한국 CAD/CAM 학회 논문집
제9권 제3호 2004년 9월 pp. 183-191

학술논문

유전적 프로그래밍을 이용한 노이지 데이터의 Curve Fitting과 선박설계에서의 적용

이경호*, 연윤석**

Genetic Programming Approach to Curve Fitting of Noisy Data and Its Application in Ship Design

Lee, K. H.* and Yeun, Y. S.**

ABSTRACT

This paper deals with smooth curve fitting of data corrupt by noise. Most research efforts have been concentrated on employing the smoothness penalty function with the estimation of its optimal parameter in order to avoid the "overfitting and underfitting" dilemma in noisy data fitting problems. Our approach, called DBSF(Differentiation-Based Smooth Fitting), is different from the above-mentioned method. The main idea is that optimal functions approximately estimating the derivative of noisy curve data are generated first using genetic programming, and then their integral values are evaluated and used to recover the original curve form. To show the effectiveness of this approach, DBSP is demonstrated by presenting two illustrative examples and the application of estimating the principal dimensions of bulk cargo ships in the conceptual design stage.

Key words : Smooth Fitting, Genetic Programming, Curve, Ship Design

1. 서 론

일반적으로 노이즈(Noise)가 포함된 입출력 데이터로부터 그 관계를 매핑(Mapping)할 수 있는 커브(Curve)를 근사시키는 작업은 Overfitting과 Underfitting의 딜레마에 직면하게 된다. 예를 들면 데이터의 Fitting에 가장 보편적으로 쓰이는 방법은 스플라인(Spline)을 이용하는 것인데[1-4], 보통 MSE(Mean squared-error)가 최소가 되도록 스플라인의 계수 값을 결정하게 된다. 데이터에 노이즈가 포함되어 있으면, 당연히 Fitting된 스플라인 곡선의 거동은 Overfitting을 보이게 된다. 이를 극복하기 위해서 Overfitting 정도 즉 Smoothness를 측정할 수 있는 페널티 또는 Regularization 함수를 도입하여 Overfitting이 일어나지 않도록 적절한 스플라인의 계수 값을 산정하는 방법이 가장 보편적으로 사용된다. 이것은 바로 Regularization의 기여도를 적절히 결정할 수 있는 파라메터(Regularization parameter) 값을 결정하는 문제로 귀착된다. 파라메터의 값이 작으면 Overfitting이 발생하고, 반대로 과다하면 Underfitting 또는 Oversmoothing 현상이 일어난다. 따라서 최적의 파라메터 값을 구하는 것이 무엇보다도 중요하지만, 이것은 대단히 어려운 작업이기 때문에 일반적으로 GCV (Generalized Cross-Validation)[18,19]이나 기타 다른 방법[1-6,13-16]을 이용하여 파라메터 값을 계산하여 사용하게 된다.

기 발표된 논문 Yeun *et al.* (2001)[17]은 응답변 y_i의 직접적인 smooth fitting을 다룬 것이고, 본 논문에서는 위에서 언급한 방법과는 달리 데이터의 미분 치를 추정할 수 있는 근사 함수를 먼저 구축하고 이로부터 Fitting 결과를 얻는 DBSP(Differentiation-Based Smooth Fitting) 방법을 제시하고자 한다. 본 논문의 범위는 근사 식으로부터 추정될 수 있는 미분 차는 오직 1차 미분 치에 대해서만 고려하였다. 기본적인 접근법은 유전적 프로그래밍(Genetic Programming: 이하 GP)[7]을 사용하여 우선 노이지(Noisy) 데이터로부터 미분 값을 추정할 수 있는 Smooth한 거동을 보이는 GP 트리(Tree)를 먼저 생성하고, 이것을 적분하여 원하는 근사 식을 생성하는 것이다. 그러데 일반적으

*인하대학교 선박해양공학과
**대진대학교 컴퓨터응용 기계설계공학과
- 논문투고일: 2003. 03. 26
- 심사완료일: 2003. 10. 29

로 생성된 GP 트리를 수학식 형태로 변환하더라도 적분이 불가능한 경우가 대부분이기 때문에 수치적분(Numerical integral)을 대신 사용하여 Fitting 결과를 얻게 된다. 일반적으로 어떤 함수를 적분하면 보다 Smooth한 거동을 보이게 된다. 따라서 Smooth한 미분식을 먼저 생성하고, 이것을 적분하여 구한 근사식은 당연히 Smoothness를 보장 받게 된다.

GP를 적절히 활용하기 위해서는 적합도(Fitness) 함수 및 터미널(Terminal)과 함수(Function) 집합의 정의가 가장 중요하며, 2장에서는 주로 이와 관련된 내용과 더불어 미분 치를 추정할 수 있는 GP 트리 생성 알고리즘을 다루었다. 본 연구의 결과를 검증하기 위하여 3장에서는 비교적 간단한 예제들을, 4장에서는 실제적 문제의 응용을 위하여 선박의 개념설계 단계에서 주요치수(Principal dimensions) 추정 문제를 다루었다.

2. 유전적 프로그래밍을 이용한 미분식의 도출

유전적 프로그래밍 기법은 기존의 유전적 알고리즘에서 개체(Individuals)로서 이진 스트링을 사용하는 대신 컴퓨터 프로그램을 표현하기 위한 트리구조를 사용한다. 이 프로그램 트리는 함수집합(Function Set)과 터미널 집합(Terminal Set)에 정의된 원소들의 조합으로 구성된다. 여기서도 재생산(Reproduction), 교배(Crossover), 돌연변이(Mutation) 등의 유전적 연산이 수행된다. 또한 연산 과정중의 트리의 크기와 구조는 유전적 연산에 의해 동적으로 변화하는데, 이렇게 얻어진 트리는 적합도(Fitness) 함수에 의해 평가된다.

노이즈가 포함된 데이터로부터 미분 치 추정 시 노이즈가 크게 증폭되는 효과가 있기 때문에, FDM (Finite Difference Method)에서 추정된 값은 실제 값과 비교하여 매우 큰 차이를 보이게 된다. 즉 노이지(Noisy) 데이터의 미분 값을 추정하는 문제는 Ill-posed된 문제로 분류된다. 노이즈 데이터의 미분 치 추정을 위해서 수 많은 연구가 수행되었는데, 크게 다항식(Polynomial)이나 스플라인 등으로 데이터를 적절히 Fitting한 후 이로부터 미분 치를 구하는 방법[1-4]과 디지털 필터링(Digital Filtering) 기법을 이용하여 노이즈를 제거한 후 FDM으로 미분 치를 구하는 방법[8,9] 등으로 대별될 수 있다.

본 논문에서는 기존의 방법과는 달리 미분 식으로써의 GP 트리가 만족해야 할 적합도(Fitness)를 정의하고, 진화 과정을 통하여 Smooth한 거동을 보임과 동시에 최적의 적합도 값을 갖는 GP 트리 생성 방법을 제시하고자 한다.

GP의 개체(Individual)는 트리 형태가 되는데, 함수와 터미널 집합에 정의된 원소들의 조합으로 구축된다. 유전적 연산자들(Genetic operators)이 GP 트리에 적용되어 적합도 함수를 최적화하는 방향으로 GP트리의 구조 자체가 동적으로 변화하게 된다[7]. 따라서 적절한 미분 식 탐색을 위해서는 적합도 함수, 터미널과 함수 집합의 정의가 무엇보다도 중요하다.

2.1 적합도 함수의 정의

어떤 곡선 $F(x)$가 주어면, $x = x_i$일 때 F의 값은 F_i로 표기할 수 있다. 보통 계측된 F_i의 값 $\tilde{F}_i$는 노이즈 e_i를 포함하게 되므로 식 (1)로 표현된다.

$$\begin{aligned}\tilde{F}_i &= F(x_i) + e_i \\ &= F_i + e_i\end{aligned} \tag{1}$$

계측된 샘플의 개수는 $n+1$이며, $x_0 < x_1 < \ldots < x_n$의 관계가 성립한다. x_{i+1}과 x_i의 간격은 h_i인데, 등 간격이 되어야 한다는 제한조건은 없다. GP 트리를 f라고 표기할 때, $dF/dx = f$의 관계가 성립될 수 있는 f를 찾기 위해서는 다음과 같은 2개의 적합도 함수를 고안할 수 있다.

1) 적합도 함수 1

F와 f 사이에는 $\int_{x_i}^{x_{i+1}} f(x)dx = F(x_{i+1}) - F(x_i)$ $(i = 0, \ldots, n)$ 이 만족되어야 하므로 다음과 같은 적합도 함수 ϕ_1을 정의할 수 있다.

$$\phi_1 = \frac{1}{n}\sum_{i=0}^{n-1}\left[\int_{x_i}^{x_{i+1}} f(x)dx - (\tilde{F}_{i+1} - \tilde{F}_i)\right]^2 \tag{2}$$

2) 적합도 함수 2

$\alpha_i = \int_{x_i}^{x_{i+1}} f(x)dx$ 라고 하면 다음과 같은 식들이 만족됨을 쉽게 파악할 수 있다.

$$\begin{aligned}
F_0 &= F_1 - \alpha_0 = F_m - \alpha_0 - \alpha_1 - \ldots - \alpha_{m-1} \\
F_1 &= F_2 - \alpha_1 = F_m - \alpha_1 - \alpha_2 - \ldots - \alpha_{m-1} \\
&\cdots\cdots \\
F_{m-1} &= F_m - \alpha_{m-1} \\
F_m &= F_m \\
F_{m+1} &= F_m + \alpha_m \\
F_{m+2} &= F_{m+1} + \alpha_{m+1} = F_m + \alpha_m + \alpha_{m+1} \\
&\cdots\cdots \\
F_n &= F_{n-1} + \alpha_{n-1} = F_m + \alpha_m + \alpha_{m+1} + \ldots + \alpha_{n-1}
\end{aligned}$$

위의 식들을 모두 더하여 정리하면 다음과 같이 F_m

에 대한 식이 얻어진다.

$$F_m = \frac{1}{n+1}\left[\sum_{i=0}^{n} F_i + \sum_{i=0}^{m-1}(i+1)\alpha_i - \sum_{i=0}^{n-m-1}(n-m-i)\alpha_{i+m}\right] \quad (3)$$

여기서 만일, $\frac{1}{n+1}\sum_{i=0}^{n}\bar{F}_i = \frac{1}{n+1}\sum_{i=0}^{n}(F_i+e_i) = \frac{1}{n+1}\sum_{i=0}^{n}F_i$ 이 된다면, 즉 노이즈의 평균 값이 0이거나 0에 접근한다면 한다면 F_m은 다음과 같이 나타낼 수 있다.

$$F_m \approx \bar{F}_m = \frac{1}{n+1}\left[\sum_{i=0}^{n}\bar{F}_i + \sum_{i=0}^{m-1}(i+1)\alpha_i - \sum_{i=0}^{n-m-1}(n-m-i)\alpha_{i+m}\right] \quad (4)$$

따라서 식 (4)를 사용하여 적합도 함수 ϕ_2를 아래와 같이 정의할 수 있다.

$$\phi_2 = \frac{1}{n+1}\sum_{i=0}^{n}(\bar{F}_i - \tilde{F}_i)^2 \quad (5)$$

ϕ_2의 장점은 데이터 자체와 그 미분 치를 동시에 고려하고 있다는 점이다. 많은 세대(Generation)가 지나면 GP 트리 즉 f의 두 적합도 함수 값은 수렴하게 되는데, 이때 대체적으로 ϕ_1의 값은 ϕ_2의 값보다 2배 정도의 크기를 갖는 경향이 있다. 두 적합도 함수를 동시에 사용하기 위해서 식 (6)과 같은 적합도 함수 ϕ_3의 사용을 고려할 수 있다.

$$\phi_3 = 0.5(0.5\phi_1 + \phi_2) \quad (6)$$

일반적으로 ϕ_1만을 적합도 함수로 사용하면 GP 트리는 Overfitting 현상을 보이게 되는데, 2.3절에 기술되어 있듯이 Smooth한 거동을 보이는 트리를 생성하기 위해서는 ϕ_1에 Regularization term을 첨가하여 수정할 필요가 있다.

2.2 터미널 및 함수 집합

터미널 집합은 $T = \{x, R\}$인데, x는 독립변수, R은 $|R| < 1.0$ 인 난수(Random Number)이다. 함수 집합의 구성 시 Smooth한 GP 트리의 출력 값이 얻어지는데 도움을 줄 수 있도록 적절한 함수를 선정해야 하는데 본 논문에서는 다음과 같이 함수 집합을 구성하였다.

$$F = \{+, -, *, g_1, g_2, g_3, g_4\}$$

여기서 $g_1 = a_1(b_1 x - c_1)^2$

$g_2 = a_2 / e^{[(x-b_2)/c_2]^2}$

$g_3 = a_3/(1+e^{-b_3(x-c_3)})$

$g_4 = (a_4/b_4^2)\sqrt{b_4^2 + (1-c_4 x)^2}$ 이다.

a_i, b_i, c_i는 수치 파라메터를 의미하는데, 이들 값을 추정하는 방법은 2.3절에 기술되어 있다.

2.3 미분 식 탐색을 위한 GP 알고리즘

2.2절에서 기술한 터미널과 함수 집합을 사용하여 GP 트리들이 생성되고 각 트리의 적합도를 계산한다. 이때 함수 집합에 사용된 함수 g_i의 파라메터들을 적합도 값이 보다 최소가 되도록 추정할 필요가 있는데 계산량을 줄이기 위하여 LAM(Linear Associative Memory)[10-12]와 Hooke & Jeeves 탐색 방법을 혼용하여 사용하는 방식을 취하였다. 그런데 여기서 문제가 되는 것은 추정된 파라메터 값에 따라서 Overfitting 현상, 즉 트리의 적합도 값이 매우 작으나 트리의 출력 값이 정확한 미분 치에서 크게 벗어나는 일이 발생한다. 따라서 Overfitting에 관련된 패널티를 적합도 함수에 부가하여 GP 트리가 Smooth한 거동을 보이도록 유도할 필요가 있다. 본 연구에서는 다음과 같은 식을 사용하였다.

$$\phi_3' = \phi_3/\mu + \nu L/\kappa \quad (7)$$

여기서

$L = \int_{x_0}^{x_n}[1 + (df/dx)^2]dx - (x_n - x_0) = \int_{x_0}^{x_n}(df/dx)^2 dx$

이고, μ, κ는 각각 ϕ_1, L(L은 f의 길이와 밀접한 관계가 있다)을 스케일링(Scaling)하기 위한 상수이며, ν는 L의 기여도를 나타내는 Regularization 파라메터이다. 만일 f가 독립변수 x에 대하여 수평선이 된다면 (Underfitting의 가장 극단적인 형태) L의 값은 0이 되고, 반대로 Overfitting이 심할수록 L의 값은 증가하게 된다. μ, κ 그리고 ν 값의 결정하는 방법과 더불어 최적의 f를 탐색하기 위한 GP 알고리즘을 간략히 요약하면 다음과 같다.

a. 임의의 개체(트리) N개를 생성하여 개체군(Population)을 만든다. 이때, 트리가 함수 g_i를 포함하고 있으면 a_i, b_i, c_i는 적합도 함수 ϕ_2가 최소가 되도록 LAM를 사용하여 추정한다. 파라메터 추정 시 과다한 계산량을 회피하기 위하여 비선형 최적화 알고리즘의 사용을 배제하고 LAM를 사용하였다[10-12].

b. 개체군에서 M개(보통 5-10개 정도)의 최적 트리들을 선택하고, g_i를 포함하고 있으면 Hooke & Jeeves 기법을 사용하여 a_i, b_i, c_i를 보다 정확히

추정하는데, 적합도 함수는 ϕ_3'를 사용한다. Hooke & Jeeves 방법을 사용하기 전에, μ, κ 그리고 ν 값을 미리 결정해 주어야 하는데, 본 논문에서는 μ와 ν 값은 트리의 ϕ_3 값으로 κ는 트리의 L을 계산하여 사용하였다. 이와 같이 파라메터 값의 설정은 이론적인 근거에 의한 것이 아니고 본 연구 수행의 경험에 기초한 Heuristic한 것이다. 따라서 최적의 값이 될 수 없으며 많은 세대가 지나면 결국 개체 f의 거동은 Overfitting을 보이게 되는 것이 일반적이다.

c. 모든 개체에 대하여 3가지 유전적 연산자들(재생산, 교배, 돌연변이)을 적용하여 새로운 개체군을 생성하고, 주어진 세대 G가 될 때까지 a, b 과정을 반복한다.

d. c 과정을 마치면 가장 최적의 트리 즉 가장 최소의 ϕ_3'을 갖는 트리를 하나 선정하는데, b과정에서 언급했듯이 이 개체는 Overfitting 현상을 보일 가능성이 크다. Overfitting을 줄이기 위해서 ϕ_3'의 가장 중요한 파라메터는 ν인데, 보다 적절한 값의 산정을 위해서 보통 통계적인 기법인 LOOCV (Leave-One-Out Cross-Validation)[1-3]의 사용을 고려해 볼 수 있다. 그런데 LOOCV의 단점은 방대한 계산량을 요구한다는 것이다. 더욱이, 실제로 본 연구에서 LOOCV을 이용하여 ν 값을 추정하여도 원하는 결과를 얻기 어려웠다. 따라서 본 논문에서는 LOOCV 대신에 ν을 점차적으로 E(0.5-0.1 정도) 만큼씩 증가시키면서 Hooke & Jeeves 방법을 사용하여 ϕ_3'가 최소가 되도록 트리에 포함된 a_i, b_i, c_i를 결정하는 방식을 채택하였는데, 일반적으로 ν의 증가에 따라서 ϕ_3'의 값도 증가하게 된다. 이때 초기의 ϕ_3'(ν를 증가시키기 전)과 ν의 증가에 따른 현재의 ϕ_3'와 비교하여 그 변화된 양이 P%(보통 1-5% 정도) 이하가 될 때까지만 ν를 증가시켜 사용한다. 이와 같은 방법을 사용하여 최적의 결과를 얻을 수 있는 것은 아니지만 트리의 Overfitting 현상은 크게 줄일 수 있는 장점이 있다.

GP 트리의 적합도를 계산하는 식들을 살펴보면 미분과 적분 값을 계산해야 되는데, 본 논문에서는 모두 수치 미분과 적분 법을 이용하였다. 또한 GP 트리로부터 $F(x)$ 값을 추정하기 위해서는 GP 트리를 적분해야 되는데, 적분 식의 도출이 거의 불가능하기 때문에 $F(x)$ 값을 계산하기 위해서 수치 적분 법을 사용하였다.

3. 검증 예

GCV(Generalized cross validation)는 LOOCV (Leave one out cross validation)과 밀접한 관계가 있는데, 먼저 LOOCV를 간결히 예를 설명하면 다음과 같다. 만일 100개의 noisy 데이터가 있다고 가정하면, 첫 번째 데이터 한 개를 뺀 나머지 99개의 데이터로 근사모델을 만든다. 이 근사모델의 응답과 첫 번째 데이터와의 차이로부터 테스트오류를 계산한다. 다시 100개의 원래 데이터로부터 두 번째 데이터를 뺀 99개의 데이터로 새로운 근사모델을 만들고 앞에서와 같은 방식으로 두 번째 데이터에 대한 테스트 오류를 계산한다. 이 과정을 계속해서 반복적으로 수행하면 100개의 근사모델과 이에 대응하는 100개의 데이터에 대한 테스트 오류를 계산할 수 있다. 이 테스트 오류를 CV(Cross validation) 오류라고 하며, CV 오류를 가장 작게 줄일 수 있는 Regularization parameter의 값을 구하여 사용하는 것이 가장 흔히 사용되는 방법이고, 또한 다른 특별한 방법의 사용이 여의치 않을 때 Smooth fitting을 얻을 수 있는 일반성이 있는 방법이다. 그런데 이 LOOCV의 가장 큰 문제점은 막대한 계산량을 요구한다는 것인데, 최적의 Regularization parameter의 값을 구하기 위해서는 최적화 과정이 필요하고, Regularization parameter 값의 탐색 시 마다 데이터의 개수만큼 근사 모델을 구축해야 한다는 점이다. 그런데 Spline인 경우 CV 오류를 Closed form의 공식을 유도할 수 있는데, 이를 사용한 경우를 GCV Spline이라고 한다.

본 절에서는 2가지 예제를 사용하여 DBSF 방법과 가장 일반적으로 사용되고 있는 B-Spline에 GCV를 채용한 패키지[3](GCVSPL Fortran Package를 C 언어로 변환한 프로그램을 사용하였음)를 사용하여 그 결과를 비교 검토하였다. 사용된 노이즈는 평균이 0,

Table 1. Parameters used in genetic programming algorithms

Population Size(N)	1000
Number of Generations(G)	10
Selection method	Tournament with 50 trees
Reproduction probability	0.2
Crossover probability	0.75
Mutation probability	0.05
M in Section 2.3	10
E in Section 2.3	0.5
P in Section 2.3	1%

분산이 σ^2인 가우시안 노이즈(Gaussian Noise) 즉 $e_i \sim N(0, \sigma^2)$이다. 본 절에서 GP 알고리즘에 사용된 파라메터들은 Table 1에 나타나 있다.

3.1 예제 1

식 (8)과 같은 커브의 값들을 계산하여 $\{F(t_0), \ldots, F(t_n)\}(n=50,\ t_{i+1}=t_i+\pi/50,\ t_0=0.0)$을 구축한 후, $\sigma^2=0.05$와 $\sigma^2=0.1$인 노이즈를 각각 중첩하여 학습 집합 $\{\tilde{F}_0, \ldots, \tilde{F}_n\}$을 만들었다.

Table 2. Results of smooth fitting based on DBSF GP and GCV B-spline. Here, the measure of the goodness is the mean squared-error of the true and estimated value at given learning samples

σ^2	F''		F	
	GP	B-spline with GCV	GP	GCV B-spline
0.05	0.070232	0.178794	0.003555	0.005513
0.1	0.365266	1.399405	0.019151	0.021899

$$F(t) = 2\sin^2(t) \tag{8}$$

이 데이터로부터 GP 및 B-Spline(4^{th} order)을 사용하여 추정된 F와 F''에 대한 결과를 요약하면 Table 2와 같다. Fig. 1은 그 결과를 가시화한 것이다.

F와 F''의 추정에 있어서 GP에 의한 결과가 B-spline의 결과보다는 우수함을 쉽게 파악할 수 있는데, 특히 노이즈가 $N(0,0.1^2)$인 경우 B-spline은 Overfitting 거동을 보이는 반면 GP 트리는 식 (8)의 원래 형태에 근사적으로 접근하고 있다. 그러나 GP 트리와 B-spline의 F에 대한 MSE는 각각 0.019151과 0.021899로써 큰 차이를 보이고 있지는 않은데, 이것은 t_0와 t_{50} 근처에서 GP 트리가 비교적 큰 오차를 보이고 있기 때문이다.

3.2 예제 2

식 (9)과 같은 Van der Pol 식에서 $r(t)$가 Unit Step 함수들의 조합으로 이루어진 때 그 해 $\{F(t_0), \ldots,$

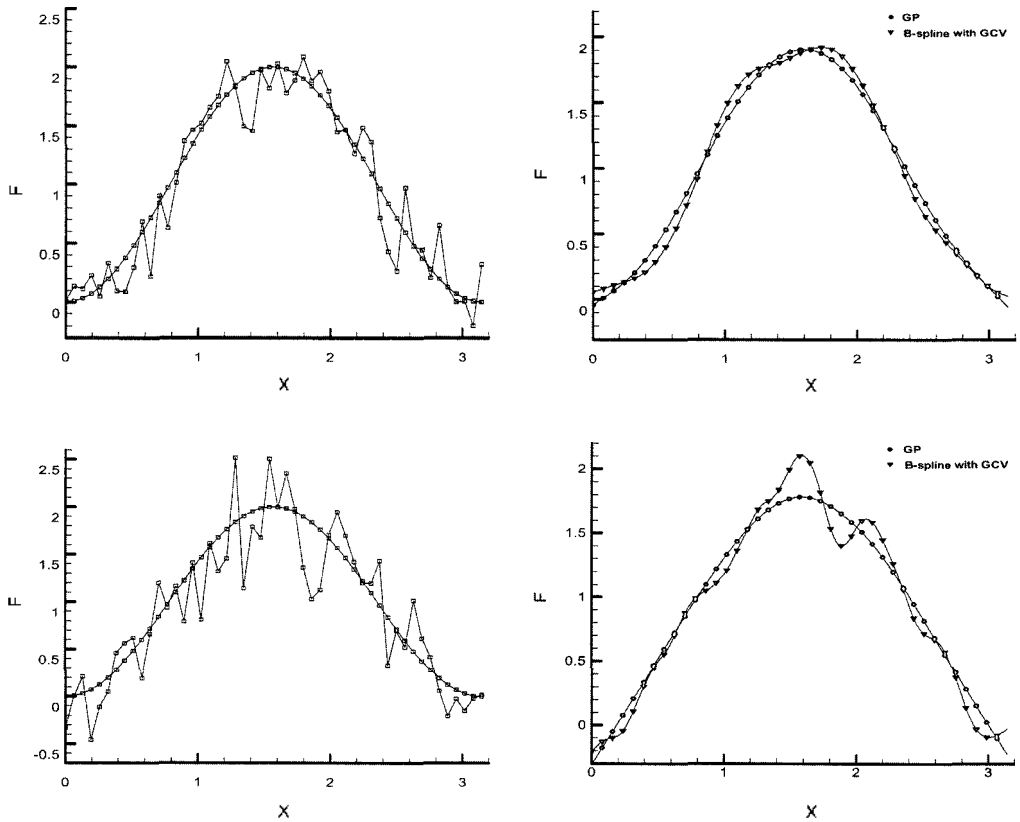


Fig. 1. Fitting by GP and B-spline with GCV. The first picture shows the equation (8) with noise $e_i \sim N(0,0.05^2)$, and the third is the case of $e_i \sim N(0,0.1^2)$.

$F(t_n)\}(n=50,\ t_{i+1}=t_i+0.1,\ t_0=0.1)$를 4th Order Runge-Kutta Method로 구한 후, $\sigma^2=0.05$와 $\sigma^2=0.1$인 노이즈를 각각 중첩하여 $\{\tilde{F}_0,\ldots,\tilde{F}_n\}$를 만들었다.

$$F''+F'+F(F-1)=r(t) \tag{9}$$

이 데이터로부터 GP 및 B-Spline(4^{th} order)을 사용하여 추정된 F와 F'에 대한 결과를 요약하면 Table 3과 같다. Fig. 2는 그 결과를 가시화한 것이다. 예제1에서와 같이 F와 F'의 추정에 있어서 GP에 의한 결과가 B-spline의 결과보다 우수하다는 사실을 알 수 있다. 노이즈가 $N(0,0.1^2)$인 경우, B-spline은 GCV를 채용했어도 Smooth fitting에는 완전히 실패한 반면 GP의 결과는 매우 정확한 결과를 보여주고 있다.

Table 3. Results of smooth fitting based on DBSF GP and GCV B-spline. Here, the measure of the goodness is the mean squared-error of the true and estimated value at given learning samples

σ^2	F'		F	
	GP	B-spline with GCV	GP	B-spline with GCV
0.05	0.212826	0.821119	0.012326	0.023756
0.1	0.112509	6.918122	0.007902	0.054719

4. 선박설계에서의 적용 예 : Bulk Carrier의 개념 설계

선박의 개념설계 단계에서는 선박의 길이, 폭 그리고 깊이 등을 과거의 실적선 자료를 토대로 추정하게 된다. 이와 같은 주요 치수들(Principal dimensions)은 일반적으로 재화중량(Deadweight)의 함수들로 여겨지는데, 실제 실적선 데이터에서도 Fig. 3에 나타나 있듯이 재화중량에 대하여 일정한 경향을 보여주게 된다. 그런데, Fig. 3에서 알 수 있듯이 실적선 데이터는 마치 노이즈가 포함되어 있는 것 같은 경향을 보이고 있는데, 이것은 선주의 특별한 요구조건이나 기타 다른 요인들이 반영된 설계와 생산에 기인한 것으로 판단된다. 따라서 개념설계 단계에서는 이런 노이즈 효과를 제거한 후 주요 치수들을 추정하는 것이 중요하다고 할 수 있다. 본 절에서는 DBSF 기법을 이용하

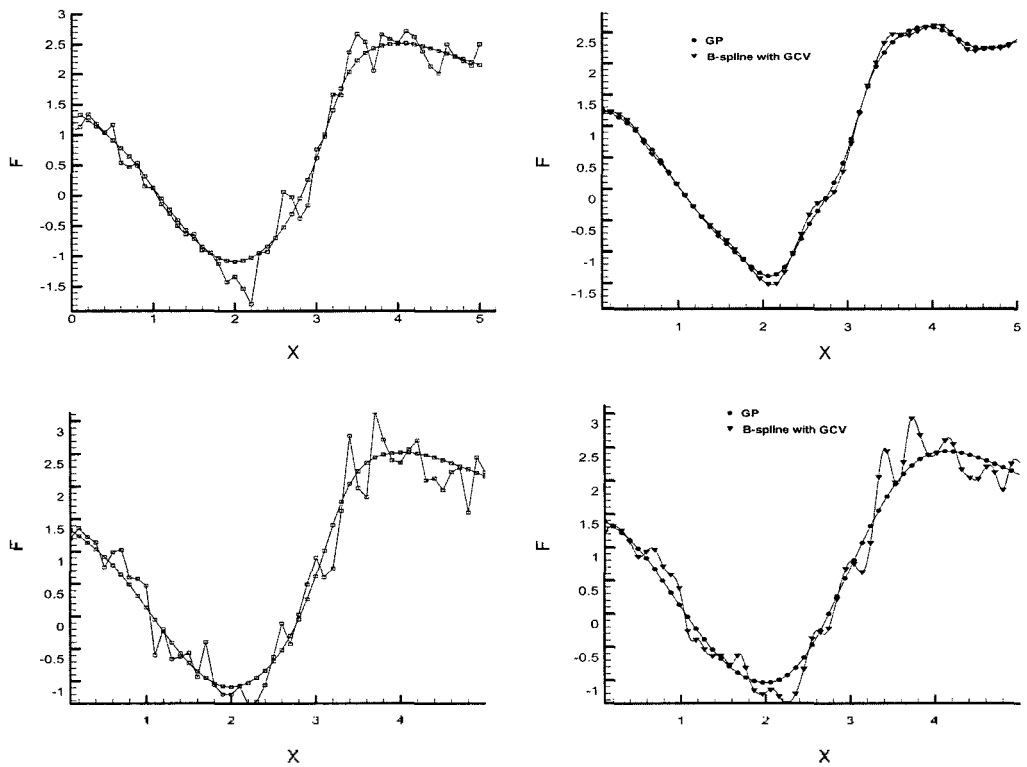


Fig. 2. Fitting by GP and B-spline with GCV. The first picture shows the equation (9) with noise $e_i \sim N(0,0.05^2)$, and the third is the case of $e_i \sim N(0,0.1^2)$.

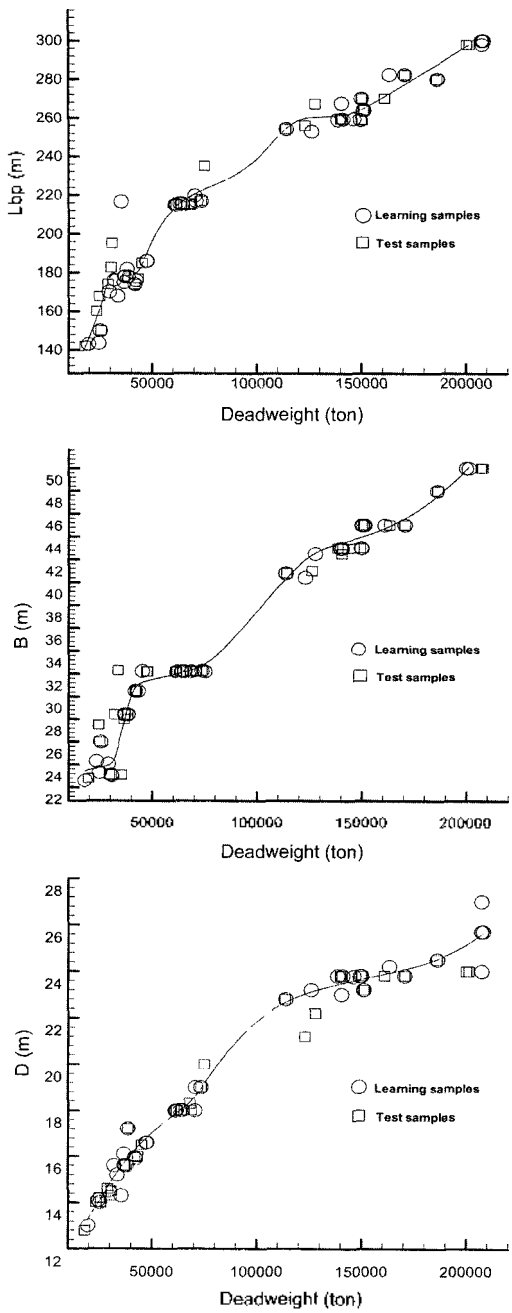


Fig. 3. The estimation of principal dimensions of bulk cargo ships by DBSF GP.

여 산적화물선(Bulk Carrier)의 주요치수를 추정하고자 한다. 100 척의 실적선 자료 중 절반은 GP 트리 생성에 사용하였고 나머지 절반은 결과를 검증하는데 사용하였다. GP 알고리즘에 사용된 파라메터들의 값은 Table 1과 동일하다. GCV B-spline은 Smooth fitting에 완전히 실패하였기 때문에 본 논문에서는 그 결과를 싣지 않았다. Fig. 3은 실적선 데이터와 함께 50개의 학습 샘플을 이용하여 학습된 GP트리의 결과를 표시한 것이다. 여기서 Lbp, B와 D는 각각 산적화물선의 Length between perpendiculars, Breadth 그리고 Depth를 의미한다. Cb(Block coefficient)의 자료는 보통 대외비로 취급되기 때문에 충분한 데이터를 확보할 수 없어 본 논문에서는 Cb에 관련된 사항은 제외 시켰다. Fig. 3을 살펴보면 몇몇의 실적선 데이터는 전체적인 경향에서 크게 이탈되고 있는데, 대부분의 기계학습(Machine learning) 기법은 이런 데이터에 악영향을 받게 된다. 그러나 DBSF GP의 결과는 이런 데이터에 크게 영향을 받지 않을 뿐만 아니라 대부분의 학습과 테스트 샘플의 분포 경향을 대체적으로 잘 반영하고 있음을 관찰할 수 있다.

5. 요약 및 결론

본 논문에서 직접적인 언급은 되어 있지 않지만 GP(Genetic Programming)을 사용한 이유 중의 하나는 GP는 Global model(모든 학습 데이터를 한꺼번에 사용하여 fitting)의 일종이라는 점이다. 반면 spline, spline과 수학적으로 equivalent한 Gaussian process, kriging 또는 Locally weighted regression은 Local model이다. 본 논문에서 주요 관심이 있는 문제의 특징은 다음 두 가지 이다. 첫째, 학습 데이터가 충분하지 않다. 둘째, 학습 데이터가 균등하게 분포하고 있지 않다(실적선의 예 참조). 이런 문제에서는 Local model 보다는 Global model이 바람직하고 신뢰성 있는 결과를 준다. 쉽게 이야기 한다면 B-spline과 GP의 비교는 공정하지 않은데(위에서 언급한 특성을 갖는 문제에서 B-spline이 GP 보다 못하다는 것은 미리 예견할 수 있기 때문에), 그럼에도 불구하고 B-spline을 사용하여 결과를 비교한 이유는 다음 두 가지이다. 첫째는 Noisy data fitting 문제는 항상 overfitting과 underfitting의 딜레마에 직면하는데, fitting의 quality를 결정하는 적절한 parameter의 값의 산출이 가장 어렵고도 중요한 문제이다. GCV B-spline에서는 그 parameter의 값을 통계적 기법인 GCV를 사용하여 자동 산출한다. 따라서 사용자가 parameter 값을 조종할 필요가 없고, GCV B-spline 패키지를 그냥 사용하면 된다. 본 논문에서 GP에 의한 fitting도 비록 heuristic한 방법이지만 사용자가 regularization parameter 값을 조종하지 않고(GP 자체에 대한 parameter들은 제외하고) 사용하는 것을 목표

로 하고 있다. 둘째로는 GCV B-spline는 그 소스 코드가 공개되어 있다. GP를 이용한 fitting은 B-spline을 대체하고자 하는 것이 아니고, B-spline이 문제가 될 수 있는(위에서 언급한) 경우에 대한 하나의 대안으로 사용하고자 하는 것이다.

본 논문에서는 노이즈가 포함된 데이터의 Smooth fitting을 위하여 DBSF GP 기법을 제시하였다. 기존의 방법과는 달리 유전적 프로그래밍을 사용하여 주어진 데이터의 미분식을 먼저 탐색하고, 찾아진 최적의 미분식을 적분함으로써 Smooth fitting을 수행하는 방안이다. 즉 본 방법의 핵심은 바로 미분식을 나타내는 GP 트리의 Smoothness를 우선적으로 고려함으로써 F의 Smooth fitting을 보장 받는 것이라고 할 수 있다.

3절의 예를 통하여 DBSF GP 기법의 결과가 GCV B-spline의 결과보다 우수함을 보였고, 실제적인 응용을 위해서 DBSF GP기법을 산적화물선의 주요치수 추정에 적용하였다.

그러나 DBSF GP 기법의 단점은 다른 방법들과 비교하여 과도한 계산시간이 요구된다는 점이다. 또한 GP에서는 Regularization 파라메터 추정에 사용되는 방법들(GCV[1-3], L-curve method[13,14], Zero-crossing method[15], Composite residual and smoothing operator[16])을 적용하기가 매우 어렵기 때문에 주로 본 논문과 같이 이론적 기반보다는 Heuristic한 방법에 의존할 수 밖에는 없다는 분제점이 있다.

감사의 글

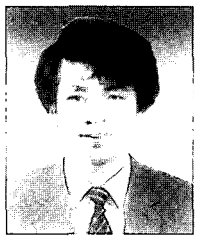
본 논문은 한국과학재단의 해외 Post-doc. 연구지원에 의해 연구된 결과의 일부임을 밝혀둔다.

참고문헌

1. Wabba, G., "How to smooth curves and surfaces with splines and cross validation," 24th Conf. On the Design Experiments, US Army Research Office, 1997.
2. Craven, P. and Wahba, G., "Smoothing noisy data with spline functions," *Numerische Mathematik*, Vol. 31, pp. 377-403, 1979.
3. Woltring, H., "A fortran package for generalized cross-validation spline smoothing and differentiation", *Adv. Eng. Soft*, Vol. 8, pp. 104-113, 1986.
4. Thomas, J. B. and Richard, L. L., "Stepwise regression is an alternative to splines for fitting noisy data," *J. Biomechanics*, Vol. 29, No. 2, pp. 235-238, 1996.
5. Trujillo, D. M. and Busby, H. R., "Optimal regularization of the inverse heat-conduction problem," *AIAA J. Thermophys. Heat Transfer*, Vol. 3, No. 2, pp. 423-427, 1989.
6. Busby, H. R. and Trujillo, D. M., "Optimal regularization of an inverse dynamics problem," *Computers & Structures*, Vol. 63, No. 2, pp. 243-248, 1997.
7. Koza, J. R., Genetic programming: On the Programming of Computers by Means of Natural Selection, The MIT Press, 1992.
8. Giannis, G. and Vasilios, B., "Optimal digital filtering requires a different cut-off frequency strategy for the determination of the higher derivatives," *J. Biomechanics*, Vol. 30, No. 8, pp. 851-855, 1997.
9. Giannis, G. and Vasilios, B., "A comparison of automatic filtering techniques applied to biomechanic walking data," *J. Biomechanics*, Vol. 30, No. 8, pp. 847-850, 1997.
10. Yeun, Y. S., Lee, K. H. and Yang, Y. S., "Function approximation by coupling neural networks and genetic programming trees with oblique decision trees," *AI in Engineering*, Vol. 13, No. 3, 1999.
11. Yeun, Y. S., Suh, J. C. and Yang, Y. S., "Function approximation by superimposing genetic programming trees: with applications to engineering problems," *Information Sciences*, Vol. 122, Issue 2-4, 2000.
12. 연윤석, "가중 선형 연상기억을 채용한 유전적 프로그래밍과 그 공학적 응용," 한국CAD/CAM학회논문집, Vol. 3, No. 1, pp. 57-67, 1998.
13. Hansen, P. C., "Analysis of discrete Ill-Posed problems by means of the L-Curve," *SIAM Rev.*, Vol. 34, pp. 561-580, 1992.
14. Hansen P. C. and O'Leary, D. P., "The use of the L-Curve in the regularization of discrete Ill-Posed problems," *SIAM J. Sci. Stat. Comput.*, Vol. 14, pp. 1487-1503, 1993.
15. Peter, R. J. and Ramesh, M. G., "A new method for regularization parameter determination in the inverse problem of electrocardiography", *IEEE Trans. Biomedical Engineering*, Vol. 44, No. 1, pp. 19-39, 1997.
16. Colli-Franzone, P., Guerri, L., Taccardi, B. and Viganotti, C., "Finite element approximation of regularized solutions of the inverse potential problem of electrocardiography and applications to experimental data," *Calcolo*, Vol. XXII, No. 1, 1985.
17. Yeun, Y. S., Lee, K. H., Han, S. M. and Yang, Y. S., "Smooth fitting with a method for determining the regularization parameter under the genetic programming algorithm," *Information Sciences*, Vol. 133, pp. 175-194, 2001.
18. http://www.netlib.org/gcv/
19. http://octave.sourceforge.net/index/f/gcvspl.html

이 경 호

1988년 서울대학교 조선해양공학과 학사
1990년 서울대학교 조선해양공학과 석사
1998년 서울대학교 조선해양공학과 박사
1990~2003년 한국해양연구원 연구원
2002~2003년 Univ. of Maryland Visiting Researcher
2003년~현재 안하대학교 기계공학부 조교수
관심분야: Artificial Intelligence in Design, Concurrent Engineering, SBD



연 윤 석

1989년 서울대학교 조선해양공학과 학사
1991년 서울대학교 조선해양공학과 석사
1995년 서울대학교 조선해양공학과 박사
1993년~현재 대진대학교 컴퓨터응용 기계설계공학과 부교수
관심분야: 지능형설계시스템, Evolutionary Computations for Engineering Design, Artificial Intelligence In Engineering