. It will be formed of a combination of FDI methods, in a way that every fault is monitored with the FDI method that can cope most efficiently with it. To achieve this goal, a clear and objective comparison between FDI methods must be done.

Future work will consist on:

- Testing this MMAE approach on other faults, such as sensor or actuator faults.
- Application of other FDI methods cited in the previous research paper [1]
- Definition of an objective universal performance indicator to compare the performances of FDI methods those are able to diagnose same fault.
- Definition of a hybrid diagnosis method, able to diagnose efficiently big list of mobile robotics faults in real time.

Acknowledgments

The authors wish to thank the "Region Auvergne-Rhône-Alpes" for funding this research.

References

[1] M. Almasri, N. Tricot and R. Lenain 'Toward an optimal assignment of diagnosis method to mobile robots faults', in *International Conference on Control, Decision and Information Technologies. IEEE, 2018*, vol. 5.

[2] N. De Freitas, 'Rao-Blackwellised particle filtering for fault diagnosis', in *Aerospace Conference Proceedings, 2002. IEEE*, 2002, vol. 4, pp. 4–4.

[3] F. Hutter and R. Dearden, 'The gaussian particle filter for diagnosis of non-linear systems', *IFAC Proceedings Volumes*, vol. 36, no. 5, pp. 909–914, 2003.

[4] X. Koutsoukos, J. Kurien, and F. Zhao, 'Monitoring and diagnosis of hybrid systems using particle filtering methods', in *International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[5] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun, 'Monitoring a robot swarm using a data-driven fault detection approach', *Robotics and Autonomous Systems*, vol. 97, pp. 193–203, 2017.

[6] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, 'A survey of deep neural network architectures and their applications', *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[7] P. S. Maybeck, "Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems," *International Journal of Robust and nonlinear control*, vol. 9, no. 14, pp. 1051–1070, 1999.

[8] P. Lu, L. Van Eykeren, E.-J. van Kampen, C. de Visser, and Q. Chu, "Double-model adaptive fault detection and diagnosis applied to real flight data," *Control Engineering Practice*, vol. 36, pp. 39–57, 2015.

[9] V. Strejc, "Least squares parameter estimation," in *System Identification*, Elsevier, 1981, pp. 535–550.

[10] R. C. Avram, X. Zhang, J. Campbell, and J. Muse, "IMU Sensor Fault Diagnosis and Estimation for Quadrotor UAVs," *IFAC PapersOnLine*, vol. 48, pp. 380–385, Jan. 2015.

[11] C. Liu, B. Jiang, and K. Zhang, "Incipient Fault Detection Using an Associated Adaptive and Sliding-Mode Observer for Quadrotor Helicopter Attitude Control Systems," *Circuits, Systems, and Signal Processing*, vol. 35, no. 10, pp. 3555–3574, Oct. 2016.

[12] R. Isermann, 'Model-based fault-detection and diagnosis–status and applications', *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.