

## 예측필터를 이용한 소프트웨어 개발 인력분포 예측

# A Prediction for Manpower Profile of Software Development Using Predictive Filter

이 상 윤\*  
Sang-Un Lee\*

국립 원주대학

### 요 약

소프트웨어 개발 인력 프로파일에 대한 현존하는 모든 통계적 모델들은 소프트웨어 사용과 개발 프로세스의 가정에 기반을 두고 있어 일반적으로 적용 가능한 추정과 예측 모델이 없는 실정이다. 본 논문은 예측필터를 적용하여 소프트웨어 개발 투입 인력 프로파일을 예측하였다. 먼저 소프트웨어 개발 인력분포를 살펴보고, 예측필터를 적용하기 위해 모델의 입력-출력, 모수를 결정하는 방법을 제시하였다. 이어서 제안된 모델의 유용성은 실제 개발된 소프트웨어 프로젝트로부터 획득된 데이터 분석으로 경험적으로 검증되었다. 평균 상대오차와 Pred(0.25)에 기반하여 제안된 예측필터는 잘 알려진 통계적 추정 모델들과 비교되었다. 검증 결과 예측필터는 단순한 구조를 갖고 있으면서도 소프트웨어 인력분포를 적절히 표현하는 결과를 보였다.

### Abstract

Most of the existing statistical models of software manpower profile are based on the assumptions of the usage and development process. Therefore, there is no universally applicable estimation and prediction model. To develop a prediction model, this paper suggests the predictive filter as a prediction model for software manpower profile. First of all, we investigate the software manpower profile and we suggest the input-output of predictive filter and method for parameter determination. Then, its usefulness is empirically verified by analyzing the actual data obtained from the software projects. Based on the average relative prediction error and Pred(0.25), the suggested predictive filter is compared with other well-known statistical estimation models. As a result, the predictive filter generally has a simple structure and on the other hand, it adapts the software manpower profile very well.

**Key words** : Software Life Cycle, Life Cycle Manpower, Rayleigh Model, Gamma Model, Predictive Filter

## 1. 서 론

소프트웨어 프로젝트 개발에 소요되는 인력 (Work Effort 또는 Manpower) 측정은 프로젝트 관리자에게는 필수적으로 요구되는 능력이다. 소프트웨어 개발 인력 측정은 추정과 예측활동으로 구분된다. 프로젝트 계획단계에서는 개발에 투입될 총 인력의 규모를 추정하는데 초점을 두고 있다. 반면에 실제 소프트웨어를 개발하는 과정에서는 주어진 한정된 자원을 효율적으로 사용하기 위해 어느 시점에서 얼마만큼의 인력으로 배분할 것인가를 예측하여 프로젝트를 성공적으로 완수하는데 목표를 두고 있다. 소프트웨어 개발 인력 예측 모델은 개발에 투입된 실제 인력에 대한 현재까지 수집된 데이터에 기반하여 미래의 투입 인력을 예측하기 위한 능력을 갖고 있어야만 한다. 이것은 현재까지 투입된 인력이 어떤 분포를 따르므로 추후에도 이 분포를 따를 것이라고 예측하는 방법과 현재까지 투입된 인력 데이터를 이용해 미래의 투입 인력을 예측하는 방법이 있다. 전자의 방법은 대부분 모수적 방법인 통계적 모델을 이용하는 경우이며, 후자는 통계적 모델보다는 비모수적 방법인 신경망 (Neural Network)과 퍼지

(Fuzzy) 등 비통계적 모델을 이용하는 것이 보다 정확한 값을 유도할 수 있다[1].

과거의 개발 경험 데이터들을 기반으로 소프트웨어 개발인력을 추정하기 위한 많은 모델들이 개발되었다. 대표적인 모델로는 Putnam[2]의 SLIM (Software Life-cycle Management) 모델인 Rayleigh 모델과 Phillai et al.[3]의 Gamma 모델이 있다. 일반적으로 통계적 모델들은 소프트웨어 개발에 투입되는 인력이 통계적으로 볼 때 어떠한 분포를 따른다는 가정에 기반을 두고 있다. 또한 소프트웨어 개발과 관련된 다양한 가정을 반영하는 모수들을 포함하고 있다. 따라서 통계적 모델 적용시 개발될 소프트웨어 프로젝트에 투입될 인력분포 형태를 결정한 후 유사한 특성을 가진 소프트웨어 프로젝트들로부터 유도된 모델을 선택하는 것이 모델의 모수 추정에 특히 중요하다.

새로 개발되는 소프트웨어는 모두 유일한 특성을 갖고 있다[1]. 즉, 기존에 개발되어 사용 중인 소프트웨어와 동일한 소프트웨어를 개발하는 경우는 없을 것이다. 이로 인해 유일한 속성들을 가진 소프트웨어를 개발하는데 투입되는 개발인력 분포를 사전에 인지하기란 사실상 불가능하므로 통계적 모델을 적용하는데 어려움이 있다. 모든 환경에 적용될 수 있는 보편적인 통계적 모델 (Universally Applicable Model)이 없는 실정이다. 따라서 개발 환경과 관련된 어떠한 가정

접수일자 : 2006년 4월 5일  
완료일자 : 2006년 8월 14일

도 요구하지 않는 모델 개발이 절실히 요구된다. 만약 소프트웨어 시스템의 과거 개발에 투입된 인력의 이력 데이터에 기반하여 모델을 구축할 수 있는 시스템을 갖고 있다면 통계적 모델들이 선택하는 이들 가정은 고려하지 않아도 된다. 이에 적합한 모델로서 신경망이 있다. 신경망은 단지 과거의 이력 데이터만을 입력으로 요구하며, 통계적 모델처럼 사전에 어떠한 가정도 요구하지 않는 특징을 갖고 있다[4].

다양한 신경망 중에서 대표적인 전향망 (FFN, Feed Forward Network)은 추정 성능은 뛰어나나 모수 (가중치, Weights)의 수가 너무 많은 구조를 갖고 있어 복잡한 모델이 될 수 있다. 또한 가중치의 초기값을 랜덤하게 설정하는 특성으로 인해 동일한 프로젝트에 대해서도 훈련을 수행할 때마다 다른 결과를 나타내어 특정한 모델을 유도하기가 불가능하다[5]. 따라서 본 논문은 “단순하면서도 성능이 우수하고, 널리 사용되고 있는” 좋은 모델이 되기 위한 조건을 만족시키기 위해 예측필터 (Predictive Filter)를 적용하여 소프트웨어 개발인력의 프로파일을 예측하고자 한다.

2장에서는 기존의 개발인력 추정과 관련된 모델들과 문제점을 살펴본다. 3장에서는 소프트웨어 개발인력을 예측하는 분야에 예측필터를 적용하기 위해 모델의 입력-출력, 모수를 결정하는 방법을 논하고 모델을 구성한다. 이어서, 4장에서는 제안된 모델을 실제 데이터를 이용하여 적용해 보고, 기존 제안된 모델들과 성능을 평가하여 모델의 적합성을 제시한다.

## 2. 기존 연구 및 문제점

소프트웨어 개발인력 분포를 살펴보면, 단위 시간당 일정한 인력을 투입하는 일양분포 (Uniform Distribution) 형태를 취하는 경우는 거의 없으며, 투입 인력이 상승하다가 절정에 도달한 후 하강하는 분포를 따르는 것이 일반적인 현상이다[6].

Norden[7]은 IBM사에서 개발된 다양한 하드웨어 개발과정에 투입된 인력은 근사적으로 Rayleigh 분포를 따른다는 사실을 관찰하였다. 이 관찰 결과가 이후 Putnam[2]에 의해 소프트웨어 프로젝트에 적용되었다. 소프트웨어 생명주기 (Life Cycle) 전반 (계획, 요구명세 분석, 설계, 코딩, 시험과 유지보수 중 계획과 요구사항 분석단계 제외)에 걸쳐 소요되는 총 인력뿐만 아니라 각 단계에 투입되는 인력도 그림 1의 Rayleigh 분포를 따름을 관찰하였다.

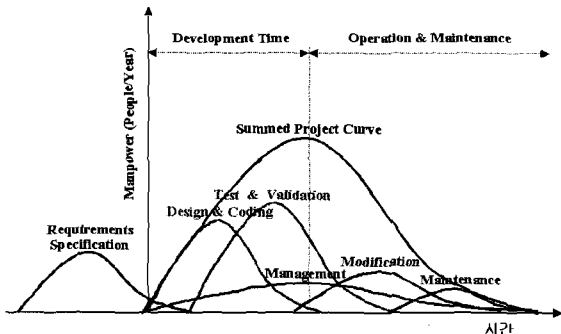


그림 1. 소프트웨어 개발인력 분포

Fig. 1. A Manpower Distribution of Software Development

임의의 시간  $t$  시점까지 프로젝트에 투입된 누적 인력  $y$ 를 식 (1)로, 임의의 시간  $t$  시점에서 프로젝트에 투입되는 인력  $\dot{y}$ 는 식 (2)로 표현된다.

$$y = E(1 - e^{-at^2}) MY \quad (1)$$

$$\dot{y} = 2Eate^{-at^2} = \frac{E}{t_d^2} te^{-\frac{t}{2t_d}}, a = \frac{1}{2t_d^2} MY/YR \quad (2)$$

여기서  $E$ 는 Rayleigh 곡선 아래에 있는 영역으로 년 인원으로 표시되며, 소프트웨어 생명주기 전반에 걸쳐 투입된 총 인력이다.  $t_d$ 는  $\dot{y}$ 이 최대가 되는 시점 ( $t_{y_{max}} = t_d$ )으로 소프트웨어 생명주기 전반에 걸쳐 총 소요되는 개발노력의 약 40%에 도달하는 개발 종료 시점이다.

이 모델에서 기울기  $\ln(\frac{\dot{y}}{t})$ 가 음수가 되는 개발단계에서는 모수 추정이 불가능한 단점을 보완하기 위해 Phillai et al.[3]은  $\ln(\frac{\dot{y}}{t})$  대신  $\ln(\frac{\dot{y}}{t^2}) = \gamma + \mu t$  변환을 통해

$$y = e^{\gamma t^2} e^{-\mu t} = \frac{1}{\Gamma(a)\beta^a} t^{a-1} e^{-\frac{t}{\beta}}, a=3, \beta = \frac{t_d}{2} \text{로 치환하여 식 (3)의 Gamma 분포 모델을 제시하였다.}$$

$$\dot{y} = \frac{8E}{\Gamma(3)t_d^3} (t^2 e^{-\frac{2}{t_d}t}) = \frac{4E}{t_d^3} t^2 e^{-\frac{2}{t_d}t} \quad (3)$$

두 모델이 공통적으로  $t_d$  (개발 완료 시점에서 투입인력이 절정에 도달)를 모수로 사용하고 있다. 이는 개발이 종료되는 시점에서 인력이 최대로 소요되며, 개발이 진행되는 동안 인력을 계속적으로 증가시켜야 함을 의미한다.

Phillai et al.[3]이 Gamma 모델을 제안하면서 적용한 데이터는 Warburton[8]이 제시한 그림 2 데이터이다.

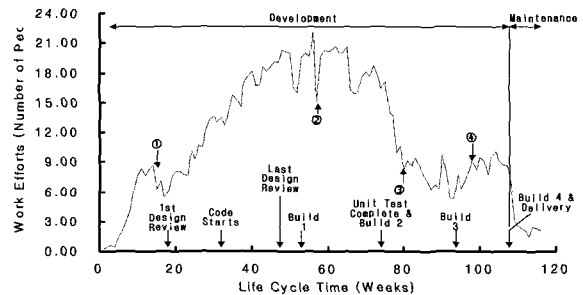


그림 2. 소프트웨어 생명주기 투입인력 사례

Fig. 2. Example of Software Life cycle Manpower Distribution

이 데이터는 소나와 화력제어계통 (Sonar and Fire Control System)의 훈련/시뮬레이터 프로젝트로 요구사항 명세화, 설계, 코딩, 시험, 시스템 통합과 납품을 포함한 개발 전 과정에 투입된 인력분포이다. 이 시스템은 116주의 기간 동안 1300.8명 (Man/Year)이 투입되었으며, 7일 이상의 MTTF (Mean Time To Failure) 품질수준이 달성되었을 때 개발이 종료되고 고객에게 납품 (Delivery) 되었다.

그림 2에서 투입 인력이 절정에 도달하는 시점을 살펴보면 Putnam[2]이 제시한 전형적인 소프트웨어 개발 인력 분포인 그림 1은 개발이 종료된 시점인 반면 그림 2의 실제 데이터에서는 설계 종료 시점에서 발생하고 있다. 따라서  $t_d$ 가

개발 종료시점 ( $t_d=106$  Weeks)이 아니라 개발 과정 중에서 인력이 최대로 소요되는 시점(대략  $t_d=52$  Weeks)이라고 가정할 수 있다.

생명주기의 초기에 개발에 소요되는 인력 추정시 개발 종료 시점인  $t_d$ 값은 알 수 있지만 실제로 개발 중 최대의 인력이 투입되는 시점은 알지 못한다. 이에 따라 다음의 가정에 근거하여 저자는 시그모이드 모델을 연구하였다.

- 가정 1. 투입인력이 절정에 도달하는 시점은 개발 완료 시점이 아니며, 개발 과정 중에 발생한다.
- 가정 2. 설계 시작 시점부터 개발인력을 추정하는 모델이 아니라 계획과 요구사항 분석 단계도 포함하여 소프트웨어 생명주기 전체가 아닌 개발주기에 한정된 인력 분포를 표현해야 한다.
- 가정 3. 계획과 요구사항 분석단계 까지 고려함으로써, 시작 시점에서 투입 인력이 0에 가까운 값이 아니고 어느 정도의 절편 값을 갖는다.

위에 제시된 가정들을 모두 만족시킬 수 있는 함수로 식 (4)의 시그모이드 함수 (Sigmoid Function)[9]를 적용할 수 있다. 시그모이드 함수는 그리스 문자인 Sigma 글자와 같은 형태를 갖고 있어 불리워진 함수로 신경망에서 뉴런의 입력과 출력간의 비선형성 정도를 표현하는데 사용되고 있다[10].

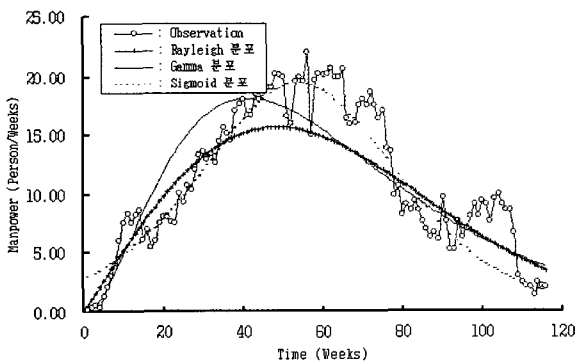
$$y = \frac{1}{1 + e^{-at}}, \quad \dot{y} = \frac{e^{-at}}{(1 + e^{-at})^2} \quad (4)$$

식 (4)의 형태를 취하는 식 (5)의 모델을 적용하여 시간에 따른 투입인력을 추정하였다. 여기서  $a$ 와  $b$ 는 상수이다.

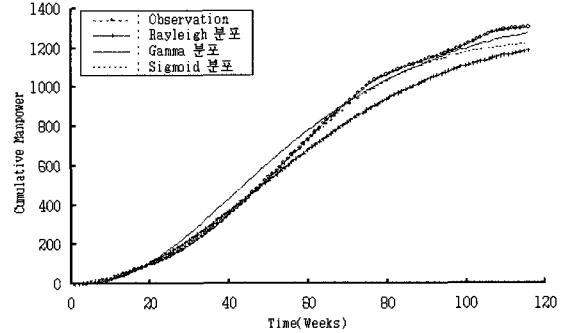
$$y = \frac{Ke^{-a}}{(1 + be^{-a})^2}, \quad a = \frac{t-a}{b}, \quad a > 0, b > 0 \quad (5)$$

그림 2 데이터에 대해, 식 (5)의 시그모이드 분포 모델의 모수 추정 결과  $\hat{a} = 7.014$ ,  $\hat{b} = 16.711$ 을 얻었다. 또한  $t_d$  모수에 대해 Rayleigh 분포 모델은  $t_d = 48.438$ 로 추정되었다. 이들 3개 모델을 실제 데이터에 적용하여 투입인력을 적합시킨 결과는 그림 3에 제시되어 있다.

Gamma 분포 모델이 이론적으로 Rayleigh 분포 모델보다 타당하나 실제로 인력이 절정에 도달하는 시점이 보다 좌측으로 이동되어 나쁜 결과를 얻었으며, 또한 두 모델 모두 실제의  $t_d$  (52 Weeks) 시점을 적절히 표현하지 못하는 단점을 갖고 있다. 이에 비해 시그모이드 분포 모델은 인력이 절정에 도달하는 시점을 적절히 정확히 추정하였다.



(a) 단위시간에 따른 개발인력 추정



(b) 단위시간에 따른 누적 개발인력 추정

그림 3. 투입인력 절정 시점 ( $t_d=52$  Weeks) 고려시 인력 추정  
Fig. 3. Manpower Estimation in accordance with Manpower Peak Time

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로 모델의 성능을 평가하는 것이 필요하다. 소프트웨어 공학 분야에서는 결정계수 (Coefficient of Determination,  $R^2$ )[11], MMRE와 Pred(0.25)[12]를 일반적으로 적용하고 있다. 결정계수는 주어진 모델이 실제 데이터를 얼마나 잘 설명할 수 있는가를 평가한다.  $R^2$  ( $0 \leq R^2 \leq 1$ )이 0에 가까우면 추정된 회귀직선은 쓸모가 없으며, 값이 클수록 쓸모 있는 회귀직선이 된다. 상대오차 (Relative Error, RE)는  $RE = \frac{(\text{실측치} - \text{추정치})}{\text{실측치}} \times 100(\%)$ 로 구해지며, MRE (Magnitude of the RE)는  $|RE|$ 이며,  $n$ 개의 데이터에 대한 MMRE (Mean MRE)는  $\frac{1}{n} \sum_{i=1}^n MRE_i$ ,  $i=1, 2, \dots, n$ 로 계산된다. MMRE가 작은 값이면 좋은 모델임을 알 수 있다 [13,14]. Pred(0.25)는 주어진 모델로 추정 또는 예측한 결과 값이 실제 값의 25% 이내에 속하는 비율이다. Conte et al.[13]은 MMRE가 25% 이하인 경우와 Pred(0.25)가 0.75(75%) 이상인 경우 뛰어난 예측 모델로 선정될 수 있음을 제시하였다.

$R^2$ , MMRE와 Pred(0.25) 기준을 적용한 모델의 성능은 표 1에 제시하였다. 세 가지 모델 모두 특정한 하나의 평가 기준에서는 다른 모델들 보다 성능이 우위를 점하고 있다. 그러나 Conte et al.[13]이 제시한 좋은 예측모델이 되기 위한 조건인 MMRE, Pred(0.25) 측면에서는 모두 좋은 모델로서의 성능을 갖고 있지 못함을 알 수 있다.

표 1. 모델 성능 비교

Table 1. Model Performance Comparison Table

모델	$t_d=52$ 인 경우		
	$R^2$	MMRE	Pred(0.25)
Putnam의 Rayleigh 모델	0.8344	33.94%	92.24%
Phillai et al.의 Gamma 모델	0.7360	30.62%	76.72%
시그모이드 모델	0.8733	47.37%	90.52%

따라서 3장에서는 “단순하면서도 모델의 성능이 우수하고 널리 사용되고 있는” 좋은 모델이 되기 위한 조건을 만족시키면서 소프트웨어 개발 투입인력의 프로파일을 적절히 예측할 수 있는 모델을 제시한다.

### 3. 예측필터를 이용한 개발인력 프로파일 예측

#### 3.1 적응필터와 예측필터

적응필터 (Adaptive Filter)는 ADALINE (Adaptive Linear Neuron Network)에 TDL (Time Delay Line)이 결합된 형태를 취하고 있다[5]. ADALINE은 퍼셉트론과 유사하지만 전달함수 (또는 작동함수)로 선형을 사용한다. 이 망은 단지 선형적으로 분리할 수 있는 문제를 해결할 때만 적용할 수 있다. 비선형 형태를 취하는 문제를 해결하고자 할 때, 새로운 컴포넌트인 TDL을 이용하여 ADALINE 망의 활용을 넓힐 수 있다. 적응필터는 환경변화에 자동적으로 적응하는 능력을 갖고 있다. 그림 4는 적응필터의 단순한 구조를 보여주고 있다. 여기서  $x_i$ 는  $i$ 번째 시간의 입력 신호이며,

$\hat{y}_i = \sum_{k=0}^d w_k x_{i-k} + b$ 는 이와 관련된 출력이다.  $d$ 는 지연의 수,  $w_k$ 는 적응 가중치 (Weight),  $b$ 는 바이어스 (bias or threshold)이다.

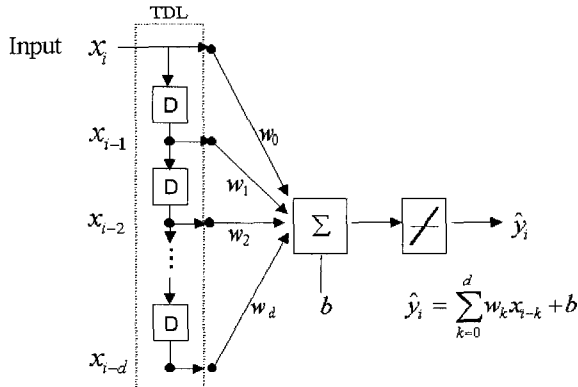


그림 4. 적응필터  
Fig. 4. Adaptive Filter

적응필터는 가중치가 적용된  $x_i, x_{i-1}, \dots, x_{i-d}$ 의 부분적인 이전 데이터를 받아들여 선형 결합시켜  $i$ 번째 출력  $\hat{y}_i$ 을 얻는다. 여기서  $i$ 번째 입력  $x_i$ 에 연결된 가중치  $w_0$  선을 제거하고  $i$ 번째 입력인  $x_i$ 를 원하는 출력  $y_i$ 로 연결하면 그림 5와 같이  $x_{i-1}, x_{i-2}, \dots, x_{i-d}$ 의 이전 데이터를 받아들여 다음 단계인  $i$ 번째 데이터  $x_i$ 를 얻을 수 있다. 이를 다음 단계 예측 (Next-step Prediction)이라 하며, 이 망을 예측 필터 (Predictive Filter)라 한다. 이 망은 디지털 신호처리 분야에서는 일반적으로 FIR (Finite Impulse Response) 필터 (또는 입력지연 신경망, Input-delay Neural Network)으로 널리 알려져 있다. 이 구조는 음성인식과 시계열 예측을 포함한 다수의 실제 응용분야에서 성공적으로 적용되고 있다 [15]. 예측필터를 이용하면 몇 단계 이후 시점까지도 예측이 가능한 모델을 설계할 수 있다. 그러나 본 논문은 다음 단계 투입 인력을 예측하는데 한정시킨다. 여기서 단위 시간 간격  $t_i - t_{i-1}$ 은 일정하다고 가정한다[16].

#### 3.2 예측필터 모델 설계 및 임·출력 선정

소프트웨어 개발 투입 인력 분포를 유도하는 통계적 모델들은 독립변수인 개발 단위시간  $t_i (i=1, 2, \dots)$ 에 투입되는 인력 (종속변수)을 추정한다. 즉 모델의 독립변수인 입력으

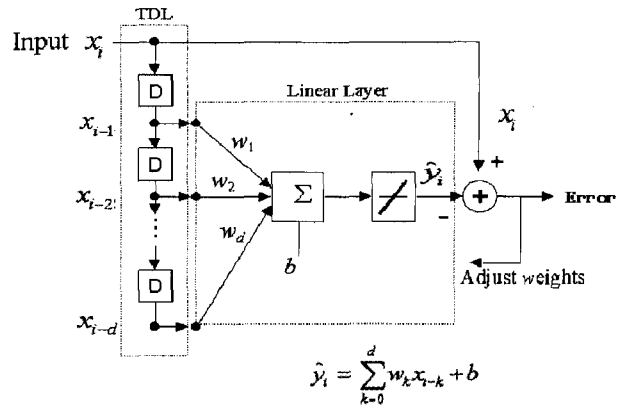
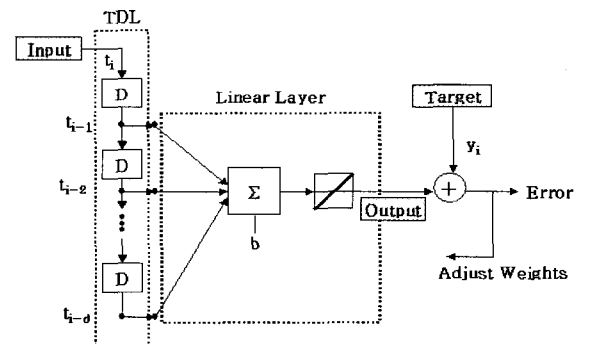


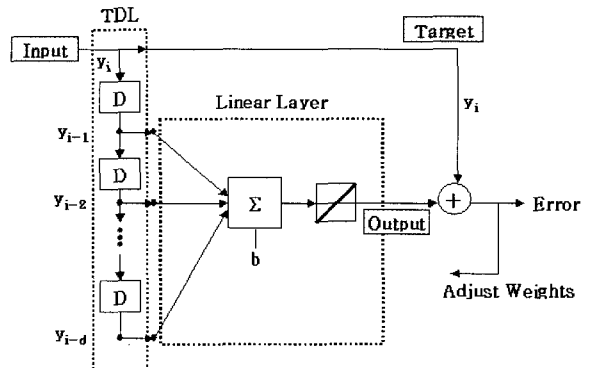
그림 5. 예측필터  
Fig. 5. Predictive Filter

로 단위시간을, 종속변수인 출력으로 투입인력을 표현한다. 본 논문에서는 투입 인력 데이터를 임의의 단위 시간  $t_i$  시점에서 투입된 인력  $\hat{y}_i$ 이 아닌 임의의 단위시간  $t_i$  시점까지 투입된 누적 투입인력 (Cumulative Number of Manpower)  $y_i$ 을 이용한다.

예측필터에서는 종속변수인 소프트웨어 개발 누적 인력  $E_i$  분포를 추정하기 위해 입력으로 개발 단위시간  $t_i$  또는 누적 투입인력  $y_i$  데이터를 사용할 수 있다. 입력으로  $t_i$ 를 선정하였을 경우와 이전의 누적 개발인력  $y_i$  데이터를 선정하는 모델이 그림 6에 제시되어 있다.



(a) 단위시간을 입력으로 하는 인력 추정과 예측



(b) 누적 투입인력을 입력으로 하는 예측 모델 설계  
그림 6. 소프트웨어 개발인력 프로파일 예측모델 설계  
Fig. 6. A Predictive Model Design of Software Manpower Profile

모델의 출력을 어떻게 선정하느냐에 따라 추정 모델이 되느냐 아니면 예측 모델이 되느냐를 결정한다. 모델의 출력은 우리가 추정 또는 예측하고자 하는 개발에 투입되는 인력이 되며,  $t_i$  시점에서 투입된 인력을 추정하는 데이터는  $y_i$ 로 선정시 추정문제를 해결하는 모델이 되며,  $t_{i-1}$  시점까지의 데이터를 이용해 다음 시점인  $t_i$  시점에서 투입될 누적 인력  $y_i$ 를 출력으로 하는 모델은 예측모델이 된다.

만약 원하는 신호인 목표 (Target)  $y_i$ 에 망의 출력인  $\hat{y}_i$ 를 가능한 일치시키기 위해서는 망의 가중치를 조절하는 방법을 가진다면 적응필터는 미래의 신호 값을 예측할 수 있다.

### 3.3 모델 구현 및 TDL 수 결정

통계적 추정 모델들은 수집된 전체 데이터를 이용하여 모수를 추정한다. 이에 비해 예측필터는 모든 데이터를 입력으로 하지 않으며, 단지 이전의 몇 개 데이터만을 사용한다는

```

predvalin.m
format short;
WF_E = [0.2 0.6 1.2 1.6 3 5.1 8.2 12.3 18.3 25.9 ...
        34.3 41.9 50.2 58.9 65.1 72.1 77.6 83.6 91.2 99.3 ...
        107.3 115.0 122.6 132.7 142.0 152.7 163.1 175.2 188.5 202.1 ...
        215.1 228.6 241.3 255.8 271.5 286.7 301.3 318.4 336.1 354.2 ...
        370.9 387.6 406.1 424.2 442.7 461.7 480.7 501.0 521.1 541.1 ...
        557.7 573.7 593.3 613.3 632.9 654.9 670.0 689.7 710.0 730.1 ...
        750.4 771.1 791.1 811.1 831.7 848.1 864.0 880.1 897.7 915.8 ...
        933.4 952.1 969.7 986.1 1003.2 1017.2 1030.8 1040.7 1051.3 1059.6 ...
        1068.7 1077.3 1086.7 1095.3 1103.0 1110.0 1116.3 1123.0 1129.2 1138.9 ...
        1146.6 1151.9 1157.2 1164.9 1171.2 1178.3 1186.4 1195.5 1203.7 1213.1 ...
        1222.2 1229.9 1239.5 1249.5 1258.6 1267.3 1275.9 1282.6 1285.7 1288.2 ...
        1290.4 1292.5 1294.0 1296.5 1298.7 1300.8];

DataCount = 1;
Data1PredError = []; Data2PredError = []; Data3PredError = [];
for DataCount = 1 : 3
    Error = [];
    TimeDelay = 0;
    TrainingSetSize = 0;
    if DataCount == 1
        Tdata = WF_E;
    end
    for TrainingSetSize = 3 : (length(Tdata) - 1)
        for TimeDelay = 1 : (TrainingSetSize - 1)
            error = PredViaLinearNN(Tdata, TrainingSetSize, TimeDelay);
            if DataCount == 1
                Data1PredError(TimeDelay, (TrainingSetSize-2)) = error;
            elseif DataCount == 2
                Data2PredError(TimeDelay, (TrainingSetSize-2)) = error;
            elseif DataCount == 3
                Data3PredError(TimeDelay, (TrainingSetSize-2)) = error;
            end
        end
    end
    if DataCount == 1
        save WFE Data1PredError -ascii;
    elseif DataCount == 2
        save UPBD Data2PredError -ascii;
    elseif DataCount == 3
        save UPRD Data3PredError -ascii;
    end
end

PredViaLinearNN.m
function error = PredViaLinearNN(Tdata, TrainingSetSize, TimeDelay)
TrainingSet = Tdata(1 : TrainingSetSize + 1);
Columns = length(TrainingSet);
P = [ ]; T = [ ];
for j = 1 : (Columns - TimeDelay)
    P(:, j) = TrainingSet(j + TimeDelay) : -1 : j;
    T = [T TrainingSet(j + TimeDelay)];
end
net = newlind(P, 1 : (Columns - TimeDelay - 1), T(:, 1 : (Columns - TimeDelay - 1)));
a = sim(net, P(:, Columns - TimeDelay));
error = T(:, Columns - TimeDelay) - a;
    
```

그림 7. 예측필터를 이용한 MatLab 인력 예측 프로그램  
Fig. 7. Matlab's Manpower Predictive Program Using Predictive Filter

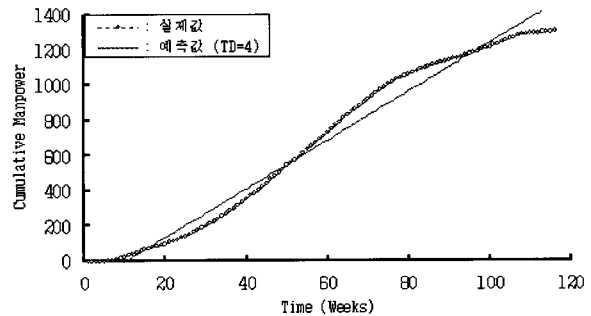
특징을 갖고 있다. 따라서 예측필터를 적용하려면 이전의 몇 개의 누적 인력 데이터 ( $y_{i-1}, y_{i-2}, \dots$ )를 사용하는 것이 가장 좋은지를 결정해야 한다. 이것이 TDL의 수를 결정하는 문제이다. 이 문제를 해결하는 최적의 방법은 주어지는 데이터의 특성에 따라 변화함으로써 이론상으로 제시될 수가 없으

며, 단지 주어진 데이터에 따라 망이 최적으로 훈련됨에 따라 결정될 수 있다. 즉 주어진 데이터에 따라 시행착오법 (Trial-and-error)으로 결정된다. 본 논문에서는 MatLab NN Toolbox의 예측필터를 이용하여 그림 7과 같이 프로그램을 구현하여 실험을 하였으며, TDL은 1부터 20까지 변화시키면서 MMRE가 최소가 되는 TDL을 선정하였다. 결론적으로 예측필터로 다음단계 인력을 예측하는 망을 훈련시킨 결과 단위시간을 입력으로 하는 경우는 TDL의 개수는 4를, 누적 투입인력을 입력으로 하는 경우의 TDL의 개수는 3을 얻었다. 즉, 이전의 3개 인력 데이터만을 이용하면 다음 단계의 인력을 추정할 수 있다는 것을 의미한다. 4장에서는 적절한 TDL의 개수를 얻기 위해 다양한 TDL의 수에 따른 분석을 수행하였다.

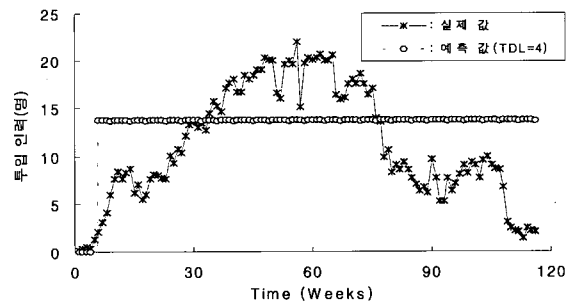
## 4. 적용 결과 및 분석

### 4.1 단위시간을 입력으로 하는 예측필터 적용

그림 6의 (a) 모델인 단위시간을 입력으로 하여 누적인력을 예측하기 위해 그림 2의 Warbuton[8] 데이터에 적용한 결과 실측값과 추정 값 비교와 상대오차를 그림 8에 제시하였다. 이 경우, TDL은 4개로  $n$ 번째 시점의 누적 인력  $y_n$ 을 예측하기 위해서는 이전 4개 시점인  $t_{n-1}, t_{n-2}, t_{n-3}, t_{n-4}$  값을 입력으로 받아들여 예측필터의 가중치를 조절하여 결과를 얻는다.



(a) 누적 인력분포 (실제 값과 예측 값)



(b) 단위 인력분포 (실제 값과 예측 값)

그림 8. 단위시간을 입력으로 하는 예측필터 모델  
Fig. 8. Predictive Filter Model in accordance with Unit Time Input

단위시간을 입력으로 하는 예측 모델의 경우 실제의 누적 투입 인력 곡선인 S자형 곡선을 표현하지 못하고 거의 선형 형태를 취하고 있다. 또한, 이를 단위시간에 따른 투입인력 그래프로 표현하면 그림 9와 같으며, 거의 모든 시점에서 동

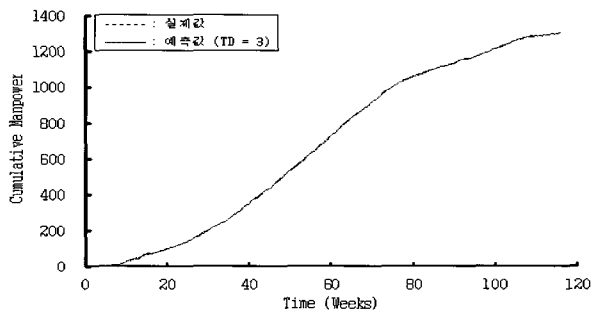
일한 인력 값을 나타내어 단위시간당 투입인력을 적절히 표현하지 못하여 좋은 예측모델로 선정하는데 실패하였다.

4.2 누적 투입 인력을 입력으로 하는 예측모델 적용

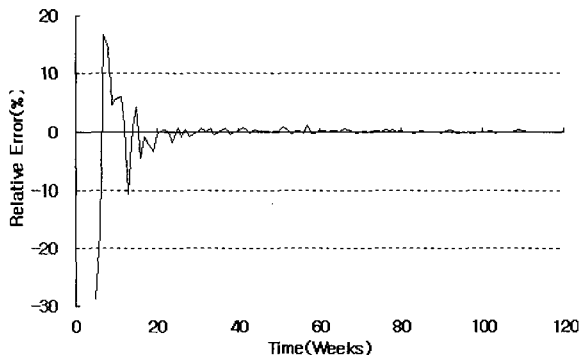
그림 6의 (b) 모델인 이전 시점부터 현재 시점까지의 누적 인력을 입력으로 다음 시점의 누적인력을 예측하기 위해 그림 7의 프로그램으로 그림 2 데이터에 적용한 결과는 그림 9에 제시하였다. 이는 최적의 값으로 TDL은 3을 얻은 경우이다. (a)는 시간에 따른 누적 인력 분포의 실제 값과 추정 값이며, (b)는 3번째 시점부터 예측한 결과의 상대오차이다. 모델의 성능은 표 2에 나타내었다.

본 모델은 그림 9의 (b)에서 알 수 있듯이 13번째 시점까지 약 10%의 상대오차를 나타내며, 실제 데이터에 적용하다가 이후부터는 거의 오차가 발생하지 않고 실제 데이터를 예측하고 있다. 또한, Conte et al.[13]의 MMRE와 Pred(0.25) 기준도 모두 만족시키며, 평균 상대오차는 1.27%, Pred(0.25)는 99.11%로 실제 데이터를 거의 정확히 예측함을 알 수 있다. 또한, 이를 단위시간에 따른 투입인력 그래프로 표현하면 그림 10과 같다.

그림 10을 그림 3의 (a)에서 보이고 있는 Rayleigh, Gamma, Sigmoid와 같은 통계적 모델들과 비교시 실제 인력 값을 비교적 잘 예측함을 알 수 있다. 지금까지 거론된 모든 소프트웨어 개발 인력 분포 추정 및 예측 모델에 대한 모델의 성능을 비교한 결과는 표 3에 제시하였다.



(a) 누적 인력분포 (실제 값과 예측 값)



(b) 3번째 시점부터 예측시의 상대오차

그림 9. 누적 투입 인력을 입력으로 하는 예측필터 모델  
Fig. 9. Predictive Filter Model in accordance with Cumulative Manpower Input

Rayleigh, Gamma, Sigmoid와 같은 통계적 모델들은 개발 각 시점에서 투입될 인력의 값들을 모두 알고 있다는 가정하에 이 값들에 적합한 함수식인 추정 성능면에서 볼 때 MMRE와 Pred(0.25)도 좋은 결과를 얻지 못하고 있다. 또한,

표 2. 누적 투입 인력을 입력으로 하는 예측필터 모델 성능  
Table 2. The performance of Predictive Filter Model in accordance with Cumulative Manpower Input

TDL의 수	모델 성능	
	MMRE	Pred(0.25)
3	1.27%	99.11%

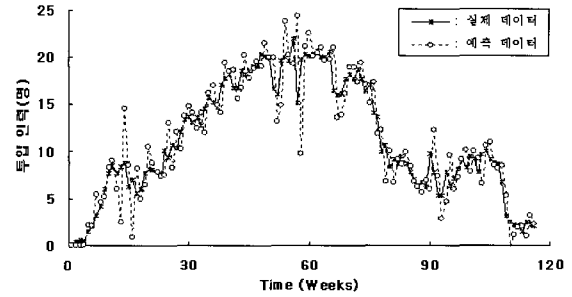


그림 10. 예측필터 모델의 단위시간당 인력 예측 값  
Fig. 10. Predictive Manpower per Unit Time of Predictive Filter Model

몇 개의 값만을 알고 있을 때, 이 값들이 해당 분포를 따르는 지도 알 수 없기 때문에 다음 시점에서의 투입 인력의 양을 예측하기도 어렵다. 반면에, 예측필터를 적용하면, 이전 시점의 몇 개의 데이터만 이용하면 다음 시점에서 투입해야 할 인력의 양을 거의 정확히 예측할 수 있는 장점을 갖고 있다. 따라서 실제 프로젝트 수행시 예측필터를 이용하여 각 시점에서 투입될 인력의 양을 보다 정확히 예측하여 적용하면 효율적인 프로젝트 관리를 수행할 수 있을 것이다.

표 3. 소프트웨어 개발 인력 분포 추정과 예측 모델 성능  
Table 3. The Model Performance of Estimation and Prediction for Development Manpower Distribution

모델	추정 성능		예측 성능	
	MMRE	Pred(0.25)	MMRE	Pred(0.25)
Rayleigh 분포 모델	33.94%	92.24%	-	-
Gamma 분포 모델	30.62%	76.72%	-	-
Sigmoid 분포 모델	47.37%	90.52%	-	-
누적 인력 입력 예측 필터 모델 (TDL = 3)	-	-	1.27%	99.11%

5. 결론 및 향후 연구과제

소프트웨어 개발 인력을 예측하는 단순하고 성능이 우수하며 널리 사용될 수 있는 모델을 개발하기 위해 본 논문은 예측필터를 제안하였다. 제안된 모델의 적합성을 검증하기 위해 실제 개발된 프로젝트의 인력 분포에 기반하여 다음 단계 투입 인력을 예측하는 예측력을 연구하였다. 모델의 평가에는 좋은 예측모델로서 갖추어야 할 MMRE와 Pred(0.25) 기준을 적용하였다. 이 평가 기준에 기반하여 제안된 예측필터 모델의 다음단계 투입 인력 예측력은 실제 데이터를 거의 정확히 예측하는 성능을 보였다.

보다 일반화된 예측모델을 유도하기 위해서는 보다 많은 인력분포 데이터가 요구된다. 또한, 예측필터의 단점은 시간 간격이 일정한 경우에만 적용할 수 있는 점이다. 그러나 소프트웨어 개발 시점 적용시 시간 간격이 일정하지 않을 경우도 발생할 수 있다. 이러한 경우에도 제안된 예측필터 원하는 성능을 발휘할 수 있는지를 검증할 필요가 있다. 따라서 추후 이 분야에 대한 연구가 수행될 것이다.

### 참 고 문 헌

[1] N. Karunanithi, D. Whitley and Y. K. Malaiya, "Using Neural Networks in Reliability Prediction," IEEE Software, Vol. 18, pp. 53-59, 1992.

[2] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. on Software Eng., Vol. SE-4, No. 4, 1978.

[3] K. Phillai and V. S. Sukumaran Nair, "A Model for Software Development Effort and Cost Estimation," IEEE Trans. on Software Eng., Vol. 23, No. 8, pp. 485-497, 1997.

[4] J. Y. Park, S. U. Lee, and J. H. Park, "Neural Network Modeling for Software Reliability Prediction from Failure Time Data," Journal of Electrical Eng. and Information Science, Vol. 4, No. 4, pp. 533-539, 1999.

[5] H. Demuth and M. Beale, "Neural Network Toolbox User's Guide, Ver 3.0," Math Works, Inc. 1997.

[6] P. Kruchten, "Planning an Iterative Project," The Rational Edge e-zine for the Rational Community, [http://www.rational-edge.com/contents/oct\\_02/f\\_iterativePlanning\\_pk.jsp](http://www.rational-edge.com/contents/oct_02/f_iterativePlanning_pk.jsp), 2002.

[7] P. V. Norden, "Curve Fitting for a Model of Applied Research and Development Scheduling," IBM J. Research and Development, Vol. 3, No. 2, pp. 232-248, 1958.

[8] R. D. H. Warburton, "Managing and Predicting the Costs of Real-Time Software," IEEE Trans. on Software Eng., Vol. SE-9, No. 5, pp. 562-569, 1983.

[9] M. T. Hargan, H. B. Demuth, and M. Beale, "Neural Network Design," PWS Publishing Company, 1996.

[10] <http://users.computerweekly.net/robmorton/projects/neural/sogmoid.html>

[11] 김우철 et al., "현대통계학," 영지출판사, 1994.

[12] 권기태 et al. "소프트웨어 유지보수 비용산정 모델 개선," 한국정보과학회 소프트웨어공학회지 제16권, 제2호, pp. 31-42, 2003.

[13] S. D. Conte, H. E. Dunsmore, and V. Y. Shen, "Software Engineering Metrics and Models," Menlo Park, CA: Benjamin Cummings, 1986.

[14] J. Verner and G. Tate, "A Software Size Model," IEEE Trans. on Software Eng., Vol. 18, No. 4, pp. 265-278, 1992.

[15] D. S. Clouse, C. L. Giles, B. G. Horne, and G. W. Cottrell, "Time-Delay Neural Networks: Representation and Induction of Finite State Machines," IEEE Trans. on Neural Networks, Vol. 8, No. 5, pp. 1065-1070, 1997.

[16] 박중양, 이상운, 박재홍, "예측필터를 이용한 소프트웨어 신뢰성 예측," 한국정보처리학회 논문지, 제7권, 제7호, pp. 2076-2085, 2000.

### 저 자 소 개



#### 이상운(Sang-Un Lee)

1983년 ~ 1987년 : 한국항공대학교 항공전자공학과 (학사)

1995년 ~ 1997년 : 경상대학교 컴퓨터학과 (석사)

1998년 ~ 2001년 : 경상대학교 컴퓨터학과 (박사)

1992년 ~ 2002년 : 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

2003년 : 강원도립대학 컴퓨터응용과 전임강사

2004년 ~ 현재 : 국립 원주대학 여성교양과 조교수

관심분야 : 소프트웨어 프로젝트 관리, 소프트웨어 개발 방법론, 소프트웨어 척도 (소프트웨어 규모, 개발 노력, 개발기간, 팀 규모), 분석과 설계 방법론, 소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성, 신경망, 뉴로-퍼지

e-mail : [sulee@sky.wonju.ac.kr](mailto:sulee@sky.wonju.ac.kr)