다중 시계열 패턴 분석에 의한 소프트웨어 계측☆

Software Measurement by Analyzing Multiple Time-Series Patterns

김 계 영*
Gye-Young Kim

요 약

본 논문에서는 다중 시계열 패턴을 분석하여 계측값을 예측하는 방법에 관하여 기술한다. 본 논문의 목적은 표본패턴들 중에서 입력패턴과 가장 유사한 패턴을 찾은 다음 그 표본패턴이 가지는 실측값과의 오차율을 산출하는 것이다. 따라서 인식이 아니라 계측이며 하드웨어가 아닌 소프트웨어 기술을 제안하다. 본 논문에서 제안하는 방법은 초기화, 인식 및 계측 등의 단계로 구성된다. 초기화 단계에서는 중요도를 사용하여 인자들 각각의 가중치를 산출한다. 학습 단계에서는 수집된 표본패턴을 먼저 DTW와 LBG 알고리즘을 사용하여 각 인자별 독립적으로 군집화를 수행한 다음, 모든 표본패턴에 대하여 군집의 번호들로 구성된 코드열을 생성한다. 계측 단계에서는 입력패턴에 대한 코드열을 생성한 다음 해쉉으로 표본패턴들 중에서 같은 코드열을 가지는 표본들을 찾고, 이 표본들 중에서 입력패턴에 가장 잘 정합되는 하나의 표본을 선택하다. 최종적으로 이 패턴이 가지고 있는 실측값과 오차율을 출력한다. 성능평가는 반도체생산장치 중에서 하나인 식각장치로부터 얻어진 자료에 적용하여 수행한다.

Abstract

This paper describes a new measuring technique by analysing multiple time-series patterns. This paper's goal is that extracts a really measured value having a sample pattern which is the best matched with an inputted time-series, and calculates a difference ratio with the value. Therefore, the proposed technique is not a recognition but a measurement, and not a hardware but a software. The proposed technique is consisted of three stages, initialization, learning and measurement. In the initialization stage, it decides weights of all parameters using importance given by an operator. In the learning stage, it classifies sample patterns using LBG and DTW algorithm, and then creates code sequences for all the patterns. In the measurement stage, it creates a code sequence for an inputted time-series pattern, finds samples having the same code sequence by hashing, and then selects the best matched sample. Finally it outputs the really measured value with the sample and the difference ratio. For the purpose of performance evaluation, we tested on multiple time-series patterns obtained from etching machine which is a semiconductor manufacturing.

⌐ Keyword : Language education, Penmanship, Handwriting evaluation, Artificial neural network, Dynamic time warping

1. 서 론

뇌파(EEG), 심장박동변이(HRV)와 같은 생체 분야, 환율(FX), 주식(stock)과 같은 금융 분야 그리고 동적 시스템 제어, 교통 흐름 예측 등 많은 응용 분야에서 다루어지는 시계열(time-series)자료들은 비예측적이며 복잡한 패턴을 보인다. 이러한 시계열 자료들을 분석 및 예측에 있어서 핵심 문제는 외관상 무질서한(random) 시계열로부터 계측에 대한 사전 지식이 없이 과거의 짧은 기록을 토대로 분석 모델을 구축하고, 이를 이용하여 미래를 예측하는 것이다. 또한 과거에는 실생활에서 발생하는 방대한 자료들을 인지하지 못했으나 과학이 점점 발달함에 따라 이러한 방대한 자료들을 관리하고 수집할 수 있게 되었으며, 이들을 분석함으로써 새로운 정보를 획득하기 위한 다양한 시도가 이루어지고 있다. 또한 수많은 자료들이 서로 연관되고 영향을 미침으로 인하여 하나의

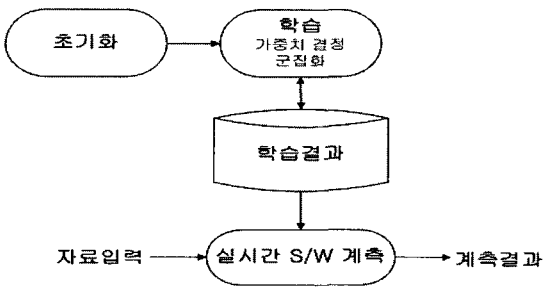
* 정 회 원 : 숭실대학교 컴퓨터학부 교수(제 1저자)
gykim@computing.soongsil.ac.kr
[2004/07/09 투고 - 2004/08/01 심사 - 2004/12/06 심사 완료]
☆ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

시계열 자료의 분석만으로는 얻을 수 없는 정보들을 다수의 시계열 자료의 분석을 통해서 얻으려는 연구가 진행되고 있다.

다중 시계열 패턴을 분석하여 계측하는 자동화된 도구를 개발하는데 있어 문제점은 기존의 패턴인식에서 사용되는 자료의 형태와 목적이 다른데 있다. 기존 패턴인식에 시계열 패턴인식의 대표적인 분야는 음성인식[1]과 온라인 필기문자인식[2]이라 할 수 있다. 이들은 하나의 시계열 패턴과 하나의 인식코드를 가지며, 같은 부류(class)의 시계열 패턴들은 같은 인식코드를 가지는 특성이 있다. 따라서 통계적 또는 비모수적 접근법[3] 이나 신경망[4] 등의 방법을 통하여 학습용 표본패턴들을 축약하고 인식하는 방법들이 다수 개발되어 있다. 그러나 본 논문의 목적인 다중시계열 패턴 분석을 통한 계측은 다음 같은 두 가지 관점에서 기존의 시계열 패턴인식과 다르다. 첫 번째는 자료의 형태가 다른 점이다. 즉, 다중 시계열 패턴이란 하나의 패턴은 다수의 인자들로 구성되며, 각 인자들이 가지는 자료가 시계열 형태를 이루는 것을 의미한다. 두 번째는 인식이 아니라 계측이라는 것이다. 즉, 기존의 시계열 패턴인식에서 입력패턴이 어떤 특정한 클래스에 속하는가를 결정하여 그 클래스의 인식코드를 출력하는 것이지만, 계측에서는 특정한 입력패턴은 어떤 결과를 산출할 것인지를 예측하는 것이다.

이 목적을 달성하기 위한 기존의 대표적인 접근 방법에는 예측 모델에 의한 방법[5, 6], 그리고 패턴인식에 의한 방법[7]이 있다. 첫 번째 방법은 처리를 위한 인자들과 측정값 사이의 관계를 함수 또는 신경망으로 표현한 다음, 입력되는 인자값으로 부터 계측값을 예측하는 방법이다. 이 방법은 두 인자들 사이의 관계를 정의하는 함수를 정확히 표현하는 것이 어려운 문제점이 있다. 두 번째 방법은 사전에 수집된 실측값과 이때의 상태자료들을 패턴인식 모델에 학습시킨 후, 입력된 상태자료와 학습된 상태자료와 비교하여 가장 유사한 상태자료가 가지는 계측값을 출력하는 방법으로 대표적인 예는 UPM(Universal Process Modeling)[7]이다. UPM에서는 인자마다의 최근 수백여 개의 표본을 취득 및 저장한 다음 입력되는 인자값을 이 자료들과 비교하여 가장 유사한 자료를 찾아서 오차율을 산출한 다음 이 오차가 사전에 정의된 범위 내에 있는지를 검사한다. 따라서 UPM은 특정한 인자값이 정의된 범위 밖으로 벗어나는 것을 감시하므로 모든 인자들에 의하여 만들어지는 최종 측정값을 계측하지 못하는 단점이 있다.

본 논문에서는 다중 시계열 데이터를 분석하여 각각의 인자들의 상태자료를 조합하여 모든 인자들에 의하여 만들어지는 최종 측정값을 계측하는 기술을 제안한다. 즉, 본 논문에서는 사전에 수집된 자료들 중에서 입력된 패턴과 가장 유사한 패턴을 찾은 다음 표본패턴의 어떤 인자가 가지는 시계열 자료와 입력패턴에서 대응하는 인자의 시계열 자료와의 차를 분석하여 표본패턴이 가지는 계측값에서 이 오차도 만큼의 오류범위를 산출하는 방법을 제안한다. 따라서 본 논문의 핵심은 방대한 량의 표본패턴들 중에서 입력패턴와 유사한 표본패턴을 효과적으로 찾는 것이다.



〈그림 1〉 제안하는 기술의 개요도

제안하는 방법은 그림 1과 같이 초기화, 학습, 그리고 계측 단계로 구성된다. 초기화 단계에서는 계측 대상이 되는 장치의 특성을 파악하는 단계로 사용되는 인자들의 종류와 중요도 및 임계값 등을 설정하고 각 인자의 가중치를 산출한다. 학습단계에서는 LBG(Line, Buzo, and Gray)[8] 알

고리즘을 사용한 인자별로 군집화를 수행하여 군집의 수와 각 군집의 대표값을 산출한 다음 학습에 사용된 표본패턴들의 각 인자가 가지는 시계열 자료가 속하는 군집의 번호들로 구성된 코드열을 생성한다. 이때, 두 시계열 패턴의 유사도를 산출하기 위하여 본 논문에서는 DTW(Dynamic Time Warping)[9]를 사용한다. 계측 단계에서는 다중 시계열 패턴을 실시간으로 입력받아 학습결과를 참조하여 코드열로 표현한 다음 해쉉을 통하여 표본패턴들 중에서 같은 코드열을 가지는 표본들을 산출한 후, 이 표본들 중에서 입력패턴과 가장 잘 정합되는 표본을 찾아서 이 표본 가지는 계측값과 오차율을 산출한다.

본 논문의 구성은 다음과 같다. 제 2절에서는 초기화에 관하여 기술한다. 제 3절에서는 DTW을 사용하여 두 시계열 패턴의 유사도를 측정하는 방법을 설명한다. 제 4절에서는 LBG를 사용하여 군집화하고 코드열을 생성하는 방법과 탐색 그리고 가중치를 사용하여 입력패턴이 가질 것으로 예측되는 계측값을 산출하는 방법에 대하여 각각 기술한다. 그리고 제 5절에서는 실험결과 및 성능평가를 보인 후, 마지막 제 6절에서는 결론 및 향후연구에 관하여 논술한다.

2. 초기화

초기화 단계에서는 분석할 장치가 가지는 인자에 대한 자료를 입력받아 인자들이 가지는 특성에 적합하게 학습과 계측을 수행하기 위하여 인자들 각각의 중요도를 결정하고 가중치 산출한다. 인자별 가중치를 산출하는 방법은 다음과 같다. 먼저, 사용자는 각 인자의 상대적인 중요도를 숫자로 입력한다. 예를 들어, 인자의 중요도를 3 종류로 나누고자 하는 경우, 매우 중요한 인자는 3을, 그 다음 2를, 중요도가 가장 낮은 경우는 1로 입력한다. 입력된 중요도를 참조하여 가중치는 식 1을 통하여 산출된다. 식 1에서 n은 중요도가 설정된 인자의 수이고, L(i)은 i 번째 인자의 중요도이다.

$$W(i) = L(i) / \sum_{i=1}^{n} L(i) \quad (1)$$

식 1에서 가중치 $W(i)$는 각 인자의 중요도를 모든 인자들이 가지는 중요도의 합으로 정규화한 값으로 0에서 1사이의 값을 가지게 된다.

본 논문에서는 입력받은 자료로부터 특징벡터를 정의할 때, 이 자료들은 서로 다른 시계열 패턴을 가지므로 이들을 정렬(alignment)하여 유사도를 구하기 위하여 DTW 알고리즘을 사용한다. 군집화를 수행할 때는 표본패턴이 속할 군집을 모르기 때문에 자율(unsupervised) 군집화 알고리즘의 하나인 LBG 방법을 사용하였다. 또한 본 논문에서 취급하는 자료는 다중 시계열 패턴이고 표본패턴들 중에서 입력패턴과 가장 유사한 패턴을 찾는 것이므로 모든 표본패턴의 코드열을 생성한다. 먼저, DTW를 사용하여 주어진 두 시계열 패턴의 유사도를 산출하는 방법에 관하여 설명하면 다음과 같다.

3. DTW에 의한 유사도 측정

본 논문에서 사용하는 자료의 형태는 동일한 시간간격 마다 변하는 시계열 패턴이다. 따라서 군집화와 계측값을 예측하기 위하여 두 개의 시계열 패턴 즉, 참조패턴(reference pattern)과 대응패턴(corresponding pattern) 사이의 유사도를 산출하여야 하는데, 패턴의 길이와 위치가 일정하지 않으므로 본 논문에서는 DTW(Dynamic Time Warping)를 사용하여 유사도를 측정한다.

참조패턴과 대응패턴을 각각 $Q = q_1, q_2, \cdots, q_n$와 $C = c_1, c_2, \cdots, c_m$라 할 때, DTW 알고리즘을 사용하여 비선형적인 대응관계로부터 유사도를 산출하는 방법은 식 2와 같다.

$$\begin{aligned} \gamma[i][j] &= d(q_i, c_j) + \min\{\gamma[i-1][j-1], \\ &\qquad \gamma[i-1][j], \gamma[i][j-1]\} \\ \gamma[1][1] &= d(q_1, c_1),\ \gamma[1][j] = \infty; (j > 1) \end{aligned} \quad (2)$$

여기서, $d(q_i,c_j)=\sqrt{(q_i-c_j)^2}$, $1 \leq i \leq n$, $1 \leq j \leq m$

식 2의 결과는 정규화되지 않아 길이 따라 유사도가 서로 다르게 산출될 뿐 아니라 유사성 클수록 적은 값을 가지는 비유사도이다. 따라서 최종적으로 다음 식 3을 사용하여 정규화된 유사도로 재산출한다.

$$D(R, C) = 1 - \gamma[n][m]/n \quad (3)$$

식 2아 3은 다음과 같은 제약사항에 근거하여 형성된 것이다. 첫째는 끝점정렬(end-point alignment) 즉, 참조패턴의 첫 번째 자료는 대응패턴의 첫 번째 자료와 반드시 대응되며, 마지막 자료도 역시 다른 패턴의 마지막 자료와 반드시 대응되어야 하는 조건이다. 둘째는 연속성(continuity)으로 인접한 셀로만 이동할 수 있는 조건이다. 셋째는 단조성(monotonicity)으로 시간축을 따라서 증가하여야 하는 조건이다. 따라서 식 2와 3의 DTW는 동적계획법(dynamic programming)의 일종으로 m× n 크기의 배열 γ의 시작위치인 $\gamma[1][1]$에서 끝위치인 $\gamma[n][m]$ 사이에 있는 수많은 대응경로들 중에서 최적의 대응경로를 탐색하는 방법을 통하여 두 패턴의 유사도를 측정하는 과정을 의미한다.

지금까지는 길이가 다른 시계열 패턴의 유사도를 산출하는 방법에 관하여 설명하였다. 다음 절에서는 군집화와 코드열 생성 및 계측에 관하여 설명한다.

4. LBG에 의한 군집화와 코드화 및 계측

계측 단계에서는 기본적으로 실시간으로 입력된 상태자료와 학습 단계에서 수집한 자료와 비교하여 유사도가 큰 시계열 패턴이 가지는 측정값을 선택한 다음 오차의 정도를 산출한다. 이때, 표본자료가 많은 경우는 가장 큰 유사도를 가지는 패턴을 찾는데 너무 많은 계산시간이 소요되는 문제점이 있다. 따라서 유사한 특성을 가지는 시계열 패턴들을 하나의 집단으로 분류하고 그 집단의 대표를 선택하는 군집화를 수행하여야 한다. 군집화를 위하여 본 논문에서는 LBG 알고리즘 사용한다. LBG 알고리즘의 처리 과정은 K-평균(mean)과 유사하다. 차이점은 군집의 대표를 선택하는 방법 즉, K-평균은 군집을 이루는 패턴들의 평균을 그 군집의 대표로 결정하는 반면, LBG는 군집에 속하는 패턴들까지 거리의 평균을 최소로 하는 패턴을 그 군집의 대표로 결정한다. 본 논문의 목적은 일반적인 패턴인식 즉, 학습된 자료 중에서 유사도가 가장 큰 패턴의 식별자를 출력하는 것이 아니라 그 패턴이 가지는 측정값과 오차를 출력하는 것이다. 따라서 군집에 속하는 시계열 패턴의 각 자료들을 평균하는 것은 측정값을 평균하는 것인데, 이것은 유사한 시계열 패턴은 유사한 측정값을 가진다는 전제하에서 성립된다. 하지만 본 논문에서 취급하는 다중 시계열 패턴은 그 형태가 유사하여도 측정값이 유사한 것은 아니다. 따라서 주어진 자료들을 가능한 변경하지 않는 방법으로 군집의 대표를 산출하여야 하므로 본 논문에서는 <알고리즘 I>과 같이 LBG를 사용하여 군집화한 다음 각 시계열 패턴을 코드를 생성한다.

<알고리즘 I : LBG에 의한 시계열 패턴의 군집화> {

Step 1. 자료 입력 : 군집화를 위하여 초기값을 다음과 같이 설정.
  1. 2개 이상의 학습용 시계열 패턴들을 입력한다.
  2. 원하는 초기 군집의 수 K(=2)를 입력한다.
  3. 군집의 최대거리 D를 입력한다.

Step 2. 중심 선택 : 각 군집을 대표하는 패턴을 임의로 선택.

Step 3. 왜곡도 설정 : 초기 왜곡도 TD'에 상당히 큰 값을 할당.

Step 4. TD = TD'.

Step 5. 유사도 계산 : 모든 학습용 시계열 패턴 각각에 대하여 다음을 수행.
  1. DTW를 사용하여 각 시계열 패턴과 선택된 중심들과 각각 대응경로를 찾음.

2. 비유사도와 대응하는 두 자료의 차의 표준 편차를 산출.
3. 두 개의 특징에 의한 산출된 거리들 중에서 가장 적은 것을 찾음.
4. if (가장 적은 거리가 D보다 적다) 이 군집에 현재 처리중인 패턴을 할당.
5. else 군집의 수 K를 1 증가 시키고, 현재의 패턴을 새로운 군집의 중심으로 설정.

Step 6. 중심 재선택 : 모든 군집 각각에 대하여 다음을 수행.

1. 중심에 속하는 패턴들 사이의 거리를 최소로 하는 S_i 패턴을 찾음.
2. 군집의 중심을 이 패턴으로 변경.
3. 이때의 거리를 왜곡도 $D(S_i)$에 저장.

Step 7. 왜곡도 산출 : K개의 군집이 가지는 $D(S_i)$를 합하여 왜곡도 TD'에 할당.

Step 8. if (|TD - TD'| > 0에 가까운 매우 적은 값) GoTo Step 5.

Step 9. 각 군집의 중심 시계열 패턴을 저장.

}

서론에서 언급한 바와 같이 본 논문에서는 다중 시계열 패턴을 사용하므로 이에 대한 군집화는 인자 마다 별도로 이루어진다. 따라서 알고리즘 I는 각 인자에 대하여 독립적으로 적용된다.

계측 단계에서는 학습된 패턴 중에서 입력된 패턴과 가장 유사한 패턴을 찾는 과정을 포함하고 있다. 그런데, 인자 마다 독립적으로 군집화를 수행하므로 계측 단계에서 인자 마다 가장 유사한 패턴을 선택하여 찾아낸 패턴은 하나의 패턴을 위한 것이 아니라 여러 개의 패턴을 조합한 결과가 되는 문제가 있다. 이 문제를 해결하기 위하여 본 논문에서는 먼저, 학습을 위하여 수집된 표본패턴을 식 4와 같이 각 인자가 속하는 군집의 번호들로 구성된 코드열로 표현한다. 군집의 번호는 각 인자별로 다음과 같이 산출된다. 알고리즘 I의 군집화에 의하여 생성된 군집의 중심들 각각과 하나의 학습용 패턴과의 유사도와 차들의 표준편차를 DTW를 사용하여 산출한 다음, 이 두 특징에 의하여 산출되는 거리가 가장 적은 군집을 선택함에 의하여 결정된다.

$$ClassListOfPattern_i = \{C\#_1, C\#_2, \ldots, C\#_n\} \quad (4)$$

학습 과정을 통하여 산출되는 정보는 인자별 군집의 수와 각 군집의 중심패턴 그리고 학습용으로 입력된 표본패턴 모두에 대하여 식 4와 같이 군집의 번호를 나열한 코드열이다. 본 논문에서는 효과적으로 이 코드열을 저장하고 검색하기 위하여 해싱(hashing)을 사용하였다. 실시간으로 입력되는 패턴을 가지고 학습된 자료를 참조하여 계측값을 산출하는 과정은 다음과 같다.

먼저, 입력된 패턴의 코드열을 생성한다. 즉, 각 인자별로 거리가 가장 가까운 군집의 번호를 산출한다. 그 다음은 코드화된 표본패턴에서 이 코드열과 동일한 패턴들 찾은 후, 이들 중에서 입력패턴과 가장 잘 정합되는 패턴을 찾은 후, 식 5와 6을 사용하여 예측되는 계측값을 산출한다.

$$R_E = \sum_{i=1}^{n} (I_V \times D(i) \times W(i)) \quad (5)$$

$$P_V = I_V \pm R_E \quad (6)$$

식 5에서, R_E는 계측 오류값을 나타내며, I_V는 가장 잘 정합되는 표본패턴이 가지는 측정값을, $D(i)$와 $W(i)$는 i번째 인자의 비유사도와 가중치를 각각 나타낸 것이다. 따라서 식 6의 계측값 P_V는 측정값에 계측 오류값을 더하거나 뺀 값이 된다.

5. 실험 및 결과

실험은 펜티엄 Ⅲ 1.2㎓ CPU와 MS Windows XP를 탑재한 PC에서 수행하였으며, 구현을 위하여 사용한 언어는 MS Visual C++이다. 실험을

위한 자료는 반도체 제조공정 중에서 식각장치로부터 획득하였고, 계측기로부터는 ACI 자료를 획득하였다. 따라서 학습할 때 입력되는 자료는 인자들 각각에 대한 시계열 자료와 ACI값으로 구성되지만, 계측할 때는 인자들 각각의 시계열 자료만을 사용하였다. 계측 결과인 출력값은 식각공정의 결과값인 ACI값으로 하였으며, 실제 실시간 실험을 할 수 없는 관계로 실험 자료의 형태는 파일 형태로 하였다. 훈련을 위하여 사용된 자료는 15개의 훈련자료군이다. 각 자료군은 18개의 인자 각각에 대하여 8단계로 구성된 시계열 자료이다.

18개의 인자들 중에서는 공정처리의 결과에 크게 형향을 미치는 인자들도 있고, 그렇지 않는 인자들도 있다. 결과에 큰 영향을 미치는 인자들을 본 논문에서는 중요인자라 한다. 또한, 각 인자들은 8단계로 구성되는데, 이들 중에서 시작부분과 종료부분의 단계들은 그 상태가 불안정하며 공정처리의 결과에 큰 영향을 미치지 않는다. 공정처리에 직접적으로 영향을 미치는 것으로 파악된 단계들을 본 논문에서는 유효단계라 한다. 중요인자와 유효단계의 선택은 반도체생산산업 현장에서 경험적으로 습득된 지식에 근거하여 결정하였다.

성능 평가는 계측용 자료는 실제 계측기로부터 측정된 ACI 값이 없다는 가정 하에 인자들의 상태정보만을 사용하여 학습량의 증가에 따른 계측 ACI 값과 실측 ACI 값 사이의 차이가 변화하는 과정과 최종적인 결과를 평가한다. 이때 사용된 계측용 자료는 총 7개이며, 실험은 중요인자만을 사용한 경우, 중요도 자료를 변경하는 경우, 유효단계만을 사용한 경우에 대하여 평가하였다. 중요인자들만을 사용한 경우는 전체 18개의 인자 중에서 약 5개의 중요인자만을 선택하여 학습량 증가에 따른 계측 ACI 값의 정확도가 향상되는 정도를 평가하였다. 중요도 자료를 변경하는 경우에 대한 평가는 가중치가 계측결과에 미치는 영향을 평가하기 위하여 임의의 중요도를 사용함에 따른 계측 결과의 분석한다. 유효단계만을 사용한 경우에 대한 평가는 총 8개 단계 중에서 공정결과와 밀접한 관련이 있는 단계라 판단되는 3단계와 4단계 자료만을 사용하여 평가하였다.

표 1은 7개의 실험자료 F6735X, F674EX, F681YX, F70QAX, F791PX, F791TX, F796UX에 대

〈표 1〉 학습량 증가에 따른 실측 ACI 값과 계측 ACI값의 차

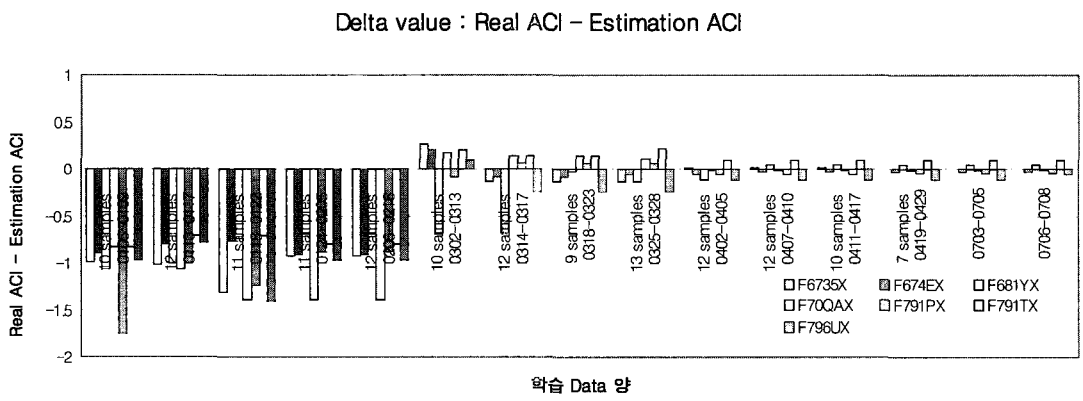
학 습		계측 ACI값			Delta Value(실측ACI - 계측 ACI)		
횟수	실측 ACI값	F6735X	F674EX	F791PX	F6735X	F674EX	F791PX
3차	F6735X : 9.217 F674EX : 9.152 F791PX : 9.139	10.528857	9.931291	9.997443	-1.311857	-0.779291	-0.858443
4차		10.140691	10.063959	9.936808	-0.923691	-0.911959	-0.797808
5차		10.140691	10.063959	9.936808	-0.923691	-0.911959	-0.797808
6차		8.949999	8.948253	9.936808	0.267001	0.203747	-0.797808
7차		9.347746	9.238796	9.936808	-0.130746	-0.008796	-0.797808
8차		9.347746	9.238796	9.284006	-0.130746	-0.008796	-0.145006
9차		9.347746	9.212985	9.391124	-0.130746	-0.060985	-0.252124
10차		9.200812	9.212985	9.379872	0.016188	-0.060985	0.240872
11차		9.200812	9.176999	9.207305	0.016188	-0.024999	-0.068305
12차		9.200812	9.176999	9.207305	0.016188	-0.024999	-0.068305
13차		9.208558	9.176999	9.207305	0.008442	-0.024999	-0.068305
14차		9.208558	9.176999	9.207305	0.008442	-0.024999	-0.068305
15차		9.208558	9.176999	9.207305	0.008442	-0.024999	-0.068305

한 실험 결과 중에서 3개의 공정에 대하여 학습 횟수가 증가함에 따라 계측 ACI 값 그리고 실측 ACI 값과의 차이가 변화하는 과정을 보여준다. 최종적으로 계측된 오류율은 약 0.08 에서 0.68% 로 평가되었다. 이 정도의 오류율은 반도체생산산업 현장에서 충분히 적용할 수 있는 수치인 것으로 알려지고 있다.

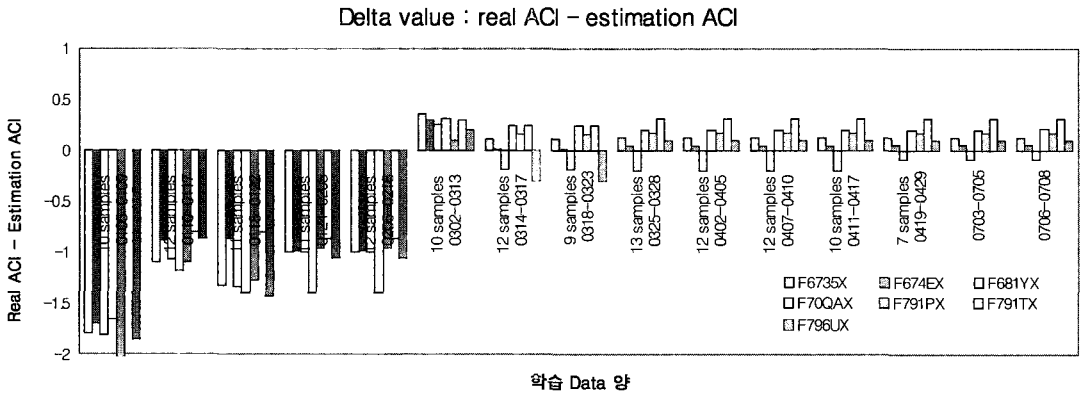
그림 2는 표 1에서 차이값이 변화하는 형태를 그림으로 나타낸 것이다. 그림 2의 결과에 따르면 초기에는 학습량이 부족하여 유사패턴을 찾는데 실패한 경우가 나타나는데, 학습량이 어느 정도 증가하면 그림과 같은 큰 오차를 나타내지는 않음을 알 수 있다.

그림 3은 중요인자만을 사용하여 평가한 경우를 보여주는데, 그림 2와 마찬가지로 학습을 진행함에 따라 ACI 차이값이 점차 감소함을 알 수 있다. 따라서 몇 개의 중요한 인자들이 ACI 값에 큰 영향을 미치는 것으로 판단된다.

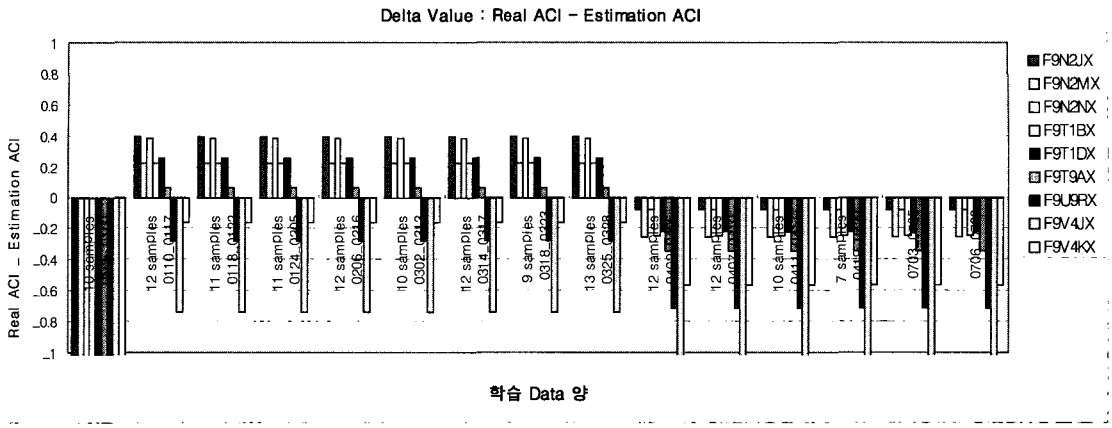
그림 4는 중요도의 순위 정보의 변경을 통해 평가한 경우를 보여주는데, 학습량이 증가함에도 오차값이 증가하는 경우가 발생함을 알 수 있다. 이것은 초기화 과정에서 권고되는 중요도 정보를 사용자가 임의로 변경하여 실험한 경우이다. 따라서 입력되는 중요도 순위가 적절하지 않을 경우



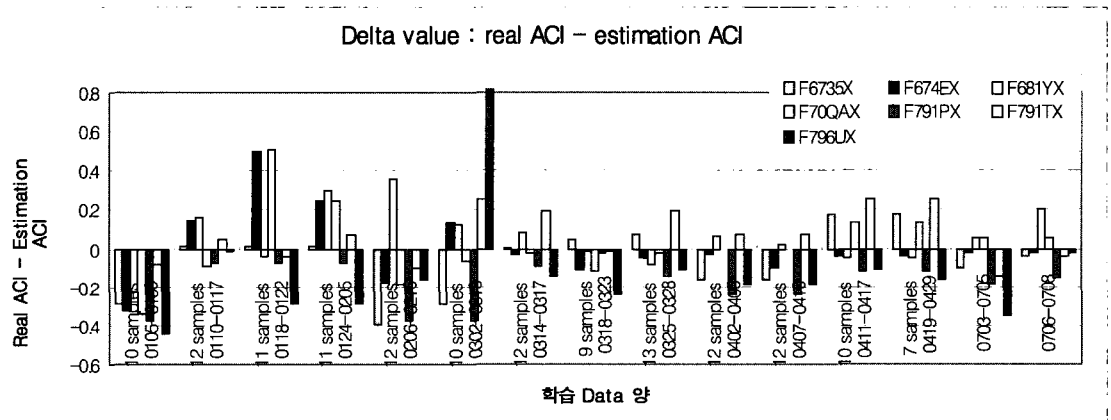
〈그림 2〉 실측 ACI 값과 계측 ACI값의 차



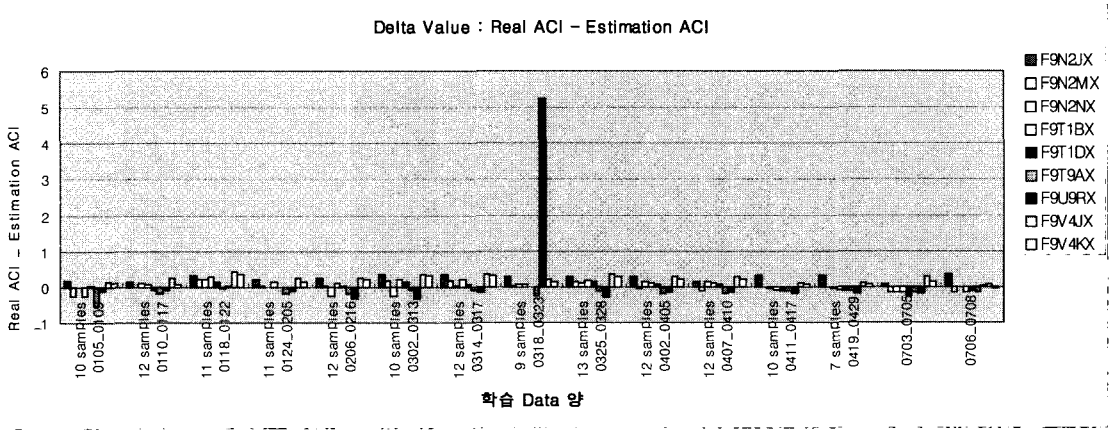
〈그림 3〉 중요인자만을 사용한 평가



〈그림 4〉 중요도 정보의 변경을 통한 평가



〈그림 5〉 유효단계의 자료만을 사용한 평가



〈그림 6〉 학습에 대한 사용자의 선택적 입력이 요구되는 경우

에는 성능에 나쁜 영향을 미침을 알 수 있다.

그림 5는 유효단계의 자료만을 사용한 평가를 보여주는데, 학습량의 증가에 따라 어느 정도는 정확성이 증가함을 알 수 있다. 하지만 현재 실험은 모든 인자의 유효단계가 단계 3과 4인 점을 고려하면 인자별로 좀 더 정확한 유효단계 정보가 요구되는 것으로 판단된다.

마지막으로 그림 6은 사용자의 선택적 입력이 요구되는 경우를 보여주는데, 그림의 가운데 부분에 큰 오차가 발생한 경우가 있다. 이것은 계측을 위하여 사용한 정보가 올바른 것임에도 훈련 자료에 유사한 패턴이 존재하지 않기 때문에 발생하는 것으로 판단된다. 결국 이러한 계측 자료에 대해서는 공정 운용자가 선택적으로 학습을 시키는 과정이 필요함을 알 수 있다.

6. 결론 및 향후연구

본 논문에서는 다중 시계열 자료를 분석하여 각각의 인자들의 상태자료를 조합하여 모든 인자들에 의하여 만들어지는 최종 측정값을 계측하는 기술을 제안했다. 즉, 본 논문에서는 사전에 수집된 자료들 중에서 입력된 패턴과 가장 유사한 패턴을 찾은 다음 표본패턴의 어떤 인자가 가지는 시계열 자료와 입력패턴에서 대응하는 인자의 시계열 자료와의 차를 분석하여 표본패턴이 가지는 계측값에서 이 오차도 만큼의 오류범위를 산출하는 방법을 제안했다.

제안된 방법은 반도체 제조공정 중에서 식각장치로부터 획득의 자료를 사용하여 성능평가를 수행하였다. 성능평가는 먼저 학습량의 증가에 따른 계측 ACI 값과 실측 ACI 값 사이의 차이가 변화하는 과정과 최종적인 결과에 대하여 수행하였다. 그 결과 최종적으로 계측된 오류율은 약 0.08 에서 0.68%로 평가되었다. 또한, 중요인자만을 사용한 경우, 중요도 자료를 변경하는 경우, 유효단계만을 사용한 경우에 대하여도 평가를 수행 하였다. 상기의 관점에 대한 성능평가를 통하여 학습량이 어느 정도 증가하면 오차가 점점 감소하는 점, 몇 개의 중요한 인자들이 ACI 값에 큰 영향을 미치는 점, 입력되는 중요도 순위가 적절하지 않을 경우에는 성능에 나쁜 영향을 미치는 점, 공정 운용자가 선택적으로 학습을 시키는 과정이 필요한 점 등을 알 수 있다.

향후연구로는 동일한 시계열 패턴이 여러 개의 실측값을 가지는 공정에 대한 확장이라 사료된다.

참 고 문 헌

[1] John R. Deller, Jr., John H. L. Hansen, and John G. Proakis, "Discrete-Time Processing of Speech Signals," IEEE Press, 2000.

[2] C. C. Tappert et al. "The state of art in on-line handwriting recognition," IEEE Trans. PAMI-12, No. 8, pp. 787-808, 1990.

[3] Ricard O. Duda, Peter E. Hart, and David G. Stork, "Pattern Classification," John Wiley & Sons, Inc., 2001.

[4] Bart Kosko, "Neural Networks and Fuzzy Systems," Prentice-Hall International, Inc., 1992.

[5] T. L. Vincent, P. P. Khargonekar, and F. L. Terry, Jr., "An extended Kalman filtering-based method of processing reflectometry data for fast In-Situ Rate Measurements," IEEE Transaction on Semiconductor Manufacture, Vol. 10, no. 1, pp. 137-145, February 1997.

[6] E. A. Rietman, "A neural network model of a contact plasma etch process for VLSI production," IEEE Transaction on Semiconductor Manufacture, Vol. 9. no. 1, pp. 95-100, February 1996.

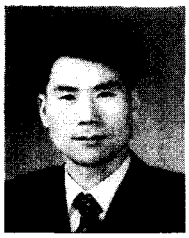
[7] P. J. O'Sullivan, J. Martinez, J. Durham,

and S. Felker, "Using UPM for real-time multivariate modeling of semiconductor manufacturing equipment," SEMATECH APC/AEC Workshop VII, New Oeleans, Louisiana, November 5-8, 1995.

[8] Linde. Y, Buzo. A, and Gray.R, "An algoritm for vector quantizer design", IEEE Trans., 1980, COM-28, No. 1, pp. 84-95.

[9] H. Sake and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. 26, No.1, pp. 43-49, 1978.

◐ 저 자 소 개 ◑



김 계 영(Gye-Young Kim)

1990년 숭실대학교 전자계산학과 졸업(학사)
1992년 숭실대학교 대학원 컴퓨터학과 졸업(석사)
1996년 숭실대학교 대학원 컴퓨터학과 졸업(박사)
1996년~1997년 한국전자통신연구원(Post Doc.)
1997년~2001년 한국전력공사 전력연구원(선임연구원)
2001년~현재 숭실대학교 컴퓨터학과 교수
관심분야 : 컴퓨터비전, 형태인식, 증강현실, 지능형 DTV, MPEG 7&21 etc.
E-mail : gykim@computing.soongsil.ac.kr