

# 最大エントロピー原理を用いた時間付きマルコフモデルによる 事象駆動系の故障診断

齋藤 光生\*・鈴木 達也\*  
稲垣 伸吉\*・青木 猛\*\*

Fault Diagnosis of Event-Driven Systems based on Timed Markov Model  
with Maximum Entropy Principle

Mitsuo SAITO\*, Tatsuya SUZUKI\*,  
Shinkichi INAGAKI\* and Takeshi AOKI\*\*

This paper presents a new fault diagnosis technology for event-driven controlled systems like PLC. The controlled plant is modeled by means of the Timed Markov Model, which regards the time interval between successive two events as a random variable. In order to estimate the probability density functions of the randomized time intervals, the maximum entropy principle is introduced, which can estimate probability density functions so as to maximize the uniformity with satisfying the constraints caused by measured data. Then, the fault diagnosis algorithm, which returns the probabilistic diagnosis results, is developed. Finally, the usefulness of the proposed strategy is verified through some experimental results.

**Key Words:** fault diagnosis, Timed Markov Model, event-driven system, maximum entropy principle

## 1. はじめに

PLC (Programmable Logic Controller) に代表される事象駆動型システムは各種生産設備の自動制御に用いられており、制御器全体の約 8 割とわけて高い割合を占めている。これら生産設備に対する要求は従来の高速、高効率といった観点から、安全、高信頼といった要求に変わりつつあり、これらの要求にこたえるシステム設計論の確立が急務となっている。このような背景の下、システムの故障検出や診断、ならびに故障からの復旧などのためのさまざまな方策が提案されている<sup>2), 4), 5)</sup>。

工場の自動搬送ラインのような大規模システムを対象とする場合、連続時間系よりも離散事象系の表現に基づいたシステムモデルを出発点にしたほうが、結果的に効率の良い故障検出・診断アルゴリズムを得られる場合が多い<sup>1), 3)</sup>。これは、システムの構成要素が多くなった場合、詳細な連続時間系モデルよりも、構成要素間の因果関係のみに着目した離散事象

系モデルのほうが必要な情報をコンパクトに表現できるためである。中でも、文献<sup>1)</sup>では、観測される入出力ラベルの時系列パターンを時間付きマルコフモデルで表現し、その表現に基づいたシステムの故障診断法を提案している。この手法は、入出力が ON/OFF などの離散化された信号で、プラントの運転データをある程度観測でき、かつ対象とするシステムが大規模になった場合、特に有用であると考えられる。これは、上記のようなシステムの場合、確定的なモデルを用いるよりもイベント間の生起時間間隔を統計的に表現した確率的なモデルを用いるほうが、不確実性を許容している点で、より現実的なモデルが得られるからである。しかしながら、文献<sup>1)</sup>では、時間付きマルコフモデルにおける本質的パラメータである、イベント生起間隔に関する確率密度関数の明確な推定法が示されていない。また、文献<sup>1)</sup>で示されている例題は、実機での検証が行なわれておらず、故障時の運転データ不足など実際に起こりうる問題を考慮していない。

以上の点をふまえて本論文では、まず最大エントロピー原理<sup>6), 7)</sup>に基づいた時間付きマルコフモデルにおけるイベント生起間隔に関する確率密度関数の推定手法を提案する。最大エントロピー原理は、不完全なデータから、ある制約条件のもとでエントロピーが最大となる確率密度関数を推定する手法であるが、これを用いることで、モデル化の際に、学習データ外の未知データに対して適当な確率値を割り当てることが出来るというゼロ頻度問題に対処することが可能となる。

\* 名古屋大学大学院工学研究科 名古屋市中種区不老町

\*\* 名古屋市工業研究所 名古屋市熱田区

\* Graduate School of Engineering, Nagoya University,  
Chikusa-ku, Nagoya

\*\* Nagoya Municipal Industrial Research Institute,  
Atsuta-ku, Nagoya

(Received November 4, 2005)

(Revised April 19, 2006)

つぎに、自動搬送ラインを対象として提案する故障診断法の有用性を検証する。自動搬送ラインは多くの電気・電子部品や自動車産業などの量産工場などにおいて導入がなされている現実的なシステムの一例である。

本論文ではまず2章で対象とする故障診断の問題について述べる。つぎに、3章で時間付きマルコフモデルに基づいたシステムのモデル化方法について述べ、4章で最大エントロピー原理を用いた確率密度関数の推定手法について説明し、さらに、5章では故障診断の流れを具体的に記述する。最後に6章で提案手法を自動搬送ラインを模擬した実機に適用し、その有用性を示す。

## 2. 問題設定

まず、本論文で扱う故障診断の問題を以下に記す。

*Given* : 出力系列  $\mathbf{E}_t(0 \dots t_h)$

*Find* : 故障  $r$

つまり、時刻0から $t_h$ までの出力系列が得られる状況で、得られた出力系列のみを用いて故障診断を行なう。

ここで出力系列はつぎのように表記する。

$$\mathbf{E}_t(0 \dots t_h) = (e_0, t_0; e_1, t_1; \dots; e_H, t_H), t_H \leq t_h. \quad (1)$$

$e_H$  は  $H$  番目に生じたイベント、 $t_H$  はそのイベントの生起時刻を表わしている。ただし、イベントとはシステムを構成するセンサやアクチュエータの ON/OFF などに当たる。

## 3. 時間付きマルコフモデルに基づいたシステム表現

本論文では、時間付きマルコフモデル (Timed Markov Model) を用いて故障診断を行なう。時間付きマルコフモデルとは、状態遷移確率が時間に依存するマルコフモデルである。つまり、ある状態からつぎの状態へと遷移するまでの滞在時間に応じて、遷移確率が変化する。時間付きマルコフモデルの詳細については文献<sup>1)</sup>を参照されたい。以下では、時間付きマルコフモデルに基づいたシステムの表現方法について説明する。

時間付きマルコフモデルによってシステムを表現する際に必要となる関数を以下に定義する。

**【定義 1】** イベント生起間隔に関する確率密度関数

$$f_{e_{H+1}e_H}(r, w)$$

確率密度関数  $f_{e_{H+1}e_H}(r, w)$  は、故障  $r$  が生じている状況のもとで、イベント  $e_H$  のつぎにイベント  $e_{H+1}$  が生起するまでの時間間隔  $w$  に関する確率密度分布を表わしている。

**【定義 2】** 確率分布関数  $F_{e_H}(r, w)$

確率分布関数  $F_{e_H}(r, w)$  は、故障  $r$  が生じている状況のもとで、イベント  $e_H$  の生起によって状態遷移をしてから時間  $w$  経過した時点ではつぎのイベント  $e_{H+1}$  が生じていない確率を表わしている。 $F_{e_H}(r, w)$  は、確率密度関数  $f_{e_{H+1}e_H}(r, w)$

を用いて以下のように定義する。

$$F_{e_{H+1}e_H}(r, w) = \int_0^w f_{e_{H+1}e_H}(r, w) dw$$

$$F_{e_H}(r, w) = 1 - \sum_{e_{H+1} \in \mathcal{E}} F_{e_{H+1}e_H}(r, w)$$

ただし、各パラメータは以下のとおりである。

$t_h$  : 現在の時刻

$e_H$  :  $H$  番目のイベント

$t_H$  :  $H$  番目のイベントが生じた時刻

$w$  : 前イベントからの待ち時間 ( $w = t_h - t_H$ )

$\mathcal{E}$  : イベントの集合

連続して生起する2つのイベントごとに確率密度関数  $f_{e_{H+1}e_H}(r, w)$  を、すべてのイベントに関して確率分布関数  $F_{e_H}(r, w)$  を作成することでシステムの挙動を表現することが可能となる。

## 4. 最大エントロピー原理を用いた確率密度関数の推定

前章で示したように、時間付きマルコフモデルを用いてシステムを表現するには、イベント生起間隔に関する確率密度関数  $f_{e_{H+1}e_H}(r, w)$  を事前に推定する必要がある。本論文では、正常な場合およびあらかじめ故障を意図的に起こした状況のもとで、イベント生起間隔のデータ (学習データ) を取得して頻度分布 (ヒストグラム) を作成し、それから確率密度関数を推定する。ただし、実際の工場などでは過去に故障が生じたときのデータを蓄積しておくことでこれに対応でき、意図的に故障を起こしてデータを取る必要はないと考えられる。

一般に、単純に得られた学習データのみから確率密度関数を作成すると、学習データ外の未知データに対して適当な確率値を与えることが不可能となるゼロ頻度問題が生じてしまう。学習データのサンプル数を増やすことで対応することも可能であるが、それだけでゼロ頻度問題を解決することは現実的でない。そこで本論文ではゼロ頻度問題を解決するために最大エントロピー原理 (maximum entropy principle)<sup>6), 7)</sup>を用いる。最大エントロピー原理により、観測データと矛盾しないという制約のもとでエントロピーが最大となるような確率密度関数  $f_{e_{H+1}e_H}(r, w)$  を推定することが可能となる。エントロピーは「あいまいさ・不確実性」の尺度であり、その減少は学習データから得られる情報 (制約条件) 以外の何らかの情報が考慮されたことを意味し、制約条件以外の情報が全く得られていない状況のもとでは、予測が出来ない以上、制約条件のもとでエントロピーが最大となるような分布を選択することが最も尤もらしい。結果として、エントロピーが最大となるような確率密度関数を推定することで、制約条件に考慮されない部分、つまり学習データ外の未知データに対し、なるべく一様に確率を割り振ることが可能となり、これにより、ゼロ頻度問題を解決することが可能となる。

ゼロ頻度問題の代表的な解決手法としてスムージングと関

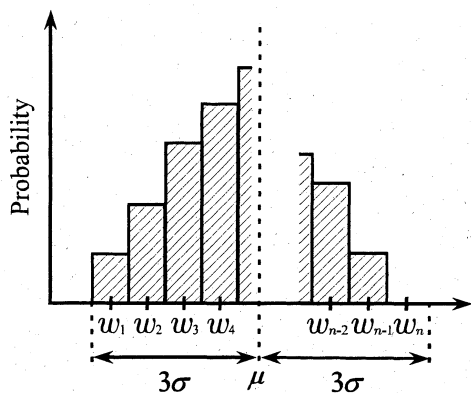


Fig. 1 Time interval of event transition

数近似が挙げられる。ただし、本論文で提案する故障診断アルゴリズムに対して、確率密度関数はヒストグラムの形式よりも関数の形式のほうが都合が良いため、ゼロ頻度問題の解決手法として関数近似、つまり確率密度関数の推定を選択した。また、確率密度関数推定の手法には、パラメトリック回帰やニューラルネットワークを用いた手法があるが、これらの手法では事前に関数型を指定する必要がある。一方、本論文で提案する最大エントロピー原理を用いた推定手法では、事前に特定の関数型を指定する必要がなく、(4)式のように指数関数に帰着されるため、表現能力が高く、さまざまな分布に適用できる。

つぎに、最大エントロピー原理を用いた確率密度関数の推定方法について説明する。まず、得られた学習データからヒストグラムを作成する。つぎに、作成したヒストグラムの各帯の中心の値を順に  $\{w_1, w_2, \dots, w_n\}$  とし、そのときの確率値も順に  $\{f_{e_{H+1}e_H}(r, w_1), f_{e_{H+1}e_H}(r, w_2), \dots, f_{e_{H+1}e_H}(r, w_n)\}$  と定める。ただし、学習データの平均値を  $\mu$ 、分散を  $\sigma^2$  とすると、 $\mu \pm 3\sigma$  内に未知データが存在するものとして  $w_n$  を決定する。この例を Fig. 1 に示す。

最大エントロピー原理では以下のようなエントロピー最大化問題を取り扱う。

Find  $f_{e_{H+1}e_H}(r, w_i)$  which maximize

$$S = - \sum_{i=1}^n f_{e_{H+1}e_H}(r, w_i) \log f_{e_{H+1}e_H}(r, w_i). \quad (2)$$

subject to

$$\begin{aligned} \sum_{i=1}^n f_{e_{H+1}e_H}(r, w_i) &= 1. \\ \sum_{i=1}^n w_i^k \cdot f_{e_{H+1}e_H}(r, w_i) &= a_k \quad (k = 1, 2, \dots, m). \\ f_{e_{H+1}e_H}(r, w_i) &\geq 0. \end{aligned} \quad (3)$$

ここで、 $a_k$  はモーメントを表わし、次数  $k (= 1, 2, \dots, m)$  に対して  $a_k = E[w^k]$  である ( $E[*]$  は期待値)。 (3) 式における 3 つの制約条件のうち 1 目と 3 目は確率変数としての制約を表わし、2 目の制約は観測データと矛盾しない

ための制約を表わし、学習データから得られる母数の情報としてモーメントを用いるのは自然である。そして、この最大化問題をラグランジュの未定乗数法を用いて解くことにより  $f_{e_{H+1}e_H}(r, w_i)$  が (4) 式のように推定できる。 (4) 式の導出過程は付録 A に示す。

$$f_{e_{H+1}e_H}(r, w_i) = \exp[-\lambda_0 - \lambda_1 w - \lambda_2 w^2 - \dots - \lambda_m w^m]. \quad (4)$$

ここで、 $\lambda_0$  は (5) 式、 $\lambda_1, \dots, \lambda_m$  はラグランジュ乗数であり、その導出過程は付録 B に示す。

$$\lambda_0 = \log \left( \sum_{i=1}^n \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right) \right). \quad (5)$$

ただし、推定結果は未知データの存在範囲  $\mu \pm 3\sigma$  に関しのみ適用し、それ以外の範囲では、確率密度関数の値を正常な場合（故障が生じていない場合）のモデルではゼロ、故障を想定したモデルでは微量  $\varepsilon$  とする。微量  $\varepsilon$  を設定する理由は、時間と共に故障の生起確率を計算してゆく中で、故障の生起確率がゼロとなってしまう、故障候補から完全に除外されることを防ぐためである。

ここで、確率密度関数の次数  $m$  の決定は重要な課題であり、確率密度関数の推定精度に大きな影響を与える。本研究では、制約条件として加えるモーメント数  $m$  の決定方法として、文献<sup>6)</sup>で示されている MinMax 値 ((6) 式) を用いる。

$$I(C_k) = [S_{\max}(N) - S_{\min}(N)] - [S_{\max}(C_m) - S_{\min}(C_m)] \quad (6)$$

ただし、 $S_{\max}(C_m)$ 、 $S_{\min}(C_m)$  はそれぞれ、制約条件  $C_m$  ( $m$  次モーメントまでを考慮) のもとでの最大エントロピー、最小エントロピーであり、 $N$  は自然拘束を表わす。そして、MinMax 値の変化率をモーメント次数  $m$  の決定指標の一つとする。

## 5. 故障診断手法

### 5.1 故障診断に用いる評価式

故障診断には、(7) 式の評価式を用いる。

$$p_M(r, t_h) = \text{Prob}(r | \mathbf{E}_t(0 \dots t_h)) \quad (7)$$

(7) 式は、イベント系列  $\mathbf{E}_t(0 \dots t_h)$  が得られたときに故障  $r$  が生じている確率を表わしている。これは故障  $r$  を想定したモデルとの適合度のようなものであり、故障  $r$  を想定したモデルによって生成されるイベント系列と実際に観測されたイベント系列とを比較することで得られる。そして、(7) 式の算出には、Case(a)：時刻  $t_h$  にイベントが観測されない場合、Case(b)：時刻  $t_h$  にイベントが観測された場合、に分けて考える必要がある。Case(a),(b) における時刻とイベントの関係を Fig. 2 に示す。

まず、候補とする故障の数を  $n_R$  とすると、時刻 0 のときはすべての故障が等確率で生じていると考え、初期値を (8)

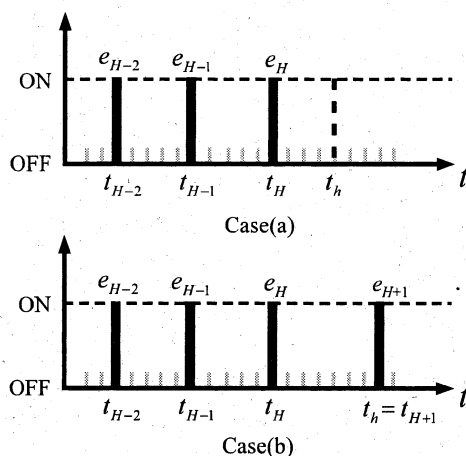


Fig. 2 Time and events in cases (a) and (b)

式のように定める.

$$p_M(r, 0) = \frac{1}{n_R}. \quad (8)$$

つぎに以下のような中間変数  $p_a(r, t_h)$  を導入する.

Case(a): 時刻  $t_h$  にイベントが観測されない場合

$$p_a(r, t_h) = F_{e_H}(r, t_h - t_H) \cdot p_M(r, t_H). \quad (9)$$

Case(b): 時刻  $t_h$  にイベントが観測された場合

$$p_a(r, t_h) = f_{e_{H+1}e_H}(r, t_h - t_H) \cdot p_M(r, t_H). \quad (10)$$

中間変数  $p_a(r, t_h)$  を用いると評価式は (11) 式のようになる.

$$p_M(r, t_h) = \frac{p_a(r, t_h)}{\sum_{r \in R} p_a(r, t_h)}. \quad (11)$$

ただし、各種関数およびパラメータは 3 章で示したとおりであり、 $R$  は故障候補の集合である.

ここで (9)~(11) 式は直感的に以下のように解釈できる. 中間変数  $p_a(r, t_h)$  は時刻  $t_h$  に故障  $r$  が生じている割合を示しており、Case(a) ではイベントが生じていないので  $F_{e_H}(r, t_h - t_H)$  を、Case(b) ではイベントが生じた瞬間であるから  $f_{e_{H+1}e_H}(r, t_h - t_H)$  を、一つ前のイベントが生じた時刻の評価値  $p_M(r, t_H)$  にそれぞれ乗じたものである. そして全故障に対して和をとったもので中間変数を正規化することにより、時刻  $t_h$  において故障  $r$  が生じている確率を算出することが可能となる.

## 5.2 故障診断法

本節では、前節で導出した評価式を利用した故障診断法について説明する.

観測されたイベント系列が故障  $r$  を想定したモデルによって生成されたイベント系列に含まれない場合、想定した故障  $r$  が生じていないと判断する. つまり、 $p_M(r, t_h) < \xi$  ( $\xi$ : 閾値) であれば故障  $r$  を『偽』と考え、故障  $r$  が生じていないと判断して故障候補から除外する. そして、同様の手順を別の故障を想定したモデルすべてに当てはめ、徐々に故障候補

を減らしていき、実際に生じている故障の検出・同定を行う方法が本論文で提案する故障診断法である.

具体的にどのような場合に故障を検出・同定したと判断できるか以下に述べる. まず故障の候補として、故障なしを示す正常  $r_0$  および故障  $r_1, r_2, \dots, r_n$  を考える. また、閾値  $\xi$  は十分小さな値をヒューリスティックに決定する.

### 故障検出

$p_M(r_0, t_h) < \xi$  ならば、故障なしを示す正常  $r_0$  が『偽』となり、何らかの故障が生じていると判断することができる. よって、故障の検出のみを行なうのであれば、正常な場合のモデルのみを生成することで可能となる.

### 故障同定

$p_M(r_1, t_h) < \xi$  ならば、ある故障  $r_1$  が『偽』となり、 $r_1$  を故障候補から除外する. ほかの故障候補に関しても評価値を算出し、その値が閾値以下となれば同様に候補から除外する. そして、すべての候補に関して計算し、ある故障  $r_k$  に関してのみ  $p_M(r_k, t_h) \geq \xi$  であれば、 $r_k$  だけが故障候補として残るので、故障  $r_k$  を同定することが可能である. 故障候補を 1 つに絞れない場合は、評価値を比較し、最も大きな値を取る故障候補を実際に生じている故障とみなす. よって、閾値  $\xi$  を適当な大きさに設定しておけば、 $\xi$  の大きさによって診断結果が変更されることはない.

また、事前に想定した故障以外のものが生じた場合、その故障を想定したモデルが存在しないので、生じている故障を同定することは不可能である. ただし、その場合には必ず  $p_M(r_0, t_h) < \xi$  となることから前述した故障検出を考慮すると、何らかの故障が生じていると判断することが可能である. よって、想定外の故障が生じた場合でも少なくとも故障検出は可能であることがわかる. そして実際の工場などではこのときのデータを蓄積し、また新たな故障のモデルを作成することで、その新たな故障を同定することが可能となる.

## 6. 自動搬送ラインへの適用例

本節では、自動搬送ラインに対して故障診断の実験を行ない、提案する故障診断法の有用性を検証する. まず、対象とする自動搬送ラインについて説明し、つぎに確率密度関数の推定例および故障診断の実験結果を示す.

### 6.1 実験の対象とする自動搬送ラインについて

故障診断の実験の対象とする自動搬送ラインの実機を Fig. 3 に示す. Fig. 3 で使用されている PLC は、KEYENCE 社製 KZ-A500 であり、接点命令の処理速度は平均  $0.44\mu s$ , 入力リレーはそれぞれ 2048 点を有する. 本実験装置は、現実の生産システムで用いられているセンサ郡、アクチュエータ郡を比較的忠実に再現したものであり、本実験を通して有用性が検証できれば、現実のシステムに適用する際も有用な結果が得られると期待できる. つぎに、実験で対象とした部分の模式図を Fig. 4 に示す. これは Fig. 3 の並列する 2 つのラインの一方であり、ベルトコンベヤ 2 本 (L1, L2) とクレーン 2 機 (CLANE1, CLANE2) でワークを搬送するシステム

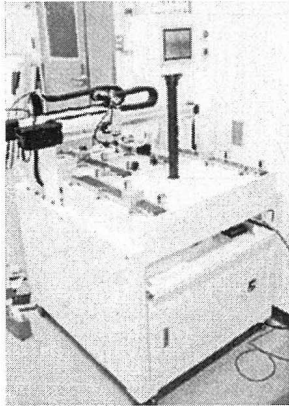
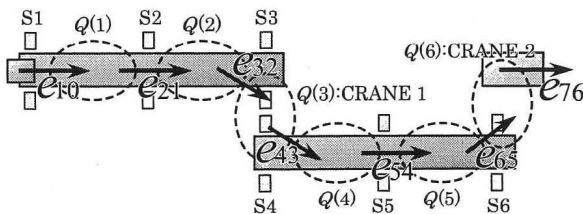


Fig. 3 Automatic transfer line

Fig. 4 State  $Q(i)$  and events

である。

また、ベルトコンベヤの両端と中央には、それぞれベルトコンベヤを挟むようにセンサ (S1~S6) が取り付けられており、各センサ間およびクレーンが移動する空間を順に状態  $Q(i) (i=1, 2, \dots, 6)$  と考える。このように状態を捉え、イベントをつぎのように定義する。時刻  $t_h - \delta t$  には状態  $Q(i)$  に存在しており、時刻  $t_h + \delta t$  に状態  $Q(j)$  に状態遷移したとき、時刻  $t_h$  にイベント  $e_{ji}$  が生じたとする。ただし、 $\delta t$  は微小時間である。本論文では、センサ間を状態と定義したことにより、ワークがセンサを横切ること (センサ ON) がイベントに相当する。

## 6.2 故障候補

5章で述べたように、故障の同定まで行なうには事前に故障候補を複数想定しなければならない。そこで、本論文で扱う故障の候補として以下のような故障を想定する。

$r_0$  : 故障なし (正常)

$r_1$  : センサ S5 が 20% の確率で無反応 (確率的な故障)

$r_2$  : ベルトコンベヤ L2 の速度低下

故障候補  $r_1$  のセンサ S5 が 20% の確率で反応しないとは、配線不良などによってセンサが反応しない場合が生じるという故障であり、センサが完全に反応しなくなる故障の初期段階を想定している。故障候補  $r_2$  とは、ベルトコンベヤ L2 の速度が通常時の 8.3cm/sec から、アクチュエータの疲労などにより 7.2cm/sec に低下することを想定した故障である。低下後の速度は、ベルトコンベヤ L1 と同じ速度を設定した。

## 6.3 確率密度関数の推定結果

本節では、4章で説明した最大エントロピー原理に基づい

た確率密度関数の推定結果を示す。前節で示したすべての故障候補に関して、連続して生起する 2 つのイベントごとに確率密度関数を推定する。まずイベント間の待ち時間のデータを取得するため、以下の条件で実験を行ない、学習データを取得した。

### 実験条件

- (1) 同じベルトコンベヤ上にワークが同時に 2 個存在しないように一定間隔 (約 14sec) でワークを投入し続ける。
- (2) 時刻 0 においてライン上にワークは存在していないものとする。 (初期状態)
- (3) システム内に同時に存在しているワークは最大 2 個とする。
- (4) データ取得のサンプリングタイムは 0.1sec とする。

実験条件より、システム内に同時に存在しているワーク数は最大 2 個であることから、考えられるイベントの組み合わせは全部で 42 通りである。しかし、そのすべての組の確率密度関数が必要なわけではない。イベント間の因果関係を考慮することにより、あらかじめ除外できるイベントの組を挙げておくことができ、確率密度関数を効率よく求めることが可能である。上述した実験条件のもとでイベント間の因果関係を考慮すると、16 通りの確率密度関数のみを推定するだけでよいことがわかる。

つぎに、推定した確率密度関数の例 ( $f_{e_{65}e_{21}}(r_0, w)$  および  $f_{e_{54}e_{10}}(r_2, w)$ ) を示す。Fig. 5, Fig. 6 のヒストグラムがそれぞれ事前実験によって得られた学習データである。このヒストグラムの各帯の中心の値とその確率値を用いて、各モーメントを求めた (Table 1)。

Table 1 Moment values

Moment constraints	$f_{e_{65}e_{21}}(r_0, w)$	$f_{e_{54}e_{10}}(r_2, w)$
$N$ (Natural constraint)	1.000000	1.000000
$a_1$ (1st moment)	0.854677	1.196355
$a_2$ (2nd moment)	1.089569	1.575832
$a_3$ (3rd moment)	1.667405	2.202415
$a_4$ (4th moment)	2.851862	3.215130

Table 2 Max entropy, minimum entropy and minmax values for  $f_{e_{65}e_{21}}(r_0, w)$ 

Constraint sets	MaxEnt	MinEnt	MinMax
$\{ N \}$	1.94591	0.00000	0.00000
$\{ N, a_1 \}$	1.72943	0.61688	0.83336
$\{ N, a_1, a_2 \}$	1.65292	0.85502	1.14801
$\{ N, a_1, a_2, a_3 \}$	1.65251	1.20027	1.49367
$\{ N, a_1, a_2, a_3, a_4 \}$	1.61606	1.29006	1.58991

また、各制約条件のもとでの最大エントロピー、最小エントロピーおよび MinMax 値 ((6) 式) を求めた。  $f_{e_{65}e_{21}}(r_0, w)$  に対する結果を Table 2 に示す。Table 2 の MinMax 値の変化率、計算量および分布の形状を考慮して、制約条件には 2 次モーメントまでを用いることにした。  $f_{e_{54}e_{10}}(r_2, w)$  に関

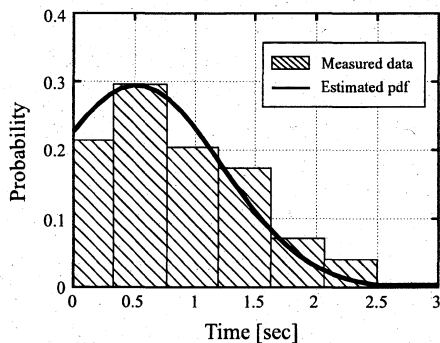


Fig. 5 Measured data and estimated probability density function ( $f_{e_{65}e_{21}}(r_0, w)$ )

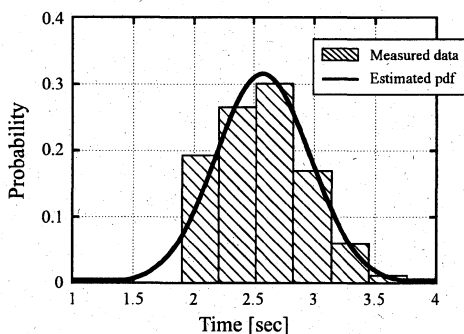


Fig. 6 Measured data and estimated probability density function ( $f_{e_{54}e_{10}}(r_2, w)$ )

しても同様である。

そして、これらの値を制約条件 (3) 式として利用する。さらに、(2) 式のエントロピー最大化問題をラグランジュの未定乗数法を用いて解いた結果がそれぞれ (12), (13) 式となる。また、Fig. 5, Fig. 6 の実線がその推定結果であり、得られた学習データのヒストグラムを上手く再現できていることがわかる。特に Fig. 6 では、学習データには存在しなかった  $1.41 \leq w \leq 1.90$  の範囲に適当な確率値を付加することが出来ていると考えられ、提案した最大エントロピー原理に基づいた確率密度関数の推定手法がゼロ頻度問題に対して有効であることがわかる。

$$f_{e_{65}e_{21}}(r_0, w) = \begin{cases} \exp(-2.184 + 2.237w - 1.348w^2) & (0 \leq w \leq 2.51) \\ 0 & (2.51 < w) \end{cases} \quad (12)$$

$$f_{e_{54}e_{10}}(r_2, w) = \begin{cases} \exp(-22.67 + 16.73w - 3.250w^2) & (1.41 \leq w \leq 3.72) \\ \varepsilon & (else) \end{cases} \quad (13)$$

#### 6.4 故障検出結果

本節では、故障検出の実験結果を示す。ここでは、意図的に起こす故障としてワークの脱落を想定した。この故障は想定外の故障であるが、このような場合でも故障の検出は可能となる。実験条件としては、確率密度関数の推定を行なった条件に対し、4 番目に投入したワークが状態  $Q(5)$  にあると

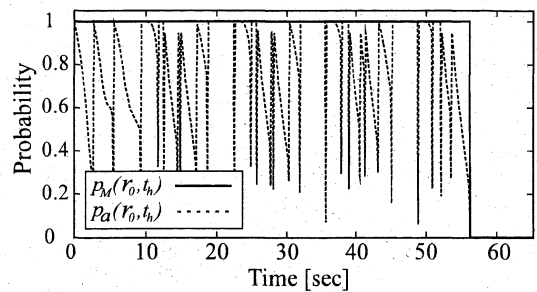


Fig. 7 Detection of fault

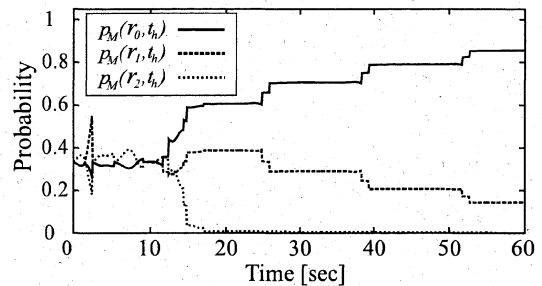


Fig. 8 Diagnosis of fault  $r_0$

き (56sec 過ぎ) にライン上から脱落させた。また、確率密度関数の微少量  $\varepsilon$  の値を 0.01, 評価値の閾値  $\xi$  を 0.01 とし、実験を行なった。

故障検出結果を Fig. 7 に示す。Fig. 7 において、56sec 過ぎに意図的にワークを脱落させたことにより、その数秒後に生じするはずのイベントが生じせず、評価値  $p_M(r_0, t_h)$  が 0 となり閾値を下回る。よって、正常  $r_0$  が『偽』であると判断され、何かしらの故障を検出したと判断できる。

#### 6.5 故障同定結果

故障の同定のための実験条件としては、確率密度関数推定を行なった条件に対し、ワークを 5 個搬送し終えた時点を実験終了とする。さらに、故障検出の実験と同様に微少量  $\varepsilon$  の値を 0.01, 評価値の閾値  $\xi$  を 0.01 とし、実験を行なった。そして、具体的な故障をつぎのように設定した。

$r_0$  : 故障なし (正常)

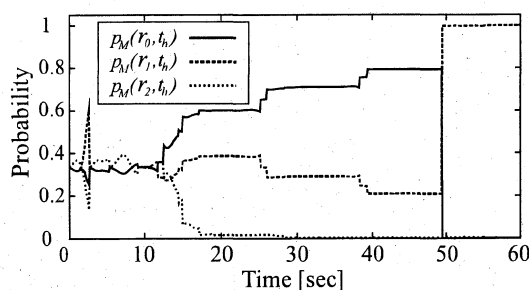
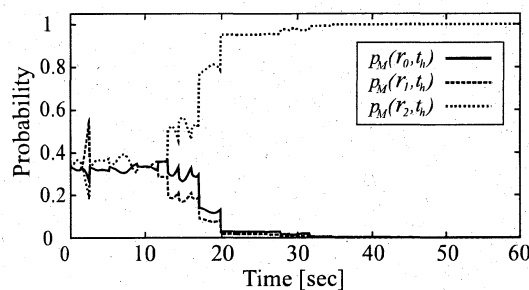
$r_1$  : 20% の確率でセンサ S5 が無反応。投入した 5 個のワークのうち 1 個 (4 番目) のみ反応しない。

$r_2$  : ベルトコンベヤ L2 の速度低下。実験開始前にベルトコンベヤ L2 の速度を低下させておく。

まず、診断開始以前に故障が生起していたときの、 $r_0, r_1, r_2$  の同定結果を Fig. 8~Fig. 10 に示す。

#### 正常 $r_0$ 同定結果

Fig. 8 は故障が生じなかった場合の結果である。まず 10sec すぎあたりから  $p_M(r_2, t_h)$  が減少していき、17sec ごろに閾値  $\xi$  を下回ったため候補から除外する。その後、ワークがセンサ S5 を横切るとにセンサ S5 が ON となり、故障  $r_1$  が『真』である確率  $p_M(r_1, t_h)$  が徐々に低下していく。最終的

Fig. 9 Diagnosis of fault  $r_1$ Fig. 10 Diagnosis of fault  $r_2$ 

に、 $p_M(r_0, t_h) > p_M(r_1, t_h)$  であるから正常  $r_0$  を『真』であるとみなし、故障が生じていないと判断できる。

#### 故障 $r_1$ 同定結果

Fig. 9 は故障  $r_1$  の同定結果である。ただし、51sec ごろにセンサ S5 をワークが通過する際にセンサが ON とならない場合である。正常  $r_0$  の同定結果と同様に、まず故障  $r_2$  の評価値が閾値を下回り、候補から除外される。そして、4 番目のワークがセンサ S5 を 51sec ごろに通過するが、故障の仮定よりセンサ S5 は ON とならない。これは正常  $r_0$  を想定したモデルが生成するイベント系列に即さないため、数秒後の約 53sec に  $p_M(r_0, t_h) = 0$  となり閾値を下回ったため、候補から除外される。最終的に故障  $r_1$  のみが候補に残り、故障  $r_1$  が唯一に同定される。

#### 故障 $r_2$ 同定結果

Fig. 10 は故障  $r_2$  が生じていた場合の結果である。実験開始時からベルトコンベヤ L2 の速度が正常時よりも遅くなっているため、初めてワークがベルトコンベヤ L2 で搬送され始める 10sec すぎから、故障  $r_2$  が『真』である確率  $p_M(r_2, t_h)$  が増加していく。その後、実験を続けていくと  $p_M(r_2, t_h)$  がさらに増加していき、逆に  $p_M(r_0, t_h)$ ,  $p_M(r_1, t_h)$  が減少していくことがわかる。そして、27sec ごろに  $p_M(r_1, t_h)$  が、31sec ごろに  $p_M(r_0, t_h)$  が閾値を下回ったため、正常  $r_0$  と故障  $r_1$  が共に候補から除外される。最終的に故障  $r_2$  のみが候補として残っていることから、故障  $r_2$  が生じていると判断できる。

#### 診断途中から故障が生じた場合の同定結果

Fig. 8~Fig. 10 は故障が診断開始以前に生起しているという仮定のもとでの結果であった。ここでは、この仮定を取り除き、故障の生起時刻によらずに故障診断が可能か否かを検

証する。

実験条件は確率密度関数の推定を行なった条件と同一である。そして、診断開始（時刻 0）以前には故障は生起しておらず、診断途中でベルトコンベヤ L2 の速度を低下させた（故障  $r_2$ ）。また、これまでの実験と同様に微量  $\varepsilon$  の値を 0.01、評価値の閾値  $\xi$  を 0.01 として実験を行なった。その結果を Fig. 11 に示す。

まず、 $p_M(r_2, t_h)$  が低下し、閾値を下回ったため故障  $r_2$  が候補から除外される。なぜなら、これまでに観測されたイベント系列が故障  $r_2$  を想定したモデルから生成される確率が低いためである。しかし、32sec ごろに故障  $r_2$  が生起したことで、その後に観測されたイベント系列が故障  $r_2$  を想定したモデルから生成されるものと一致する確率が高くなり、 $p_M(r_2, t_h)$  が増加して再び故障候補に加わる。これは、確率密度関数に微量  $\varepsilon$  を付加したことにより、評価値がゼロとならないことから可能となる。それ以降は、 $p_M(r_0, t_h)$  と  $p_M(r_1, t_h)$  が減少し続け、故障  $r_1$  が 75sec ごろ、正常  $r_0$  が 90sec ごろに候補から除外される。そして診断終了時点で候補には故障  $r_2$  のみが残っていることから、故障  $r_2$  が同定結果となる。このことから、確率密度関数に微量  $\varepsilon$  を付加したことにより、故障の生起時刻によらずに診断が可能であることが示された。

以上の結果から、すべての場合において、実際に生じている故障と本論文で提案した故障診断法による診断結果とが一致しており、提案した手法によって故障診断が可能であることが実験によって示された。

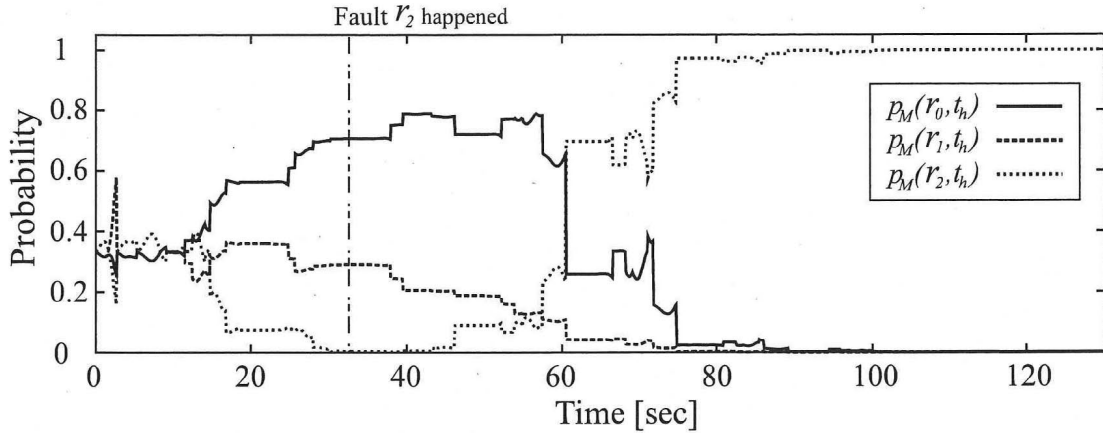
## 7. おわりに

本論文では、自動搬送ラインに対して最大エントロピー原理に基づいた時間付きマルコフモデルの推定手法を提案し、得られたモデルに基づいた故障診断法を提案した。また、実験によりその有効性を検証した。提案手法を用いることで、故障の生起時刻やワーク数に依存せずに故障診断が可能であると考えられる。また、ワーク数の増加に伴うイベント数の組み合わせ爆発を低減する手法（システムのモジュール化、状態の階層化など）の開発、提案手法では診断が不可能である故障（ソフトウェアやコントローラなど）への対応などが今後の課題である。

## 参考文献

- 1) Jan. Lunze: Diagnosis of Quantized Systems Based on a Timed Discrete-Event Model, *IEEE Trans. Syst. Man. Cybern.*, **30**-3, 322/335 (2000)
- 2) D.N. Pandalai and L.E. Holloway: Template Languages for Fault Monitoring of Timed Discrete Event Processes, *IEEE Trans. Automa. Contr.*, **45**-5, 868/882 (2000)
- 3) S. Hashtrudi Zad, R.H. Kwong and W.M. Wonham: Fault Diagnosis in Timed Discrete-Event Systems, *Proc. 38th IEEE Conf. Decision Contr.*, 1756/1761 (1999)
- 4) F. Schiller, J. Schroder and J. Lunze: Diagnosis of transient fault in quantised systems, *Engineering Applications*



Fig. 11 Diagnosis result in the case that fault  $r_2$  happened at 32sec

of Artificial Intelligence, 14, 519/536 (2001)

- 5) A. Darwiche and G. Provan; Exploiting system structure in model-based diagnosis of discrete-event systems, In *Proc. 7th Intl. Workshop on Principles of Diagnosis*, 95/105 (1996)
- 6) Munirathnam Srikanth, H.K. Kesavan and Peter H. Roe: Probability Density Function Estimation using the Min-Max Measure, *IEEE Trans. Syst., Man, Cybern.*, 30-1, 77/83 (2000)
- 7) 辻井, 北: 確率的言語モデル, 東京大学出版会, 155/174 (1999)

## 《付 録》

### A. 確率密度関数の推定式 (4) の導出

ラグランジュの未定乗数法を用いる。ラグランジュ関数を  $\mathcal{L}(f_{e_{H+1}e_H}(r, w_i), \Lambda)$ , 未定乗数を  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$  とすると,

$$\mathcal{L}(f_{e_{H+1}e_H}(r, w_i), \Lambda) = - \sum_{i=1}^n f_{e_{H+1}e_H}(r, w_i) \log f_{e_{H+1}e_H}(r, w_i) + \sum_{j=1}^m \lambda_j \left( a_j - \sum_{i=1}^n w_i^j f_{e_{H+1}e_H}(r, w_i) \right).$$

$\Lambda$  を固定し,  $f_{e_{H+1}e_H}(r, w_i)$  に関して  $\mathcal{L}(f_{e_{H+1}e_H}(r, w_i), \Lambda)$  を最大化する。

$$\frac{\partial \mathcal{L}}{\partial f_{e_{H+1}e_H}(r, w_i)} = 0.$$

これを解くと,

$$-\log f_{e_{H+1}e_H}(r, w_i) - 1 + \sum_{j=1}^m \lambda_j w_i^j = 0,$$

$$\therefore f_{e_{H+1}e_H}(r, w_i) = \frac{1}{C} \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right) \quad (C: \text{const}). \quad (\text{A.1})$$

さらに,  $\sum_i f_{e_{H+1}e_H}(r, w_i) = 1$  より

$$\frac{1}{C} \sum_{i=1}^n \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right) = 1 \quad (\text{A.2})$$

すなわち,

$$\log C = \log \left( \sum_{i=1}^n \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right) \right) \quad (\text{A.3})$$

$\log C = \lambda_0$  と置き換えれば (5) 式が得られ, (5) 式を (A.1) に代入することで (4) が得られる。

### B. ラグランジュ乗数の導出

(4) 式を (3) 式に代入すると,

$$a_k = \frac{\sum_{i=1}^n w_i^k \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right)}{\sum_{i=1}^n \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right)} \quad (k = 1, 2, \dots, m)$$

すると, 上式のように未知数  $m$  個の非線形連立方程式が得られる。さらに,

$$R_k = a_k - \frac{\sum_{i=1}^n w_i^k \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right)}{\sum_{i=1}^n \exp \left( - \sum_{j=1}^m \lambda_j w_i^j \right)}$$

$$R = \sum_{k=1}^m R_k^2$$

とし,  $R$  を最小化する  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$  を求める。そして (5) 式に代入することで  $\lambda_0$  を求めることが出来る。

## [著 者 紹 介]

齋 藤 光 生



1982年9月15日生。2005年名古屋大学工学部機械航空工学科卒業。同年, 同大学大学院工学研究科機械理工学専攻博士課程前期課程進学, 現在に至る。



## 鈴木 達也 (正会員)



1964年1月3日生。91年名古屋大学大学院博士課程後期課程電子機械工学専攻修了。工学博士。同年名古屋大学工学部助手。2000年同助教授。2006年同教授、現在に至る。この間、98年から1年間、U.C.Berkeley 客員研究員。ハイブリッドシステム論に基づく制御、診断、行動解析などに関する研究に従事。95年電気学会論文賞、2004年SICEシステム情報部門学術講演会論文賞などを受賞。電気学会、電子情報通信学会、日本機械学会、システム制御情報学会、日本ロボット学会、日本人間工学会、IEEEの各会員。

## 稲垣 伸吉 (正会員)



2000年名古屋大学大学院電子機械工学科博士前期課程修了。同年、東京大学大学院工学系研究科博士課程入学。2003年名古屋大学大学院工学研究科機械理工学専攻助手。同年理化学研究所BMC非常勤研究員兼任。現在に至る。自律分散システム、歩行ロボット、確率的情報処理の研究に従事。IEEE, JSME, RSJ, の会員。博士(工学)。

## 青木 猛



1962年3月21日生。86年名古屋大学大学院博士課程前期課程航空工学専攻修了。同年名古屋市工業研究所入庁、現在に至る。工学博士。この間、ロボティクス、ソフトコンピューティングおよびメデイカルエレクトロニクスに関する研究に従事。日本知能情報ファジィ学会、人工臓器学会の各会員。

=====