A Novel Approach to Fault Detection and Identification in Suction Foot Control of a Climbing Robot

Jiang Yong1,2, Wang Hongguang1,2, Fang Lijin1, and Zhao Mingyang1
1Robotics Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
2Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China
E-mail: jiangyong@sia.cn hgwang@sia.cn ljfang@sia.cn myzhao@sia.cn

Abstract—This paper presents a multiple-model and Boolean logic reasoning (MMLBR) approach to detect and identify faults in the suction foot control of a climbing robot. For this control system, some fault models are easily given by kinematics equations. Moreover, the logic relations of the system states have been known in advance. Based on the combination of the multiple-model adaptive estimation (MMAE) algorithm and the Boolean logic reasoning, the MMLBR approach is properly fit for the fault detection and identification (FDI) application to the climbing robot. In the MMLBR architecture, the MMAE algorithm is used to reliably detect and identify the model-known faults. Then based on the robot’s states and the results of the MMAE, other faults are detected and identified using the Boolean logic reasoning. Experimental results validated that the faults of the sensors and actuators in the suction foot control of the robot can be readily detected and identified by the MMLBR approach.

Index Terms - Fault detection and identification, Multiple-model adaptive estimation, Boolean logic reasoning, MMLBR, Climbing robot

I. INTRODUCTION

Since mobile robots are mostly designed to act autonomously, without direct supervision, the demand of fault detection and identification (FDI) in such parts as sensors, actuators and control units are significantly increasing for assuring mobile robots reliability and safety. Over the years, a number of different FDI techniques making use of either model based approach (for example, Kalman filter based approach) or model-free based approach (for example, soft computing based approach) have been proposed.

Model based FDI depends heavily on the presence of an analytical model of the process. Based on the concept of analytical redundancy in this approach, a residual signal is generated by comparing the measured output signal and the estimated one from a nominal system model. After being processed, this residual can be used as the indicator of the fault. Roumeliotis et al. [1] presented an execution of the multiple-model adaptive estimation (MMAE) to the FDI in the wheeled mobile robot Pioneer . Subsequently, Goel et al. [2] used the MMAE algorithm that integrated with a back propagation neural network to diagnose the complex faults in Pioneer AT, a four-wheeled robot. Further, in work by Hashimoto et al. [3] an interacting multiple model (IMM) approach to sensor FDI in the dead reckoning of mobile robot was proposed and thus greatly improved the performance for the systems with frequent mode jumps. Washington [4] presented MAKSI (Markov and Kalman state identification), an on-board method for fault diagnosis of the robot Rover. The MAKSI is based on a combination of continuous probabilistic state estimation using Kalman filters and discrete qualitative state estimation using a Markov model representation. In addition to Kalman filters, recently particle filters have been developed and applied to the FDI problem of mobile robots. Dearden et al. [5] successfully used particle filters to detect and classify the faults in the Rover.

For nonlinear or uncertain systems, model based FDI approaches do not perform well. To solve this problem, model-free based FDI approaches have been proposed. Young Moon Park et al. [6] presented a LBEES (logic based expert system) for fault diagnosis of power systems. Soika [7] proposed a concept for fault detection and calibration of external sensors on mobile robots, based on a grid representation of the environment. Other approaches, such as HMM2 (second-order hidden Markov models) based FDI [8] and FTA (fault tree analysis) based FDI [9] have also been studied.

In this paper, we present a MMLBR (multiple-model and Boolean logic reasoning) approach to detect and identify the faults in the suction foot control of a climbing robot. Based on a combination of the MMAE algorithm and the Boolean logic reasoning, the MMLBR approach is properly fit for the FDI to the sensors and actuators in the mobile robot. The organization of the paper is as follows. Section II briefly describes the mechanical structure and the suction foot of the climbing robot. Section III proposes the MMLBR approach. Section IV analyses the experimental results where the MMLBR approach is used to detect and identify the faults in the suction foot control of the climbing robot. Finally, section V outlines the main conclusions of the work.

II. CLIMBING ROBOT

A. Mechanical Structure

The mechanical structure of the climbing robot is designed as a bipedal robot with an under-actuated mechanism [10]. As shown in Fig. 1, motors 1 and 3 independently drive joints 1 and 5, respectively, thereby adjusting the tilt angle of the suction feet 1 and 2 so that the robot can grip the surface firmly. Motor 2 is installed to control joints 2, 3 and 4, separately. Joints 2 and 4 are revolute joints providing steering capability of the feet relative to the legs. Joint 3 represents the prismatic motion of the legs that allows the robot to expand and contract its legs.
The under-actuated mechanism enables the robot to drive five joints by only three motors, thus reducing both the weight and the power consumption of the robot, and achieving good balance between compactness and maneuverability.

B. Suction Foot

The bipedal climbing robot is supported by two suction feet that provide the robot with the ability to walk on a horizontal surface as well as climb a vertical wall. The suction foot is shown in Fig. 2.

Its main components are a diaphragm-type vacuum pump, a suction cup, a pressure sensor and a micro machined shape memory alloy valve. The connector integrates the foot components and serves as a mounting platform for the robot body. The suction cup is used for adherence. The pressure sensor monitors the pressure level inside the suction cup to ensure that the foot is firmly attached to the object surface. The foot is released through the actuation of the valve by a signal from the robot controller. Two touch sensors are also attached to the suction cup in opposite directions. This gives information on which part of the suction cup has touched the surface and facilitates the robot in adjusting the suction foot orientation.

III. FDI Approach

The MMBLR approach to FDI is based entirely on a combination of the MMAE algorithm and the Boolean logic reasoning. The basic idea behind the MMBLR is shown schematically in Fig. 3.

In general, for mobile robots, following two assumptions are valid. One is that some fault models of the system are easily given by robot kinematics and dynamics. The other is that the logic relations of the system states could be known in advance. In the MMBLR architecture, the MMAE algorithm is used to reliably detect and identify the model-known faults. Then based on the system states of the mobile robot and the results of the MMAE, other faults are detected and identified using the Boolean logic reasoning.

A. MMAE Algorithm

The MMAE algorithm is composed of a bank of parallel Kalman filters, each with a different internal fault model, and a hypothesis testing computation, as diagrammatically shown in Fig. 4.

Each Kalman filter uses its own fault model, along with a given input \( \hat{u} \), to develop an estimation of the current system states \( \hat{x}_k \), independent of the other filters. The filter then uses this estimation, along with the current measurement of those states \( \hat{z}_k \), to form the residual \( \hat{r}_k \), which is the difference between the measurement and the filter’s prediction of the measurements before they arrive. The residuals from the filters are used by the hypothesis testing computation as a relative indication of how close each of the filter models are to
the true model. The smaller the residual, the closer the filter model matches the true model. The hypothesis testing computation first scales the residuals to account for various uncertainties and noises in the measurements, and then computes the conditional probability for each of the hypotheses modeled in the bank of Kalman filters \( P_k \). These probabilities are then used to weight the individual Kalman filter state estimates to produce a blended estimate of the true state \( x_{MAE} \), which can then be used as the optimal estimation of the states by a control system. When used for fault identification, each of the Kalman filters would model a different fault condition, and the residuals from each filter would indicate how close that filter's model is to the actual fault condition. By monitoring these residuals, the hypothesis test computation can estimate the current fault status of the system \( \hat{a} \).

The model for the \( k^{th} \) filter is assumed to be a linear time-invariant, discrete-time system of the form

\[
\begin{align*}
    x_k(t_{i+1}) &= \Phi_k x_k(t_i) + B_k u(t_i) + G_k w_k(t_i) \\
    z(t_i) &= H_k x_k(t_i) + v_k(t_i)
\end{align*}
\]

where \( x_k \) is the Kalman filter model state vector; \( \Phi_k \) is the Kalman filter model state transition matrix; \( B_k \) is the Kalman filter model control input matrix; \( u(t) \) is the system input vector; \( G_k \) is the Kalman filter model noise input matrix; \( w_k \) is a white discrete-time dynamics noise input; \( z \) is the Kalman filter model measurement vector; \( H_k \) is the Kalman filter model output matrix; \( v_k \) is a white measurement noise input.

The statistics characteristics of \( w_k \) and \( v_k \) are

\[
\begin{align*}
    E[w_k(t_i)] &= 0 & E[v_k(t_i)] &= 0 \\
    E[w_k(t_i) w_k^T(t_j)] &= Q_k & t_i = t_j \\
    0 & t_i \neq t_j \\
    E[v_k(t_i) v_k^T(t_j)] &= R_k & t_i = t_j \\
    0 & t_i \neq t_j \\
    E[w_k(t_i) v_k^T(t_j)] &= 0
\end{align*}
\]

The state estimate \( \hat{x}_k(t_{i-1}) \) and its covariance matrix \( P_k(t_{i-1}) \) are propagated forward from time \( t_{i-1} \) to time \( t_i \) by the discrete time propagation equations

\[
\begin{align*}
    \hat{x}_k(t_i) &= \Phi_k \hat{x}_k(t_{i-1}) + B_k u(t_{i-1}) \\
    P_k(t_i) &= \Phi_k P_k(t_{i-1}) \Phi_k^T + G_k Q_k(t_{i-1}) G_k^T
\end{align*}
\]

The propagated optimal state estimate and its covariance matrix are then updated with information from the current measurement \( z(t_i) \) weighted according to a Kalman filter gain \( K_k(t_i) \). The update equations are

\[
\begin{align*}
    K_k(t_i) &= P_k(t_i) H_k^T \left[ H_k P_k(t_i) H_k^T + R_k(t_i) \right]^{-1} \\
    \hat{x}_k(t_i) &= \hat{x}_k(t_{i-1}) + K_k(t_i) [z(t_i) - H_k \hat{x}_k(t_{i-1})] \\
    P_k(t_i) &= P_k(t_{i-1}) - K_k(t_i) H_k P_k(t_{i-1}) H_k^T
\end{align*}
\]

Given that fault \( k \) has occurred \((1 \leq k \leq n)\) and the measurement history vector is

\[
    z_{i-1} = [z^T(t_1) \cdots z^T(t_{i-1})]^T
\]

The conditional density function of the measurement \( z(t_i) \) is computed as

\[
f_{z(t_i)|z_{i-1}}(z(t_i)|a_k, z_{i-1}) = \left(2\pi|S_k|^{1/2}\right)^{m/2} e^{-1/2(z(t_i)-a_k)^T S_k^{-1}(z(t_i)-a_k)} \]

where \( a_k \) is a vector of parameters specific to the \( k^{th} \) fault; \( m \) is the dimension of the measurement vector \( z(t_i) \); \( S_k \) is the covariance of the residual. The conditional probability \( p_k \) can be computed as

\[
p_k(t_i) = \frac{f_{z(t_i)|a_k, z_{i-1}}(z(t_i)|a_k, z_{i-1}) p_k(t_{i-1})}{\sum_{j=1}^{n} f_{z(t_i)|a_j, z_{i-1}}(z(t_i)|a_j, z_{i-1}) p_j(t_{i-1})}
\]

B. Boolean Logic Reasoning

The Boolean logic reasoning (BLR) based FDI in mobile robots is used to detect and identify the faults of the sensors, actuators and control units according to the logic relations of the system states. Here, the BLR approach relies on the key assumption that the characteristics of the system can be distinguished using Boolean algebra, i.e., “0” and “1”. The BLR approach consists of a system state module, a Boolean logic rule module and a logic reasoning module, as shown in Fig. 3.

For using the Boolean logic reasoning to detect and identify the faults in mobile robots, first of all, we must define a cause Boolean function \( A(A_1, A_2, \cdots, A_m) \), where \( A_1, A_2, \cdots, A_m \) are \( m \) fault causes, an omen Boolean function \( C(c_1, c_2, \cdots, c_n) \), where \( c_1, c_2, \cdots, c_n \) are \( n \) fault omens, and a decision Boolean function \( E(A_1, A_2, \cdots, A_m, c_1, c_2, \cdots, c_n) \). The decision function \( E(\cdot) \) is formulated based on the logic relations of the system states. Then the basic problem of the BLR based FDI is how to detect and identify the fault causes (cause function \( A(\cdot) \)), if the fault omens (omen function \( C(\cdot) \) and
the decision rules (decision function \( E(\cdot) \)) have been known in advance.

The process of the BLR based FDI is described as follows.

**Step 1**: Analyze the possible fault omens \( c_1, c_2, \ldots, c_n \) and causes \( A_1, A_2, \ldots, A_m \) in the system.

**Step 2**: Define the logic rules \( R_i, i = 1, 2, \ldots \), for example

\[
R_1: A_1 \rightarrow c_1, \text{ i.e., IF the fault cause is } A_1, \text{ THEN the omen is } c_1 .
\]

\[
R_2: c_1 \land c_2 \rightarrow A_2 , \text{ i.e., IF the fault omen is the logic AND of } c_1 \text{ and } c_2 , \text{ THEN the cause is } A_2 .
\]

\[
R_3: A_1 \lor A_3 \rightarrow c_1 \land c_3 , \text{ i.e., IF the fault cause is the logic OR of } A_1 \text{ and } A_3 , \text{ THEN the omen is the logic AND of } c_1 \text{ and } c_3 .
\]

**Step 3**: Based on the truth table of the fault causes, omens and logic rules, deduce the decision function \( E(\cdot) \).

**Step 4**: Identify the fault causes using decision function \( E(\cdot) \) and real omens occurred in the system.

### IV. EXPERIMENTS

#### A. Suction Foot Control

The bipedal climbing robot has the ability to transit between differently inclined surfaces. When moving from a floor to an object surface, the most important thing the robot needs to do is to adjust the suction foot orientation in order that the suction cup can fully contact the object surface. The process of the suction foot control is as follows.

**Step 1**: Rotate Joint 1 to bring the suction cup (foot 2) near the object surface.

**Step 2**: When either of the two touch sensors attached to the suction cup generates a signal (The signal means that this part of the suction cup has touched the object surface), Joint 5 is rotated to make the untouched part near the object surface.

**Step 3**: If neither of the two touch sensors has generated a signal, carry out **Step 1**.

**Step 4**: When both of the two touch sensors generate signals synchronously, start the vacuum pump and then the pressure sensor monitors the pressure level inside the suction cup to ensure that the foot 2 is firmly attached to the object surface.

#### B. Experimental Results

In the experiments reported here, the MMBLR approach is used to detect and identify the faults in the suction foot control of the climbing robot, as shown in Table I.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>FAULTS AND FDI APPROACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main components of the suction foot control</td>
<td>FDI approach</td>
</tr>
<tr>
<td>Touch sensor 1</td>
<td>MMAE</td>
</tr>
</tbody>
</table>

A D-H coordinate system has been established for each robot link, as shown in Fig. 5.

![Fig. 5 D-H Coordinate System](image)

A kinematics of the climbing robot is then given by

\[
X(t + 1) = \Phi X(t) + BU(t) + G_1 W(t)
\]

\[
Z(t) = HX(t) + G_2 V(t)
\]

\[
\Phi = H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -b & -c & -a & 0 \\ -c & b & 0 & a \\ -b & c & a & 0 \\ c & b & a & 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} g_1 & 0 & 0 & 0 \\ 0 & g_1 & 0 & 0 \\ 0 & 0 & g_1 & 0 \\ 0 & 0 & 0 & g_1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} g_2 & 0 & 0 & 0 \\ 0 & g_2 & 0 & 0 \\ 0 & 0 & g_2 & 0 \\ 0 & 0 & 0 & g_2 \end{bmatrix}
\]

where \( t + 1 \) and \( t \) are sampling times. The state vector \( X = [x_1 \ x_2 \ x_3 \ x_4]^T \) represents the points \( A \) and \( B \) (where the touch sensors 1 and 2 are attached to the suction cup, separately) positions with respect to the reference frame. Each input \( u_i \ (i = 1, \cdots, 4) \) of the input vector \( U = [u_1 \ u_2 \ u_3 \ u_4]^T \) is the function of the joint variables \( \theta_1 \) and \( \theta_2 \). The noise vectors \( W = [w_1 \ w_2 \ w_3 \ w_4]^T \) and \( V = [v_1 \ v_2 \ v_3 \ v_4]^T \) are assumed to be zero-mean white Gaussian sequences. The statistics characteristics of \( W \) and \( V \) are described as (2)-(5). The parameters \( a = 180, b = 40, c = 22, g_1 = 2, g_2 = 0.1 \) .

The experimental results of the MMAE based FDI to the touch sensors are shown in Fig. 6.
As shown in Fig. 6 (a), the experimental results indicate that both of the two touch sensors fail to function synchronously. In this case, the rotational angle $\theta_1$ of the joint 1 overruns a limit $\theta = 101^\circ$ at time $t = 8$ seconds and the fault conditional probability rises gently to 1. In Fig. 6 (b), the results show the case of the touch sensor 1 fault. In this case, the rotational angle $\theta_2$ of the joint 5 overruns a limit $\theta = 128^\circ$ at time $t = 4$ seconds and the fault conditional probability rises sharply to 1. In Fig. 6 (c), the results show the case of the touch sensor 2 fault. Similar to the second case, the rotational angle $\theta_2$ of the joint 5 overruns a limit $\theta = 100^\circ$ at time $t = 4$ seconds and the fault conditional probability rises sharply to 1.

The two fault omens of the suction foot control are represented by

$c_1 : \{\text{The suction foot is not firmly attached to the object surface}\}$

$c_2 : \{\text{The measured values of the pressure sensor are changeless at all times}\}$

The fault causes are indicated by

$A_1 : \{\text{The vacuum pump fault}\}$

$A_2 : \{\text{The touch sensor fault (includes the touch sensor 1, the touch sensor 2, or the touch sensors 1 and 2)}\}$

$A_3 : \{\text{The pressure sensor fault}\}$

The logic rules are denoted by

$R_1 : A_1 \rightarrow c_1c_2$

$R_2 : c_1 \rightarrow A_1 + A_2$

$R_3 : c_2c_1 \rightarrow A_3$

$R_4 : A_1 \rightarrow c_2$

$R_5 : c_2 \rightarrow A_1 + A_2 + A_3$

We assume that both of the two touch sensors have not faulted. Then based on the Boolean truth table (Table II), the decision function $E(\cdot)$ is given by
\[ E = \overline{A_A}A_2A_3c_1c_2 + \overline{A_A}A_2A_3c_1c_2 + A_A\overline{A_2}A_3c_1c_2 \]
\[ + A_A\overline{A_2}A_3c_1c_2 \]

If the fault omens \( c_1 \) and \( c_2 \) have occurred, we could deduce that the fault cause is \( A_A \), i.e., the vacuum pump fault.

\[ A_A\overline{A_3} + A_A\overline{A_3} = A_A \]

V. CONCLUSIONS

This paper presented the MMBLR (multiple-model and Boolean logic reasoning) approach to sensor and actuator fault detection and identification in the suction foot control of the bipedal climbing robot. For this control system, the fault models of the two touch sensors are easily given by the robot kinematics. Moreover, we have known the certain logic relations of the system states in advance. Then based on the combination of the MMAE algorithm and the Boolean logic reasoning, the MMBLR approach is properly fit for the FDI in the suction foot control. The experimental results validated that the faults of the sensors and actuators in the robot can be easily detected and identified by the MMBLR approach.

REFERENCES


<table>
<thead>
<tr>
<th>( \text{Causes and omens} )</th>
<th>( C^1 )</th>
<th>( C^2 )</th>
<th>( C^3 )</th>
<th>( C^4 )</th>
<th>( C^5 )</th>
<th>( C^6 )</th>
<th>( C^7 )</th>
<th>( C^8 )</th>
<th>( C^9 )</th>
<th>( C^{10} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_A )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( A_A )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( A_A )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( \text{Logic rules} )</th>
<th>( A_Ac_1c_2 )</th>
<th>( c_1A_2 + A_2 )</th>
<th>( c_2 \rightarrow A_A )</th>
<th>( A_Ac_2 )</th>
<th>( c_1A_2 + A_2 )</th>
<th>( E(\cdot) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE II

Booelan Truth Table