# 시계열 섭동 모델링 알고리즘 : 유전자 프로그래밍과 양자역학 섭동이론의 통합

이 금 용†


## 요 약

양자역학 섭동이론과 유전자프로그래밍(GP) 기법을 접목시킴으로써 실세계(Real-world)에서 발생하는 카오스 시계열에 대하여 수학모델을 구축, 예측하기 위한 새로운 알고리즘을 개발하였다. 시계열 분석과 양자역학 파동방정식의 해를 구하는 섭동이론과의 절차적 유사성을 논하고, 이것을 GP로 구현하는 전형적 접근방안을 제시한다. 함수집합(Function Set)으로서 직교함수(Orthogonal Functions)를 이용하고 병렬 집단을 사용하는 GP를 이용하여 원 시계열에 대한 초기 수학모델을 구하고, 원 시계열 데이터로부터 모델의 평가값을 뺀 나머지로 구성되는 잔여 시계열에 대하여 다시 GP를 적용하는 과정을 일정한 종료조건이 충족될 때까지 반복함으로써 실세계 카오스 시계열에 대한 정확성 높은 수학모델을 구축하는데 성공하였다. 타 방법론과의 비교와 향후 해결과제에 대하여도 소개한다.


# Time Series Perturbation Modeling Algorithm : Combination of Genetic Programming and Quantum Mechanical Perturbation Theory

Geum Yong Lee†


## ABSTRACT

Genetic programming (GP) has been combined with quantum mechanical perturbation theory to make a new algorithm to construct mathematical models and perform predictions for chaotic time series from real world. Procedural similarities between time series modeling and perturbation theory to solve quantum mechanical wave equations are discussed, and the exemplary GP approach for implementing them is proposed. The approach is based on multiple populations and uses orthogonal functions for GP function set. GP is applied to orginial time series to get the first mathematical model. Numerical values of the model are subtracted from the orginal time series data to form a residual time series which is again subject to GP modeling procedure. The process is repeated until predetermined terminating conditions are met. The algorithm has been successfully applied to construct highly effective mathematical models for many real world chaotic time series. Comparisons with other methodologies and topics for further study are also introduced.


**키워드 : 유전자 프로그래밍(Genetic Programming), 유전적 기호 회귀분석법(Genetic Symbolic Regression), 시계열 모델링(Time Series Modeling), 섭동이론(Perturbation Theory), 시계열 섭동 모델링 알고리즘(Time Series Perturbation Modeling Algorithm)**

## 1. 서 론

시계열이라 함은 특정 시스템으로부터 시간에 따라 발생하는 수치 데이터의 나열이며, 보통 $x_0, x_1, x_2, \cdots$와 같이 스칼라 량으로 제시된다. 다이내믹스가 알려지지 않은 시스템으로부터 발생하는 시계열을 수학적으로 모델링하고 그에 기반하여 아직 관측되지 않은 시간에 발생할 시계열 수치 데이터를 예측하는 것이 여러 학문분야에서 가지는 중요성은 아무리 강조해도 지나치지 않다.

시계열 데이터의 모델링은 스칼라 시계열에 대하여 적절한 수학적 맵핑을 적용함으로써 구성되는 벡터 시계열 $\vec{x_0}, \vec{x_1}, \vec{x_2}, \cdots$을 구성하는 것으로부터 출발한다.

하나의 벡터는 $x_{t-(n-1)\tau}, \cdots, x_{t-2\tau}, x_{t-\tau}, x_t, x_w$와 같이 구성된다[1]. 여기서 n(*embedding dimension*)은 식 (1)에서 보는 바와 같이 시계열 모델을 구축하기 위해 사용되는 스칼라 데이터의 개수이다. 아래 첨자 t는 모델링 기준 시각(또는 현재 시각)을 나타내며, $\tau$는 지연시각(*lag time*, 또는 *lag spacing*)이라고 하는데 기준 시각의 데이터에 영향을 주는 시간 간격을 나타낸다. $w = t + T$는 모델을 사용하여 예측할 미래 시각을 표시하는데, T는 기준 시각 t에 대한 상대적인 미래 시각(*lead time*, 혹은*prediction horizon*)이다. 이러한 벡터 시계열은 n차원 유클리드 상태공간(*state space*)에서 하나의 점이 되는데, 특정 시계열에 대하여 위에서 언급한 n, $\tau$, T와 같은 상태공간 패러미터를 결정하

† 정 회 원 : 영산대학교 정보통신공학부 교수
논문접수 : 2001년 9월 3일, 심사완료 : 2002년 5월 20일

는 일은 성공적인 모델링을 위해 핵심적인 관건이지만, 본 논문에서는 다루지 않기로 한다. 상태공간 패러미터를 결정하기 위해 필요한 다양한 데이터 특성화 기법에 대한 자세한 사항은 관련 문헌 [1]을 참조하기 바라다.

시계열 모델링을 수학적으로 기술하면 식 (1)과 같다. 즉, 함수 $f$에 대한 근사함수 하는 $\tilde{f}$를 구하는 것이다. 함수 $f$는 원래의 시계열을 발생시키는 시스템의 거동에 대한 수학 모델이며, 값 $x_w^{(i)}$와 나머지 벡터 컴포넌트들과의 맵핑규칙을 제공한다. 식 (1)에서 $e^{(i)}$는 $\tilde{f}$의 $f$에 대한 근사오차이다.

$$x_w^{(i)} = f(\vec{x}_t^{(i)}) \cong \tilde{f}(\vec{x}_t^{(i)}) + e^{(i)}, i = 1, 2, 3, \ldots$$
$$\vec{x}_t^{(i)} \equiv (x_{t-(n-1)r}, \ldots, x_{t-2r}, x_{t-r}, x_t)^{(i)} \in R^n \qquad (1)$$

ARMA[2] 및 GMDH[3]와 같은 다른 시계열 모델링 기법과 마찬가지로, 소프트컴퓨팅 기법의 하나인 유전자프로그래밍(genetic programming, GP) 기반의 시계열 모델링 기법 역시 수학적 기호에 의해 모델을 표현한다. GP기반의 시계열 모델링 기법은 비교적 시스템 거동이 안정된 정적 시계열(*stationary time series*)에 적용되어 매우 의미있는 모델링 효율 및 예측성능을 보여준 바 있다[4-6].

그러나, 실세계(real world)에서 우리의 관심을 끄는 중요 시계열의 대부분은 그 거동이 매우 불안정(*nonstationary*)하며[2] 카오스적이어서 유전자 프로그래밍 기반의 실세계 시계열 모델링은 인공 신경회로망 등 타 기법에 비교하여 상대적으로 성공적인 것으로 주목받지 못한 것이 사실이다. 더욱이, 시계열 데이터에 포함된 잡음(noise) 등에 대한 적절한 대처 알고리즘이 개발되지 않았기 때문에 제한된 성능의 모델을 구축하는 데에도 과도한 계산자원을 소모해야 했다.

본 논문에서는 시계열 모델링과 양자역학 파동방정식(quantum mechanical wave equation)의 해를 구하는 섭동이론(perturbation theory)[7] 과의 절차적 유사성을 유전자 프로그래밍으로 접목함으로써, 실세계 시계열 모델링에 대한 유전자 프로그래밍의 유용성을 입증하고자 한다.

실세계의 시계열 모델링과 섭동이론의 공통적인 최종 목적은 불안정하거나 확률론적(nonstationary or stochastic) 성질을 가지는 시스템 거동에 대한 수학적 정식화이다. 단지, 이들 2가지 기법으로 해결해야 할 문제로서 제시되는 시스템 거동의 표현 형식이 다를 뿐이다. 섭동이론은 변수값에 많은 불확실성이 내포되어 있긴 하지만 수학적으로 잘 정의된 파동방정식을 만족시키는 해를 구해야 하며[7, 8], 시계열 모델링은 수치 데이터의 나열로서 암묵적으로(implicitly) 주어지는 시스템 거동을 가장 효율적으로 표현할 수 있는 수학적 형태를 구하는 것이다.

시스템 거동의 제시 형식이 가지는 위와 같은 차이점에도 불구하고, 해를 구하는 절차에 있어서는 매우 유사하다는 점에 착안하여, 본 논문에서는 수치데이터로서 표현되는 미지의 시스템 거동을 수학적 기호로 정식화 할 수 있는 GSR(genetic sysmbolic regression ; GP기반의 기호 회귀분석법[4])을 재해석함으로써 시계열 모델링에 섭동이론을 적용할 수 있었다.

제 2절에서는 본 논문에서 제시되는 새로운 알고리즘을 정리하였으며, 제 3절에서는 다수의 실세계 시계열에 대한 적용 사례를 소개한다. 제 4절에서는 본 논문의 결론과 향후의 해결과제를 제시하고 있다.

## 2. 시계열 섭동 모델링 알고리즘

### 2.1 섭동이론(Perturbation Theory)

기본적으로 불안정한 다이내믹스를 가지고 있는 파동방정식은 소위 정해(正解, exact solution)를 가지고 있지 않는 것이 보통이다. 섭동이론[8]은 비섭동(unperturbed) 및 섭동(perturbed) 해밀토니언(Hamiltonian) 이라는 2개의 함수로서 양자역학 파동 방정식의 해를 근사하여 구하는 방법을 제시한다.

이제, $\Psi_u(\vec{x})$를 비섭동 해밀토니언이라고 하고, $\Psi_p(\vec{x})$을 섭동 해밀토니언이라고 하면

$$\Phi(\vec{x}) = \Psi_u(\vec{x}) + \Psi_p(\vec{x}) \qquad (2)$$

그 선형결합, 즉 식 (2)가 양자역학 시스템 거동을 기술하는 파동방정식을 만족하는 해가 되는 것이다. 달리 말한다면, 불안정한 시스템 거동은 안정된 거동함수(비섭동 해밀토니언)와 불안정한 거동함수(섭동 해밀토니언)의 선형결합으로 나타낼 수 있는 것이다. 이때 불안정한 거동함수, $\Psi_p(\vec{x})$, 즉, 섭동 해밀토니언 역시 또 다른 2개 거동함수(섭동, 비섭동 쌍)의 선형결합으로 표현할 수 있다. 이와 같은 과정은 일정한 기준을 만족하여 해밀토니언 쌍을 구하는 것이 더 이상 의미가 없어질 때까지, 연속하는 섭동 해밀토니언에 대하여 자기 반복적으로(recursively) 계속될 수 있다.

섭동이론은 연속하는 해밀토니언 쌍을 구하는 매우 효율적인 방법을 가지고 있다. 최종적으로 얻어지는 모든 비섭동 해밀토니언을 선형결합하면 양자역학 파동 방정식의 해가 된다.

### 2.2 시계열 모델링과 섭동이론의 유사성

시계열 모델링에 있어서는 수치로 표현되는 시계열 자체가 섭동이론에서의 파동방정식과 동일한 역할을 한다고 볼 수 있다. 섭동이론이 파동방정식의 비섭동 해밀토니언 계열을 연속적으로 구했던 것과 마찬가지로, 수치의 형태로 주어지는 암묵적 시스템 방정식 혹은 거동으로부터 연속적인 비섭동 시계열 해밀토니언을 구할 수만 있다면, 섭동이론과 시계열 모델링은 근본적으로 유사한 문제해결 방식을 가지

고 있다고 보아도 무방할 것이다.

이를 위하여, 본 논문에서는 식 (1)과 같은 시계열 모델링 정식화에서 원 시스템의 거동을 표현하는 함수 $f$를, 식 (2)에서와 같이 식 (3)으로 표현한다. 즉,

$$f(\vec{x}) = f_u(\vec{x}) + f_p(\vec{x}) \tag{3}$$

여기서, $f_u(\vec{x})$는 비섭동 시계열 모델이며, $f_p(\vec{x})$는 섭동 시계열 모델이다.

### 2.3 비섭동 시계열 해밀토니언과 유전자 프로그래밍

양자역학 파동방정식과는 달리 시계열 모델링에 있어서의 시스템 거동은 수학 방정식으로 주어지지 않기 때문에 식 (3)의 시계열 모델을 섭동이론을 이용하여 직접 구하는 것은 불가능하다. 그러나, 앞서 서론에서 언급하였던 유전자 프로그래밍의 기법, GSR을 사용하면 주어진 시계열에서 안정적으로 모델링되는 부분, 즉 비섭동되는 시스템 거동 모델 $f_u(\vec{x})$을 구할 수 있다.

첫번째 비섭동 시계열 모델은 GSR을 원래의 시계열 데이터에 우선 적용함으로써 구해지는 수학모델을 사용하면 된다. 이것은 Koza 교수의 통상적인 방법[4]으로 가능한 것이다. 그렇다면 남은 문제는 어떻게 하여 연속적인 비섭동 시계열 모델을 구하는가 하는 것이다. 이것을 위하여 본 논문에서 제안하는 방법은 다음과 같다.

두번째 비섭동 시계열 모델은 식 (4)로 구해지는 잔여 시계열(*residual time series*), $f_p(\vec{x})$에 대하여 GSR을 다시 적용하는 것이다.

$$f(\vec{x}) - f_u(\vec{x}) = f_p(\vec{x}) \tag{4}$$

세번째 이후의 비섭동 시계열 모델은 위의 과정을 연속하는 잔여 시계열에 GSR를 계속하여 적용하면 될 것이다.

일반적으로 표현하면, j 번째 모델링에서 얻어지는 비섭동 시계열 모델을 $f_u(\vec{x})^j$이라고 표시할 때, 유전자 프로그래밍을 이용한 시계열 섭동이론의 결과식은 식 (5)와 같다.

$$f(\vec{x}) = \sum_{j=0}^{J} a^j f_u(\vec{x})^j + f_p(\vec{x})^J \tag{5}$$

식 (5)에서 $f_u(\vec{x})^0 = 1$이며, 상수 $a^j, j = 0, 1, 2, \cdots, J$는 함수 $f$로 나타내어지는 원 시계열에 대하여 최소자승법으로 구하여 $f_p(\vec{x})^J$이 모든 시계열 데이터 관측 시각에서 무시할 수 있을 만큼 작은 값이 되도록 하는 것이 좋다.

### 2.4 복수 진화집단을 사용하는 유전자 프로그래밍에 있어서의 시계열 섭동 모델의 결정

Koza교수가 제안한 바와 같이[4], 유전자 프로그래밍을 이용하여 시계열의 모델을 구축하기 위해서는 우선 수학식을 나타내는 기호구조체 (symbolic forms, 진화집단의 개체로 사용됨)를 정해진 개수만큼 생성하여 진화집단을 형성한 후, 가능한 한 작은 오차로서 원 시계열의 데이터를 재현, 재생할 수 있도록 기호구조체 내부 각 위치에서의 기호 종류와 각 기호들의 구조적 형태(짜임새)를 진화적으로 변경해가야 한다.

정해진 기준에 의해 진화과정을 종료한 후 주어진 기호구조체로 이루어진 진화집단에서 가장 효율이 좋은, 다시 말하면 주어진 시계열 데이터를 가장 작은 오차로 근사하는 기호구조체의 수학적 형태를 주어진 시계열에 대한 모델로 결정하게 된다.

그런데, 만약 진화에서 복수(예를 들어, P개)의 집단을 사용한다면 집단의 개수 P만큼 시계열 모델의 후보가 만들어질 것이다. 이럴 경우, 어느 후보를 최종적인 시계열 모델로서 채택할 것인가가 문제이다. 가장 단순한 방법은 이들 후보중에서 가장 오차가 작은 것을 선택하면 될 것이다.

그러나, 계산자원을 소모하여 결정되고, 비록 전체적인 오차에는 차이가 있지만 시계열 데이터가 나타내는 시스템 거동을 부분적으로나마 표현하고 있는 나머지 P-1개의 모델 후보들은 무시해야 하는 것일까.

가장 좋은 모델 후보라 할지라도 오차를 가지고 있다는 것은 완벽하게 시스템 거동을 표현하고 있지 못하다는 점을 반증하는 것이기 때문에 본 논문에서는 경제적인 시계열 모델을 구축하기 위해 복수 진화집단 각각에서 결정되는 개별 시계열 모델후보들을 적절히 조합하는 알고리즘을 제안한다.

이제 $g_{pp}(\vec{x})$를 진화집단 $pp$에서 결정되는 모델후보(진화과정의 종료 이후 주어진 진화집단에서 가장 성능이 좋은 기호구조체)이라고 하자. 이 경우 $j$번째 비섭동 시계열 모델 $f_u(\vec{x})^j$은 식 (6)과 같은 형태를 가진다.

$$f_u(\vec{x})^j = \alpha_0 + \sum_{pp=1}^{P} \alpha_{pp} g_{pp}(\vec{x}) \tag{6}$$

식 (6)을 식 (5)에 대입하면,

$$\begin{aligned} f(\vec{x}) &= \sum_{j=0}^{J} a^j f_u(\vec{x})^j + f_p(\vec{x})^J \\ &= \sum_{j=0}^{J} a^j \left[ \alpha_0 + \sum_{pp=1}^{P} \alpha_{pp} g_{pp}(\vec{x}) \right]^j + f_p(\vec{x})^J \end{aligned}$$

이 되고, 식 (7)과 같이 정리된다.

$$f(\vec{x}) = \sum_{j=0}^{J} a^j \alpha_0^j + \sum_{j=1}^{J} \sum_{pp=1}^{P} a^j \alpha_{pp}^j g_{pp}(\vec{x})^j + f_p(\vec{x})^J \tag{7}$$

표기법을 변경하여 식 (7)을 다시 쓰면 다음과 같다.

$$f(\vec{x}) = \beta_0 + \sum_{K=1}^{J \times P} \beta_K g_K(\vec{x}) + f_p(\vec{x})^J \tag{8}$$

결국, 복수개의 진화집단을 사용하는 유전자 프로그래밍에 있어서의 시계열 섭동 모델은 모든 집단에서 결정, 축적된 모델 후보 전체를 최소자승계수로 선형결합한 형태가 되는 것이다. 최종적인 비섭동 시계열 모델은 다음과 같이 주어진다.

$$f_w(\vec{x})^j = \beta_0 + \sum_{K=1}^{J\times P} \beta_K g_K(\vec{x}) \qquad (9)$$

### 2.5 시계열 섭동 모델링 알고리즘 요약

섭동이론과 시계열 모델링을 유전자 프로그래밍을 이용하여 통합하는 시계열 섭동 모델링 알고리즘을 요약하면 다음과 같다.

**Step 1** : 원 시계열 데이터를 모델 구축을 위한 훈련영역(Training Data), 검증영역(Validation Data), 예측(성능시험) 영역(Forecast Data)으로 나누어 백터 시계열을 제작한다.

**Step 2** : 유전자 프로그래밍의 계산규모를 정의하는 여러가지 패러미터를 결정한다. 집단의 수 P, 재생산을 위한 개체간 교배비율(Crossover Rate), 돌연변이율(Mutation Rate), 집단간 개체 이주비율(Migration Rate), 세대 한도(Generation Limit) 등이다.

**Step 3** : 유전자 프로그래밍의 내용을 결정하는 필수요소를 결정한다. 함수집합(Function Set), 터미널 집합(Terminal set), 적응도(Fitness) 계산공식, 종료조건을 설정하기 위한 모델링 오차한도 등이 포함된다.

**Step 4** : 모델링 과정의 수를 계수하는 변수 $j$의 값을 0으로 한다.

**Step 5** : P개의 복수 집단(Multiple Populations)으로 구성되는 유전자 프로그래밍 진화집단 초기화를 실시[4]한다. 이 과정에서는 임의의 구조와 내용을 가지는 기호구조체가 함수집합 및 터미널집합으로부터 생성된다.

**Step 6** : 훈련영역 데이터에 대하여 유전자 프로그래밍을 이용하는 시계열 모델링 과정, 즉 GSR(genetic symbolic regression) [4]을 적용한다.

**Step 7** : 진화의 세대한도에 도달하거나, 진화중의 모델 평가값이 주어진 오차한도 이하일 경우에 GSR을 중단하고, P개의 집단에서 가장 우수한 개체(기호구조체)를 하나씩 선택하여 모델 후보로 결정한다. 이것이 식 (6)의 $g_{pp}(\vec{x})$, $pp = 1 \sim P$, 이다. 훈련영역 데이터에 대하여 최소자승법(Least Square Method)를 적용하여 식 (6)의 수치 계수들을 구하고, 비섭동 시계열 모델로 지정한다. 최초의 모델링 과정이 아니라면, 즉, $j \neq 0$이라면 이전의 모델링 과정에서 결정, 축적된 모든 모델 후보 함수들을 포함하여 결정되는 비섭동 시계열 모델 식 (9)을 구성한다. 그러나, 모델 후보의 수가 과도하게 증가하는 것을 방지하기 위하여 모델 후보의 사용개수를 제한하는 방법이 필요하며, 이를 위하여 제 2.6절에서 슈퍼집단(super population)이라는 개념을 도입한다.

이렇게 하여 구축된 시계열 모델의 예측 성능은 몇가지 현실적인 이유로 인해 기대에 못미칠 수도 있다. 훈련영역 시계열 데이터가 주어진 시스템 거동을 대표하기에는 불충분하였거나, 충분한 데이터가 제공되었다 하여도 모델 구축을 위해 설정된 계산자원(계산 시간, 용량, 규모 등)이 부적절하였을 수도 있다. 데이터에 과다한 잡음이 섞여 있을 수도 있기 때문에, 다양한 실세계 시계열에 대하여 최적의 훈련영역 혹은 계산자원의 적절성을 결정하는 것은 대단히 어려운 일이다.

예측영역(Step 1 참조)에서의 모델 예측 성능을 높이는데는 여러가지 방법이 있다. 가장 간단한 방법은 예측 희망위치 ($fp$) 이전까지 입수된 모든 데이터를 포함하도록 훈련영역을 확장하여 새로운 모델을 구축하는 것이다. 이 방법은 추가적인 계산자원에 대한 비용을 감당할 수 있고, 모델 구축의 적시성 확보가 반드시 전제되어야 한다는 점에서 채택하기가 용이하지는 않다.

본 논문에서는 예측위치 ($fp$) 이전까지 알려진 모든 데이터를 포함하여 식 (6) 혹은 식 (9)에서 사용되는 최소자승계수를 다시 계산하는 현실적 타협을 택함으로써 위와 같은 어려움을 피하고, 모델의 예측성능을 가능한 한 먼 곳까지 유지시키는데 성공하였다.

위치 $fp$ 이전 위치 $v$ ($S = fp - v \geq 1$)까지의 시계열 데이터의 실제 값이 알려져 있다면, $fp$에서의 데이터를 예측하기 전에 $1 \sim v$까지의 데이터에 대하여 식 (9)의 최소자승계수를 계속하여 갱신하는 것이다. $S$를 본 논문에서는 최신 데이터를 반영하여 재계산된 최소자승계수를 사용할 수 있는 구간 간격이라는 의미로서 ***impact step***이라고 칭한다.

검증영역에서의 모델의 예측성능 (검증성능)은 모델링과정의 지속 여부에 결정적인 역할을 하게 된다. 아래의 Step 9를 참조하자.

**Step 8** : 식 (4)에서 보는 바와 같이 원래의 시계열 데이터 $f$에서 식 (9)의 평가값을 뺀, 잔여 시계열(*residual time series*)를 생산한다.

**Step 9** : 잔여 시계열의 크기(order of magnitude)가 무시할 수 있을만큼 작거나, 이전 모델링 단계의 검증성능(위의 Step 7의 설명 참조)보다 좋지 않은(낮은) 검증성능을 보여 더 이상의 모델링 작업이 의미가 없는 것으로 판단되거나, 지정된 계산자원이

모두 소진되면 모델링 과정을 종료하고, 식 (9)에 의해 결정되는 모델을 주어진 시계열의 모델로서 최종 지정한다.

이전 모델링 단계보다 좋지 않은 검증성능을 보인다는 것은, 모델링이 지나치게 진행되었다는 것(over modeling), 다시 말하여 잡음이나 교란신호와 같은 시스템 거동과 무관한 수치정보가 과도하게 모델에 반영되었다는 것을 의미하므로, 훈련영역 에서의 근사성능은 개선될지 모르나, 검증영역 이후에는 오히려 예측성능 저하의 원인이 된다. 이러한 상황이 발생할 경우 모델링 과정을 곧바로 종료시킴으로써 과도한 모델링을 방지하는 접근방법은 인공신경망이론 분야에서 *조기 종료정책(early stopping policy)*으로 알려져 있는 것[9]을 채택하였다.

**Step 10** : Step 9의 종료 조건이 만족되지 않으면 $j$를 1 증가시킨 후, 위의 Step 5으로 되돌아 가되, GSR은 Step 8에서 구한 잔여 시계열에 적용한다.

### 2.6 유전자 프로그래밍 구현을 위한 상세 사항

본 절에서는 위에서 제시된 알고리즘을 구현하는 유전자 프로그래밍에서 사용할 수 있는 몇가지 기술적 상세사항을 소개함으로써 본 논문의 시계열 섭동 모델링 알고리즘의 재현에 도움을 주고자 한다.

#### 2.6.1 각 개체(기호구조체)의 성능평가

특정 진화집단의 개체(여기서는 기호구조체)가 주어진 시계열을 어느 정도 정확하게 근사하고 있는가를 측정할 수 있는 수치기준이 필요하다. 시계열 모델링 분야에서 잘 쓰이는 것으로서는 정규평균제곱오차(NMSE, Normalized Mean Squared Error [1])와 변이계수(CV, Coefficient of Variation[6])이며 다음과 같이 정의된다.

$$CV(N) = \frac{1}{\bar{x}}\left[\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\tilde{x}^{(i)})^2\right]^{0.5} \quad (10)$$

$$NMSE(N) = \frac{\sum_{i=1}^{N}(x^{(i)}-\tilde{x}^{(i)})^2}{\sum_{i=1}^{N}(x^{(i)}-\bar{x})^2} \cong \frac{\sum_{i=1}^{N}(x^{(i)}-\tilde{x}^{(i)})^2}{N\hat{\sigma}^2} = \frac{MSE(N)}{\hat{\sigma}^2} \quad (11)$$

여기에서 $\tilde{x}^{(i)}$ 및 $x^{(i)}$는 데이터 위치 $i$에서의 모델 평가값과 실제 데이터 값을 나타낸다. $\bar{x}$과 $\hat{\sigma}^2$은 주어진 시계열의 샘플 평균과 분산을 나타내고, $N$은 $NMSE(N)$ 혹은 $CV(N)$의 계산에 사용되는 데이터의 개수이다. MSE는 평균 제곱오차(Mean Squared Error)의 약자이다. 본 논문의 경우 진화집단내 개체의 적응도(Fitness)로서 $NMSE(N)$ 혹은 $CV(N)$의 역(inverse)을 사용하였다.

#### 2.6.2 슈퍼집단 및 집단간 개체 이동

본 논문에서 시계열 모델링을 위해 사용하는 유전자 프로그래밍의 경우 슈퍼집단(super population)이라고 불리우는 특수 집단을 사용하고 있다. 이것은 앞서 제 2.5절의 Step 7에서 언급 하였다시피, 모든 모델링 과정의 각 집단에서 결정된 모델후보들, 즉, 식 (9)의 $g_k(\vec{x})$의 수를 제한하여 각각의 질을 유지하기 위하여 제안되었다. 모델링 과정 종료후 슈퍼집단에 보존된 모델후보들은 식 (9)로 주어지는 최종 모델의 기본함수(Basis Functions) $g_k(\vec{x})$가 된다.

슈퍼집단의 개체(즉, 슈퍼개체)를 선발하는 방법은 토너먼트 방식으로 이루어진다. 모델링과정이 종료되어 각 집단별 1개씩 P(진화집단의 개수) 개의 모델후보 개체들이 결정된 후, 기존 슈퍼집단에 모아둔 슈퍼개체보다 더 좋은 적응도 값을 가지는 경우에만 기존 슈퍼개체를 대체할 수 있는 권한을 부여한다. 슈퍼집단의 크기(즉, 포함되는 개체의 개수)를 한정시킬 경우, 슈퍼집단에는 모든 모델링 과정에 걸쳐 가장 우수한 성능을 가지는 모델 후보들, 그것도 정해진 개수만이 축적, 보존되는 것이다. 식 (9)과 같이 주어지는 비섭동 시계열 모델에 사용되는 모델후보의 개수가 과도하게 증가하는 것을 방지할 수 있음을 알 수 있다.

진화과정중에는 집단간 개체의 이동(Migration)이 허용되는데, 슈퍼집단과 일반 진화집단간에는 진화과정중 어떠한 교류도 허용되지 않는다. 모델링 과정 종료 후 식 (9)를 이용, 비섭동 시계열 모델을 구성하기 위한 스토리지 역할만을 수행한다. 이런 의미에서, 본 논문에서 도입하는 슈퍼집단은 다른 연구[10]에서의 멀티에이전트 팀과는 근본적으로 다르다. 멀티 에이전트 혹은 팀 멤버들은 진화과정에 직접 관여하고 있다.

#### 2.6.3 3가지 비정상적 기호구조체(Three Symbolic Anomalities)에 대한 처리

유전자 프로그래밍을 이용하여 시계열 모델링 알고리즘을 구현하다 보면 수학적으로나 계산상으로 처리하기가 곤란한 비정상적 기호구조체가 다수 발생한다. 본 논문에서는 이러한 상황을 크게 3가지로 구분하고 이에 대한 적절한 처리방안을 제안하고자 한다.

첫번째 비정상 기호구조체는 수학식으로 전환할 경우 의미가 없어지는 혹은 수학적으로 정의되지 않는 종류이다. 본 논문에서는 이를 수학적 비정상체(*mathematical anomality*)라고 칭하기로 한다. 예를 들어, 영(zero)로 나누는 경우, 제곱근을 계산하는 인수(arguments)로서 음의 값이 주어지는 경우 등이 수학적 비정상체의 예라 하겠다.

수학적으로는 문제가 없으나, 계산기 에러를 유발시키는 기호구조체를 계산기적 비정상체(*computational anomality*)라고 칭한다. 오버플로우(overflow) 혹은 언더플로우(underflow)를 발생시키는 기호 구조체가 좋은 예이다.

마지막으로, 의미론적 중복 비정상체(*semantic replication*)이다. 기호구조체 내부 특정 위치의 기호의 종류와 구조형태가 다를지라도 수학적으로 동일한 식을 의미할 경우 발생한다. 예를 들어, $(+x3\,(sin(/x2\ x2)))$은 $(+x3(sin\ 1))$ $=x3$의 의미론적 중복체인 것이다. 슈퍼집단의 슈퍼개체, 그리고 각 집단으로부터 선발된 모델 후보들에 이러한 의미론적 중복체가 존재하게 될 경우 식 (9)과 같이 최소자승법을 이용하여 계수를 구할 때 Singular Matrix가 발생한다.

모든 종류의 비정상 기호구조체는 계산자원을 소모하고, 진화집단에 있어서의 유전적 다양성을 해치기 때문에 유전자 프로그래밍의 구현에 있어서 각별한 주의가 필요하다.

수학적 비정상체 혹은 계산기적 비정상체는 해당 기호구조체의 평가함수를 재정의하여 비정상적 상황을 미리 대비하면 된다. 비정상체로 판정되는 상황이 되면 미리 지정된 특정 값을 평가값으로 대체한다. 물론, 이러한 기호구조체는 아주 낮은 적응도 값을 부여하여 진화 과정에서 자연스럽게 도태시켜야 한다. 제곱근을 구하는 함수 sqrt를 재정의하여 수학적 비정상체를 방지하는 LISP 코드 샘플을 다음과 같이 예시한다.

```
(defun rsqrt (X)
      (cond
        ((<= X 0.0d0)
                  (incf *punish-count*)
                  (sqrt most-positive-double-float))
        ((<= X least-positive-double-float)
                  (incf *punish-count*)
                  (sqrt least-positive-double-float))
        ((>= X most-positive-double-float)
                  (incf *punish-count*)
                  (sqrt most-positive-double-float))
      (t
        (coerce (sqrt X) 'double-float))
  ))
```

인수 x가 0보다 작으면, *punish-count*의 수를 1 증가 시킨 후 sqrt의 값으로서 (sqrt most-positive-double-float)을 제시한다. 또한, x가 언더플로우나 오버플로우가 발생해도 유사한 대응값을 제시하고 있다. 여기서, *punish-count*는 하나의 기호구조체 내부에서 발생하는 비정상적 상황 발생수를 보관하는 전역변수이며, most- 혹은 least-positive-double-float는 주어진 계산기에서 지원가능한 가장 큰 혹은 가장 작은 배정도(double precision) 양수 값을 나타내는 LISP 시스템 상수이다.

의미론적 중복 기호구조체를 선별하고 대처하는 것이 가장 복잡한데 본 논문에서는 의미론적으로 중복되는 모든 트리구조 유형을 원래의 기호구조체에서 제거하고 추후 가장 단순한 동일의미의 기호만을 삽입하여 기호구조체의 의미가 유지되도록 하였다. 그리고 만약 있을지도 모를 제 3의 의미론적 중복 구조를 제거하기 위해 remove-duplicates라는 LISP 함수를 #'tree-equal 기준으로 검사하여 제거하였다. 다음은 이에 관한 샘플 LISP 코드이다.

여기서 사용한 LISP 언어 함수 set-difference는 2 집합의 차이를 구하며, remove-duplicates는 동일 요소를 주어진 집합에서 제거한다. tree-equal은 두개의 리스트의 구조가 동일한지 평가한다. 자세한 것은 관련 문헌[11]을 참조하자.

```
(defmethod remove-semantic-duplicates
    ((variable-no# 1#) (bases list))
      (let* ((basis-length (length bases))
             (temp-basis (set-difference bases
                                         '(x1 (rlog (rexp x1))
                                         (rexp (rlog x1))
                                         (rsqrt (m x1 x1)))
                                          :test #'tree-equal))
             (length-change
                 (- basis-length (length temp-basis))))

    (remove-duplicates
      (if (not (zerop length-change))
        (if (member 'x1 temp-basis :test #'tree-equal)
            temp-basis (push 'x1 temp-basis))
        temp-basis)
     :test #'tree-equal)))
```

2.6.4 유도 터미널 세트(DTS, Derived Terminal Set)

유전자 프로그래밍을 이용한 시계열분석의 경우 함수집합(Function Set)과 터미널 집합(Terminal Set)의 각 요소를 이용하여 시계열 모델을 수학적으로 표현하기 때문에 이들 집합을 효율적으로 정의해야 한다.

터미널 집합은 함수 기호를 위한 인수용 기호를 제공하게 된다. 예를 들어, $(sin\ x_3)$라고 하는 아주 간단한 기호구조체에 있어서도 함수 기호 *sin*를 위한 인수기호 $x_3$를 필요로 하고 있다.

본 논문에서 정의하는 유도 터미널 집합(DTS, *derived terminal set*)은 $(sin\ x_3)$, $(cos\ x_3)$과 같은 원시적인 함수를 나타내는 기호구조체를 초기화하거나 진화적 과정에서 형성시키는데 소모되는 계산자원의 낭비를 예방하는 목적을 가지고 있다.

기본적으로 DTS는 원시적인 함수를 나타내는 기호와 인수기호를 하나의 기호로 표기, 통상적인 터미널 집합에 개별 요소와 같이 당초에 삽입되어 진화 초기집단의 개체 생성시 하나의 터미널로서 활용된다. 그러나, DTS를 정의할 때 사용되는 원시함수는 가급적 일반 함수의 테일러 전개(Taylor expansion)에 사용될 수 있는 직교함수(Orthogonal Function)[12]로 하는 것이 좋다. 이는, 시계열 모델링 역시 원래의 시스템 다이내믹스를 표현하는 미지함수, 식 (1)의 함수 $f$를 전개하는 과정으로 이해할 수 있기 때문이다. 잘 알려진 직교함수에는 삼각함수, 체비쉐프 함수 등이 있다. 체비쉐프 DTS, 혹은 체비쉐프 터미널, $T_{order,i}$을 만드는 과정을 예시하면 다음과 같다.

$$T_{order,i} \equiv cos[order \times \arccos(x_i^{**})],\ x_i^{**} \in \vec{x} \quad (12)$$

여기서 *order*는 정수이며, $x_i^{**}$의 이중 별표는 시계열 변수 $x_i$가 체비쉐프 함수의 정의 범위 [-1, 1]로 맵핑되어야 함을 표시한다. 본 논문에서는 다음과 같은 단조 1차 함수로서 맵핑을 수행한다.

$$x_i^{**} = sx_i - t,\ s = 2(x_i^{MAX} - x_i^{MIN})^{-1},$$
$$t = 1 + 2x_i^{MAX}(x_i^{MAX} - x_i^{MIN})^{-1} \quad (13)$$

여기서, $x_i^{MAX}$및 $x_i^{MIN}$는 시계열 변수 $x_i$의 전역 최대값 및 최소값을 나타낸다. 훈련영역 시계열 데이터에서 계산할 수 있는 최대값 ($x_i^{\max}$) 및 최소값 ($x_i^{\min}$)이 전역값이라는 보장을 할 수 없으므로 전역 최대값과 최소값을 계산 혹은 추정하는 것은 대단히 어려울 수 있다. 본 논문에서는 식 (14)에서와 같이 임의의 양수 확장비(positive expansion ratio) $\eta$를 도입하여 전역 최대값과 최소값을 추정하고 있다. 따라서, 훈련영역 시계열 데이터에서 발견되는 최대값과 최소값의 간격을 $2\eta+1$배 확장하는 효과를 가지고 있다. 검증 및 예측영역 시계열 데이터에 이 값보다 큰 값이 존재할 경우 체비쉐프 함수가 정의되지 않으므로 모델링 과정을 다시 실시해야 한다.

$$x_i^{MAX} = x_i^{\max} + \eta \cdot \Delta,\ x_i^{MIN} - \eta \cdot \Delta,$$
$$\Delta = x_i^{\max} - x_i^{\min} \quad (14)$$

## 3. 실세계 시계열에 대한 적용

실세계에서 발생하는 시계열 데이터에 대하여, 제 2. 5절의 시계열 섭동 모델링 알고리즘을 적용한 결과를 정리한다. 본 절에서 사용된 유전 프로그래밍 계산규모와 주요 패러미터는 아래와 같으며, 모든 시계열에 동일하다.

- 진화 집단의 수 = 5, 집단의 크기(집단내의 개체의 개수) = 30, 세대 한계 = 9, 최대의 섭동 모델 후보 개수(슈퍼집단의 크기) = 5
- 함수집합 = {+, −, ×, /, sin, cos, exp, log, expt }, 터미널 집합 = $\{x_{t-(n-1)\tau}, \cdots, x_{t-2\tau}, x_{t-\tau}, x_t\} \cap \{T_{order,i} | order = 1 \sim 10$, 각 $x_i$에 대하여 정의$\}$, 초기집단의 기호구조체의 트리 깊이(depth) = 6, 교배(crossover) 후의 최대 깊이 = 18
- 교배(Crossover) 비율 = 0.8, 돌연변이 비율 = 0.1, 개체 재활용(Reproduction) 비율 = 0.1, 집단간 이동 개체비율 = 전 집단 개체 총수의 1%.
- Lag spacing = 1, Lead time = 1, Embedding dimension = 1 or 4, Impact step = 1
- 벡터 시계열의 총 개수 = 400, 훈련영역 벡터 수 $\boldsymbol{T}$ = 최초 200개, 검증영역 벡터 수 $\boldsymbol{V} = \boldsymbol{T}$ 이후 100개, 예측영역 벡터 수 $\boldsymbol{F} = \boldsymbol{V}$ 이후 나머지 100개, 모델링 과정 종료를 위한 타겟 성능(Terminating NMSE) = 0.01.

### 3.1 인간 신체 혈류에 대한 시계열

인간 신체의 혈류를 시뮬레이션하는 Mackey-Glass 방정식, 식 (15)를 풀어 생성되는 시계열은 많은 연구자들에 의해 시계열 모델링 기법의 표준 데이터로서 활용된 바 있다 [5, 6, 13].

$$\frac{dx_t}{dt} = \frac{bx_{t-\Delta}}{1 + x_{t-\Delta}^c} - ax_t,$$
$$x_{t+1} = (1-a)x_t + bx_{t-\Delta}(1 + x_{t-\Delta}^c)^{-1} \quad (15)$$

상수 $a$, $b$ 및 $\Delta$를 적절히 설정하여 식 (15)을 사용하면 매우 강한 카오스 시계열을 발생시킬 수 있다. 시계열의 최초 40개의 데이터는 임의로 주어진다. 위에서 열거한 계산규모와 패러미터를 적용하여 본 논문의 알고리즘을 구현한 결과와 다른 유전자프로그래밍 기반의 연구결과를 <표 1>에 비교하였다.

〈표1〉 인간신체혈류 시계열 분석결과 비교

| 예측성능 | 기존 연구결과 | | 본 논문의 알고리즘 |
|---|---|---|---|
| | [6] | [13] | |
| NMSE(20) | 0.031 | 0.063 | 0.019 |
| NMSE(30) | | 0.159 | 0.009 |
| NMSE(40) | 0.158 | 0.316 | 0.004 |
| NMSE(50) | 0.371 | 0.631 | 0.003 |
| NMSE(60) | 0.617 | 0.990 | 0.005 |

위의 결과는 계산규모로 보나, 예측 성능으로 보나 유전자 프로그래밍에 의한 다른 시계열 모델링 기법보다 본 논문의 방법이 훨씬 효과적임을 보여주고 있다. 이에 대한 보다 자세한 논의는 참고문헌[14]에서 이루어진 바 있다.

### 3.2 Santa Fe 및 ASHRAE 시계열 예측 경시대회

Santa Fe Institute 및 ASHRAE(American Society for Heating, Refrigerating, and Air-conditioning Engineers)는 시계열 분석과 예측에 관한 국제적인 경시대회를 개최한 바 있다[1]. 사용된 시계열은 데이터의 특성상 매우 안정적인 기계적인 시계열에서부터 매우 불안정하고 휘발성 강한 경제 시계열에 이른다. 경시대회에 제시된 대표적인 시계열에 대하여 본 논문의 알고리즘을 적용하였다.

#### 3.2.1 적용 결과

제 2.5절의 시계열 섭동 모델링 알고리즘에 의한 시계열 모델링의 결과를 <표 2>에 정리하였다. 시계열을 식별하기 위하여 축약 기호를 사용하였다. Sun = 태양 표면에서 관측된 격리 태양광 세기(solar beam isolation flux), Energy =

특정 빌딩안에서의 에너지 소모량, Laser = $NH_3$ 레이저의 레이저광 세기 변동 진폭(Intensity fluctuation), Heart = 인간 환자의 심장 박동수, Curr. = 스위스 프랑과 미 달러의 외환 교환율, Part. = 4D 포텐셜 장에서 관측되는 양자 입자의 위치, Star = 백색 왜성 PG1195의 표면 밝기이다. 이들 시계열에 대한 데이터 특성 분석자료는 경시대회의 관련자료[1]를 참조하기 바란다.

시계열 벡터의 차원, lag spacing 및 lead time $T$ 는 알고리즘 효용성 검증에서 중요하지 않으므로 단순 가정치로 고정되었다. Santa Fe 경시대회의 경우 시계열 식별 문자열의 수는 식 (1)의 n을 나타낸다.

〈표 2〉 주요 시계열 데이터에 대한 적용결과

| 경시대회명 | 시계열 | 데이터 영역별 모델 성능 | | | |
|---|---|---|---|---|---|
| | | $T$ | $V$ | $F$ | |
| | | DTS 사용 | | DTS 사용 | DTS 미사용 |
| ASHRAE | Sun | 0.005 | 0.001 | 0.002 | 0.002 |
| | Energy | 0.032 | 0.039 | 0.054 | 0.075 |
| Santa Fe | Laser, 1 | 0.007 | 0.018 | 0.015 | 0.016 |
| | Laser, 4 | 0.001 | 0.003 | 0.004 | 0.004 |
| | Heart, 1 | 0.065 | 0.190 | 0.165 | >> 1 |
| | Heart, 4 | 0.178 | 0.259 | 0.355 | Infinite |
| | Curr., 1 | 1.542 | 1.666 | 1.247 | 35.88 |
| | Curr., 4 | 8.364 | 7.878 | 15.39 | Infinite |
| | Part., 1 | 0.023 | 0.033 | 0.076 | 12.54 |
| | Part., 4 | 0.699 | 0.354 | 0.154 | Infinite |
| | Star, 1 | 0.006 | 0.008 | 0.033 | 0.028 |
| | Star, 4 | 0.002 | 0.001 | 0.002 | 0.002 |

3.2.2 결과에 대한 고찰

Sun, Laser 및 Star는 모두 안정적인 시스템 거동을 가지는 안정적 시계열 혹은 결정론적 시계열(*stationary* 혹은 *deterministic time series*)로 분류되고 있다[1, 2]. 이들 시계열들은 본 논문의 시계열 섭동 모델링 알고리즘에 의한 모델 성능이 좋은 것으로 판명되었다. 허용된 모델링 회수가 모두 소진되기 전에 타겟 성능이 달성되었다.

Santa Fe 시계열의 경우, 시계열 벡터의 차원을 달리하여 모델의 성능에 영향을 주는지 조사하였다. 한계는 있지만 결정론적 시계열에 대하여 시계열 벡터의 차원을 증가시킬수록 섭동 모델링은 좋은 성능을 보이는 것으로 관찰되었다. 표에는 n = 1 및 4의 경우를 보인다.

시스템 거동이 불안정, 불규칙한 시계열의 경우에는 시계열 벡터의 차원증가는 모델 성능저하로 이어졌다. 증가된 차원에 의해 도입되는 불안정성 및 불규칙성이 한정된 계산규모의 유용성을 떨어뜨린 결과라고 판단된다. 이와 같은 고찰은 위 표에서 시계열 별로 서로 다른 성능을 보인 점을 설명해주고 있다.

위표에서 또 한가지 주목할 사항은 유도 터미널 집합의 사용은 결정론적 시계열의 성능개선에 별다른 영향을 주지 못한 반면, 불안정 시계열에 대해서는 상당한 성능개선 효과가 있다는 사실이다. 이것은 DTS를 도입함으로써, 단순함수를 표현하는 기호구조체의 형성에 소모되는 계산자원 절약효과가 안정 시계열의 경우보다 불안정 시계열의 경우에서 훨씬 크다는 점을 시사하고 있는 것으로 평가된다. 안정 시계열의 경우 DTS 없이도 비교적 좋은 성능의 모델이 가능한 것이다.

앞서 언급했듯이 본 논문에서는 예측 영역의 성능을 개선하는 방법으로서 예측위치 이전까지의 데이터를 가능한 한 사용하여 모델의 유용성이 유지되는 영역을 확장해가는 소위 Update Extension[16]을 사용하였다. 제 2.5절 Step 7에서, 알아야 할 최신 데이터 위치를 impact step으로 개념정의한 바 있다.

원래의 경시대회에서는 예측값을 이용하여 재 예측을 수행하는 자기확장방법(Runaway Extension[1, 16])을 사용토록 하였다. 이 방법은 아주 조밀한 상태공간(state space)[17]를 형성하여 대규모의 계산이 가능한 환경에 적합하다. 제안된 알고리즘의 시험적 적용에 초점을 두어 200개의 시계열 벡터만을 사용하는 본 논문의 경우에는 적용할 수 없었다.

실제 모델링 기법의 유효성 측면에서는 어느 방법이 더 유리하다고 할 수 없다. 주어지는 데이터의 양과 모델구축을 위해 가용한 시간이 어느정도인가에 따라 적절한 방법을 선택하여 사용해야 할 것이다. Runaway extension은 모델링에 많은 시간과 자원이 소모되는 대신 자기 확장적인 예측에 용이하다. 반면, Update extension은 데이터량 혹은 계산자원이 부족하거나, 모델구축의 신속성이 요구되는 경우에 사용하면 좋다. 현대의 계산기 성능 발전속도를 생각하면 Update extension을 사용한 모델링이 점점 더 실용적인 선택이 되어 갈 것 것으로 보인다.

3.3 미국 경제 시계열

경제 시계열은 많은 시계열 중에 아주 일부분을 구성하지만 그 중요성은 아무리 강조해도 지나치지 않다. 그러나, 보통 경제 시계열은 매우 불안정하고 휘발성이 강하기 때문에, 경제 시계열에 대한 유용한 단기 모델의 구축이 가능한 것인지에 대한 지속적인 논쟁이 있다[18].

본 논문에서 제안된 시계열 섭동 모델링 알고리즘을 미국의 18개 주요 경제 시계열에 시험적으로 적용한 바가 있다[19]. 시험에 사용한 시계열은 인터넷 사이트 http : //www.economagic.com에서 다운로드 받을 수 있다.

시계열 벡터의 차원, 즉 Embedding dimension은 모두 1로 고정되었다. 이것은 앞절의 Santa Fe 시계열의 경험상 불안정한 시계열의 경우 시계열 벡터 차원의 증가는 계산자원이 한정된 상황에서 우수한 모델구축에 결코 바람직한 결과를 가져오지 못한다고 판단되었기 때문이다. 벡터의 차

원을 줄인다는 것은 식 (1)에서 변수의 개수를 줄이는 것과 동일한 의미이므로 계산시간의 절약도 기대할 수 있다.

계산 규모 및 기타 패러미터는 앞절의 ASHRAE 및 Santa Fe 경시대회의 시계열 모델링과 동일하도록 유지하였다. 그러나, 이들 시계열의 데이터 특성에 대하여 전혀 분석된 바 없으므로 *간이 모델링(pre-modeling)*개념을 도입, 시계열 데이터 특성에 관한 개괄적인 정보를 얻고자 하였다.

간이 모델링은 시계열 데이터의 특성에 대한 사전 지식을 확보하기 위해 축소된 수(예를 들어, 100)의 벡터 시계열에 대하여 모델링 과정을 적용하는 것을 말한다. 경제 시계열이라 하여도 데이터의 특성상 상대적으로 안정한 데이터가 있을 수 있다. 간이 모델링의 결과 보다 높은 성능의 모델이 구축된다면, 그 시계열은 상대적인 의미에서 그만큼 안정적인 시스템 거동을 보이는 것으로 해석할 수 있을 것이다. 이것은 앞절에서 Santa Fe 경시대회의 시계열에 대한 적용경험에 근거한 것이다.

연방준비율(Federal Fund Rates FFR), 일본 엔과 미국 달러와의 교환율 YENDOL, 그리고 30년 평균 재무 지표(Treasury Constant Maturity 30YTCM)는 매우 어려웠다. 상대적으로 불안정한 다이내믹스를 가지는 것으로 판단되며 동일한 예측성능을 가지는 모델구축에 보다 많은 계산자원이 필요하였다.

<표 3>은 시계열 섭동 모델링 알고리즘의 적용 사례이며, FFR데이터에 있어서 훈련영역 데이터 양에 따른 예측성능 개선 추이를 보여주고 있다. NT 는 훈련용 데이터 영역에서의 NMSE 를 나타내고, NF는 예측 데이터 영역(훈련 영역 이후의 100개의 시계열 벡터)에서의 NMSE를 나타낸다.

〈표 3〉 FFR에 대한 적용결과

| Data Size | $\tau$ = 1 Month | | $\tau$ = 6 Month | | $\tau$ = 12 Month | |
|---|---|---|---|---|---|---|
| | NT | NF | NT | NF | NT | NF |
| 100 | 0.096 | 0.176 | 0.534 | 10.62 | 0.812 | 3.602 |
| 200 | 0.022 | 0.082 | 0.244 | 0.485 | 0.503 | 0.748 |
| 300 | 0.030 | 0.012 | 0.206 | 0.232 | 0.352 | 0.627 |

훈련용 데이터의 양이 많아진다는 것은 상태공간의 밀도가 그 만큼 높아진다는 것을 의미하며, 상태공간의 밀도가 높을수록 더 정확한 시스템 거동의 모델링이 가능하다는 연구[16]에서 미루어 볼 때 이것이 모델의 성능개선의 원인으로 판단된다.

## 4. 결 론

본 논문은 양자역학 파동방정식의 해를 구하는 섭동이론[7,8]과 시계열 모델링의 유사성에 착안하여, 유전자 프로그래밍을 이용한 새로운 시계열 모델링 알고리즘을 제안하였다. 섭동이론은 보통은 정해를 가지지 않은 매우 복잡한 시스템 방정식의 해를 구하기 위한 효율적인 방법론을 제시하고 있으며, 시계열 모델링 또한 미지의 시스템 거동으로부터 생성되는 수치 데이터에서 원래의 시스템 거동을 기술하는 모델을 구축하기 위한 방법론이라는 점에서 유사성을 가진다.

제 2.5절에서 정리한 시계열 섭동 모델링 알고리즘에서는 시계열 자체가 섭동이론에 있어서의 파동방정식의 역할을 하게 된다. 유전자 프로그래밍에 의해 구축되는 기호구조체는 섭동이론의 해밀토니언의 기능을 하게 된다.

비 섭동된 시계열 모델 후보들을 섭동이론에서의 비섭동 해밀토니언을 구하는 것과 동일하게 구축되고, 이것들을 주어진 시계열 데이터에 대하여 최소자승법을 적용하여 계산된 수치계수로서 선형결합, 식 (9) 참조, 하여 최종적인 시계열 모델로 결정한다.

예측영역에서의 예측은 하나의 데이터를 예측할 때마다 그 이전까지 알려진 최신 데이터를 전부 고려하여 재계산된 최소자승법 계수를 이용하여 수행하는 소위 update extension을 사용한다. DTS를 이용하면 특히 불안정한 시계열에 대하여 모델 성능이 개선되는 것으로 밝혀졌다.

제안된 알고리즘은 물리시스템에서 경제시스템에 이르기까지 다양한 시스템에 발생하는 실세계 카오스 시계열에 대하여, 단순 가정(simplistically assumed)된 상태공간 패러미터와 제한된 계산자원을 사용하였음에도 비교적 성공적인 결과를 얻었다.

모든 시계열 모델링 알고리즘은 적절한 데이터 특성화 기법과 결합되었을 때 좋은 성과를 거둘 수 있다. 데이터 특성화의 결과 상태공간에 대한 최적화 패러미터를 구할 수 있으므로, 본 논문에서와 같이 단순 가정한 패러미터를 사용하는 경우보다 개선된 성능의 모델구축이 가능한 것은 자명한 것이다.

본 논문에서는 간이 모델링(pre-modeling)을 사용하여 특정 시계열의 데이터 특성에 대한 최소한의 정보, 즉 상대적인 시계열 데이터 안정성을 판단할 수 있었다. Santa Fe 시계열 경시대회에서 사용한 시계열에 대하여 제안된 알고리즘을 적용한 결과 동일한 계산규모와 패러미터를 사용하는 모델링과정에서 보다 성능이 개선된 모델이 구축되는 것은 주어진 시계열이 그만큼 상대적으로 안정적(stationary)이기 때문인 것으로 판단되었으며, 이러한 관찰은 다른 데이터 특성화 기법의 결과와도 일치하였다[1].

그러나, 본 논문에서는 유전자 프로그래밍의 특성을 결정하는 다양한 계산 패러미터의 변화에 따른 모델링 성능영향을 분석하지 못하였고, 이에 대한 것은 향후의 연구과제로 넘긴다.

## 참 고 문 헌


[1] Weigend, A. S. and Gershenfeld, N. A., Eds., "Time Series Prediction Forecasting the future and Understanding the Past," SFI Studies in the Science of Complexity, Vol.XV, Addison Wesley Publishing Co., 1993.

[2] Box, G. E. P., Jenkins, G. M. and Reinsel, G. C., "Time Series Analysis," 3rd Ed., Englewood Cliffs, NJ : Prentice Hall, 1994.

[3] Ivakhnenko, A. G., "Polynomial Theory of Complex Systems," IEEE Trans. Syst. Man Cybern., Vol.1(4), pp.364-378, 1971.

[4] Koza, J. R., "Genetic Programming II," MIT Press, 1994.

[5] Oakeley, H., "Two Scientific Application of Genetic Programming : Stack Filters and Non-Linear Equation Fitting to Chaotic Data," Advances in Genetic Programming, K. E. Kinnear Jr., Eds., MIT Press, pp.369-389, 1994.

[6] Iba, H., de Garis, H. and Sato, T., "Genetic Programming using a Minimum Description Length Principle," Advances in Genetic Programming, K. E. Kinnear Jr., Eds., MIT Press, pp.265-284, 1994.

[7] Rae, A. I. M., "Quantum Mechanics," 3rd Ed., University of Birmingham, UK, IOP Publishing Ltd., 1992.

[8] Nayeh, A. H., "Introduction to Perturbation Techniques," John Wiley & Sons, 1993.

[9] Geman, S. et al., "Neural Networks and the Bias / Variance Dilemma," Neural Computation, Vol.4, pp.1-58, 1992.

[10] Luke, S., and L. Spector, "Evolving Teamwork and Coordination with Genetic Programming," Genetic Programming 1996 : Proceedings of the First Annual Conference, MIT Press, pp.150-156, 1996.

[11] Guy L. Steele, "Common Lisp," 2nd Edition, 1991.

[12] Sansone, G., Sansome, G. and Diamond, A. H., "Orthogonal Functions," Dover Publications, 1991.

[13] Casdagli, M. C., "Nonlinear prediction of chaotic time series," Physics D. 35, pp.335-356, 1989.

[14] Geumyong, Lee, "Genetic Recursive Regression for modeling and forecasting real-world chaotic time series," Advances in Genetic Programming, Vol.3., Chapter 17, MIT Press, 1999.

[15] Kreider, J. F., "*results.asc*," ASHRAE Competition ftp site, ftp.cs.colorado.edu/pub/energy-shootout, 1993.

[16] Smith, L. A., "Does a meeting in Santa Fe imply chaos ?," Time Series Prediction Forecasting the Future and Understanding the Past, A. S. Weigend, and N. A. Gershenfeld, Eds., SFI Studies in the Science of Complexity, Addison-Wesley Publishing Co., Vol.XV, pp.323-343, 1993.

[17] Kailath, T., Linear Systems, Englewood Cliffs, NJ : Prentice Hall, 1980.

[18] Dechert, W. D., "Chaos Theory in Economics : Methods, Models and Evidence," International Library of Critical Writings in Economics, Edward Elgar Pub., No.66, 1996.

[19] Geumyong, Lee, "Modeling and Forecasting Major US Economic Time Series Based on Intelligent Symbolic Processing Technology," 영산대학교 출판사, 영산논총, 제4집, pp.220-226, Aug., 1999.



**이 금 용**

e-mail : office@java-tech.com

1987년 서울대학교 원자핵공학과 졸업 (공학학사)

1989년 한국과학기술원 원자력공학과 졸업 (공학석사)

1989년~1992년 한국전력기술㈜

1995년 동경대학교 시스템양자 공학과 졸업(공학박사)

1995년~1997년 한국원자력안전기술원 선임연구원

1997년~1999년 영산대학교 전자계산학과 전임강사

1999년~2001년 同 컴퓨터정보공학부 조교수

2002년~현재 同 정보통신공학부 조교수

관심분야 : J2EE, J2ME, LISP, Time Series 등