

# 가중 F 척도를 이용한 Trace-Driven 시물레이션 모델의 검증 방법

황보훈<sup>1</sup> · 천현재<sup>1</sup> · 이홍철<sup>1†</sup>

## Validation Technique of Trace-Driven Simulation Model Using Weighted F-measure

Hoon Hwangbo · Hyeonjae Cheon · Hong-chul Lee

### ABSTRACT

As most systems get more complicated, system analysis using simulation has been taken notice of. One of the core parts of simulation analysis is validation of a simulation model, and we can identify how well the simulation model represents the real system with this validation process. The difference between input data of two systems has an effect on the comparison between a simulation model and a real system at validation stage, and the result with such difference is not enough to ensure high credibility of the model. Accordingly, in this paper, we construct a model based on Trace-driven simulation which uses identical input data with the real system. On the other hand, to validate a model by each class, not by an unique statistic, we validate the model using a metric transformed from F-measure which estimates performance of a classifier in data mining field. Finally, this procedure enables precise validation process of a model, and it helps modification by offering feedback at the validation phase.

**Key words** : Validation technique, Trace-driven simulation, Time-series data, F-measure

### 요약

최근 시스템들이 복잡해지면서 시물레이션을 통한 시스템의 분석이 주목을 받고 있다. 시물레이션 분석에서 가장 핵심적인 부분 중의 하나가 시물레이션 모델의 검증이며, 이 과정을 통하여 시물레이션 모델이 얼마나 실제 시스템을 대변할 수 있는지를 판단한다. 모델의 검증에서 시물레이션 모델과 실제시스템의 데이터를 비교할 때 발생하는 차이는 입력 데이터의 차이에 의한 영향도 있으며, 이를 통한 모델의 검증 결과는 높은 신뢰성을 보장하지 못한다. 따라서 이 논문에서는 실제와 동일한 입력 데이터를 바탕으로 하는 Trace-Driven 시물레이션을 기반으로 모델을 설계하였다. 한편, 출력데이터들을 하나의 통계량을 통한 검증이 아닌 클래스 별 검증을 하기 위해 데이터마이닝 분야에서 분류기의 성능을 판단하는 F 척도를 응용하여 시물레이션 모델의 검증을 수행하였다. 그 결과, 제안된 검증 방법은 정밀한 모델의 검증을 가능하게 하고, 검증 시에 피드백을 제공함으로써 용이한 수정 작업을 가능하게 한다.

**주요어** : 검증 방법, Trace-driven 시물레이션, 시계열 데이터, F 척도

## 1. 서론

여러 최적화 방법들이 모델의 크기나 가정에 제약되어

있는 것에 비해, 시물레이션은 복잡하고 규모가 큰 시스템을 다룰 수 있는 강력한 의사결정도구이다. 또한, 시물레이션은 다양한 시나리오와 대안들을 제시하여 의사결정자로 하여금 효과적인 의사결정을 할 수 있도록 한다. 이러한 장점들을 바탕으로, 최근 시스템의 다양화와 복잡화의 추세에서 시물레이션이 더욱 주목을 받고 있다<sup>[20]</sup>.

시물레이션 분석 과정 중 모델의 검증 과정은 해당 모델이 실제 시스템을 얼마나 정확하게 대변하는지의 정도를 결정하는 과정이며 이는 시물레이션 분석 과정에서 가장 중요한 단계 중 하나이다<sup>[24]</sup>. 모델의 검증을 통해 모델

\* 이 연구에 참여한 연구자(의 일부)는 '2단계 BK21 사업'의 지원비를 받았음.

2009년 10월 13일 접수, 2009년 11월 30일 채택

<sup>1)</sup> 고려대학교 정보경영공학부

주 저 자 : 황보훈

교신저자 : 이홍철

E-mail; hcllee@korea.ac.kr

을 정확히 평가하는 것이 전체 분석 과정의 신뢰성을 보장하는 길이다.

보다 정확한 모델의 검증을 위해서는 어떤 특정한 통계량으로 전체에 대해서 검증하는 것보다 시계열데이터에 대해 개별적으로 검증하는 것이 합리적이다. 따라서 검증 과정에서 비교할 출력데이터로 총 처리량이나 자원의 이용률 등이 아닌 시계열 데이터를 사용하고, 단순히 평균이나 분산에 대해 비슷한지를 검증하는 것이 아니라 시간의 흐름에 따라 두 시스템 간에 유사한 값들을 갖는가에 대해 검증해야 한다.

한편, 시뮬레이션에서는 적합도 검정(goodness-of-fit test)을 통해 입력 데이터를 특정 분포로 추정하여 모델 내에서 각각의 값들을 발생시킨다<sup>[4]</sup>. 이러한 입력 값들을 바탕으로 출력 값을 얻어내고 이를 실제 시스템의 출력 값들과 비교한다. 그러나 이로 인해 발생하는 입력 값의 차이는 출력 값의 차이에 영향을 미치게 되며 모델의 검증 과정이 전적으로 모델에 대해서만 평가하지 못하는 결과를 초래한다. 시계열 데이터로 얻어지는 출력 값은 값들 사이에 종속성이나 상관성이 존재하며<sup>[4]</sup>, 연속된 데이터의 미세한 차이가 특정 장비 혹은 시스템 전체에까지 영향을 미칠 수 있다. 따라서 실제 시스템과 동일한 입력 데이터를 사용하여 입력 값의 차이에 의한 두 시스템의 차이를 최소화하는 것이 바람직하다.

본 논문에서는 시뮬레이션과 실제 시스템에 동일한 입력 값을 적용하는 Trace-driven 시뮬레이션을 기반으로 시계열 데이터를 클래스로 구분하여 각각의 클래스에 대해 평가하는 검증 방법에 대해 논의할 것이다. 2장에서 시뮬레이션 모델의 검증에 대한 관련 연구에 대해 제시하고 3장에서 가중 F 척도를 사용한 검증 방법에 대해 소개한다. 다음으로 4장에서 간단한 예제를 통해 검증 방법에 대해 평가한 후 5장에서 결론을 맺는 순서로 진행한다.

## 2. 관련 연구

시뮬레이션 모델의 검증 방법에 대해 Sargent(2008)는 표 1과 같이 정리하고 있다<sup>[3]</sup>. 정성적 또는 정량적인 방법을 동원하여 모델을 검증하며 여러 가지 방법이 함께 사용되기도 한다. 이런 다양한 방법들 중에 보다 정확한 모델의 검증을 위해서는 기존의 데이터를 이용한 검증(Historical data validation)을 수행할 필요가 있으며 이는 통계적인 검증 과정을 수반한다.

한편, Balci(1994, 1995)는 시뮬레이션 모델의 검증 및 테스트 방법에 대해 분류를 하고 동적인 기법의 일부로

표 1. 시뮬레이션 모델의 검증 기법

Animation
Comparison to other models
Degenerate tests
Event validity
Extreme condition tests
Face validity
Historical data validation
Historical methods
Internal validity
Multistage validation
Operational graphics
Parameter validity - sensitivity analysis
Predictive validation
Traces
Turing tests

표 2. 시뮬레이션 모델 검증을 위한 통계적 기법

Analysis of variance
Confidence intervals/regions
Factor analysis
Hotelling's $T^2$ tests
Multivariate analysis of variance(MANOVA)
• Standard MANOVA
• Permutation Methods
• Nonparametric Ranking Methods
Nonparametric goodness-of-fit tests
• Kolmogorov-Smirnov Test
• Cramer-Von Mises Test
• Chi-square Test
Nonparametric tests of means
• Mann-Whitney-Wilcoxon Test
• Analysis of Paired Observations
Regression analysis
Theil's inequality coefficient
Time series analysis
• Spectral Analysis
• Correlation Analysis
• Error Analysis
t-Test

표 2와 같이 통계적 기법들을 정의하고 있다<sup>[6,7]</sup>. 다양한 통계적 기법들이 있으며, 모델의 특성에 따라 다른 기법이 적용된다. 그 기법들에 대해서 이 장의 나머지 부분에서 살펴보도록 하겠다.

신뢰 구간(Confidence interval), 대응 관측의 분석(Analysis of paired observations), t-검정(t-Test) 등의 통계적 기법들은 두 개의 시스템의 평균에 대한 검정이며 이

들은 표본평균과 표본표준편차를 사용하여 통계량을 계산하고 그에 따라 모델을 검증한다<sup>[1]</sup>. 윌콕슨 순위합 검정(Wilcoxon rank-sum test) 또는 맨-위트니 검정(Mann-Whitney test)은 정규분포를 따르지는 않지만 독립이며 연속인 표본들의 두 분포의 평균이 같은가를 검정할 때 사용된다<sup>[2]</sup>. Hotelling의 T제곱 검정(Hotelling's  $T^2$  tests)은 다변량의 경우에 변수들 간의 상관관계를 고려하기 위해 공분산 행렬을 추가하여  $T^2$  통계량을 정의하고 이를 통해 평균에 대한 검정을 실시한다<sup>[21]</sup>. 그러나 이러한 방법들과 같이 평균에 대해 검정할 경우에는 평균과 전체적인 편차의 정도뿐만 아니라 각각의 시점에서 관측 값이 같은가를 판단해야 하는 시계열 데이터의 검증에서 그 타당성을 평가하기가 어렵다.

카이제곱 검정(Chi-square test)은 모델과 실제 시스템에서 각 값의 빈도의 분포가 같은가의 여부를 판단한다. 그러나 빈도를 측정하는 구간의 설정이 어려우며 비정규성을 가진 경우에는 분포가 구간에 더 민감하게 반응한다<sup>[25]</sup>. K-S 검정(Kolmogorov-Smirnov test, K-S test)은 분포에 크게 영향을 받지 않으며, 카이제곱 검정과는 다르게 모델의 각 데이터들의 누적 확률 값과 실제 시스템 데이터들의 누적 확률 값을 비교함으로써 구간의 설정에 제약을 받지 않는다<sup>[12]</sup>. 이러한 방법들은 단순히 각 값들의 빈도에 대해서만 평가를 하기 때문에 데이터들의 순서나 트렌드와 상관없이 값의 분포만 비슷하면 귀무가설을 채택한다. 이 역시 시계열 데이터의 검증에는 적합하지 않다.

Theil의 부등 계수(Theil's inequality coefficient, TIC)은 두 시스템의 값의 차이의 제곱합과 각각의 시스템의 값의 제곱의 합을 더한 값과의 비율로 얻어 진다<sup>[11]</sup>. 이 방법 또한 데이터들의 합으로 계수 값을 산출하므로 각각의 데이터들이 의미 있는 값을 가지고 있는가를 판단하기 어렵다.

분산분석(Analysis of variance, ANOVA)은 기본적으로 모집단의 정규성과 등분산을 가정하며 이는 2개 이상의 모집단의 평균들이 같은가를 F 통계량을 사용하여 검정한다<sup>[5]</sup>. 여기서 여러 개의 종속변수를 다루거나 각각의 종속변수들 간의 영향까지 고려할 때는 다변량분산분석(Multivariate analysis of variance, MANOVA)을 사용한다. 그러나 분산분석 또한 평균에 대한 검정을 하며, 이렇게 F분포를 이용하는 과정은 t분포를 이용하는 과정과 한 치의 차이도 없는 동일한 결과를 얻어낸다<sup>[3]</sup>.

회귀분석(Regression analysis)은 실제 시스템과 모델에 대해  $y = \beta_0 + \beta_1 x$ 라는 1차 선형회귀방정식을 세우고 상수항  $\beta_0 = 0$ , 기울기  $\beta_1 = 1$ 의 만족 여부를 검정한다<sup>[22]</sup>. 그러나 Kleijnen 등(1996)은  $\beta_0 = 0$ ,  $\beta_1 = 1$ 의 가설은 유효한

모델조차 기각해버리는 경우가 많으며, 실제시스템과 모델의 결과 값의 차를 결과 값의 합에 대해 회귀분석 하여 얻어낸 계수 값들로 t-검정을 통해 평균과 분산이 같은지를 검정하도록 제안하고 있다<sup>[17]</sup>. 그러나 기존의 회귀분석을 통한 모델의 검증은 적합하지 않은 것으로 수년간 판단되어 왔으며<sup>[9]</sup>, 새로 제안한 방법도 결국엔 평균과 분산의 동일성을 검정하는 데에 그치고 있다.

마지막으로 시계열분석(Time series analysis) 중에서 대표적인 스펙트럼분석(Spectral analysis)은 각 출력 프로세스에 대해 예측된 자기공분산(autocovariance) 함수를 푸리에 코사인 변환(Fourier cosine transformation)을 이용하여 표본 스펙트럼을 만들고, 두 스펙트럼의 로그 값의 차이에 대해 신뢰구간을 구해서 평가한다. 그러나 이 방법은 일정한 공분산을 가져야 하고 고차원적인 수학적 기술을 요구하며 이러한 신뢰구간을 시뮬레이션 모델의 검증에 이용하기 어렵다는 단점이 있다<sup>[18]</sup>.

### 3. 가중 F 척도를 사용한 검증 방법

본 논문에서는 데이터마이닝분야에서 분류기의 성능을 판단하는데 주로 사용하는 F 척도를 바탕으로 가중치 값을 추가하여 시뮬레이션 모델의 검증을 수행하는 방법을 제안한다. 이렇게 얻어진 가중 F 척도는 클래스별로 얻어 지므로 시뮬레이션 모델의 검증을 할 때 각각의 클래스, 즉 이벤트에 기반한 출력데이터의 부분그룹들에 대해 각 각 평가할 수가 있다.

#### 3.1 F 척도

데이터마이닝 분야에서는 모델의 성능을 평가할 때, 혼동행렬(confusion matrix)을 사용하여 정확성(accuracy) 또는 F 척도(F-measure) 등의 값으로 각 모델이 얼마나 좋은 성능을 나타내는지를 평가한다<sup>[26]</sup>.

혼동행렬은 실제의 데이터와 예측된 데이터에 해당하는 클래스들을 각각 행과 열에 배치함으로써 데이터를 클래스에 따라 구분 지어서 나타낸 행렬이다. 다음의 그림 1

		예측된 클래스	
		yes	no
실제의 클래스	yes	TP (true positive)	FN (false negative)
	no	FP (false positive)	TN (true negative)

그림 1. 혼동행렬의 구조

은 2개의 클래스에 대한 혼동행렬의 구조를 나타낸다. 표에서 제시한 혼동행렬에서는 데이터가 긍정(yes)과 부정(no)의 두 개의 클래스로 나뉘어있으며 이에 따라 행렬의 크기는 2×2가 된다. 왼쪽 상단의 참 긍정(true positive)값은 긍정적 클래스에 속해 있는 실제의 데이터에 대해 긍정적 클래스에 속한다고 예측한 데이터의 총 개수가 되며, 오른쪽 상단의 거짓 부정(false negative)값은 같은 방식으로 긍정적 클래스에 속해 있는 실제의 데이터에 대해 부정적 클래스에 속한다고 예측한 데이터의 총 개수가 된다.

여기서 정확성은 식 (1)과 같으며, 이는 전체 예측한 데이터 중에 정확히 예측한 데이터의 수로 얻어진다.

$$\text{정확성} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

F 척도는 재현성(recall)과 정밀도(precision)의 조화평균으로 구할 수 있으며, 재현성은 분류기에 의하여 정확하게 예측되는 긍정적 사례의 비율을 측정하고, 정밀도는 분류기가 긍정적 클래스로 선언한 그룹에서 실제 긍정적 사례로 판명되는 항목들의 비율을 결정한다. 이는 식 (2)와 (3)을 통해 확인할 수 있다.

$$\text{재현성} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{정밀도} = \frac{TP}{TP + FP} \quad (3)$$

재현성 수치가 큰 분류기는 부정적 클래스로 잘못 분류되는 긍정적 사례의 수가 매우 적으며, 정밀도의 값이 클수록 분류기에 의하여 검출되는 거짓 긍정 오류의 수는 적어진다.

$$F \text{ 척도} = \frac{2TP}{2TP + FP + FN} \quad (4)$$

식 (4)를 통해 F 척도를 구할 수 있으며, 임의의 두 숫자 x와 y의 조화평균은 두 수 중에서 작은 숫자에 가까운 경향이 있기 때문에 F 척도가 높다는 것은 정밀도와 재현성 모두 상당히 크다는 것을 보장한다<sup>[23]</sup>.

### 3.2 출력 데이터의 클래스 구분

F 척도를 통해 시뮬레이션 모델의 검증을 하기 위해서는 우선 출력데이터를 특정 클래스들로 구분하여야 한다. 일반적으로 2개의 클래스를 구성하고 2×2 크기의 혼동행렬을 구축하지만, 여기서는 여러 개의 클래스를 구성한다.

출력데이터는 다양한 형태가 될 수 있지만, 본 논문에서는 tact time(or interarrival time)에 대한 시계열 데이터를 출력데이터로 삼는다. tact time을 통해서 하나하나의 로드들이 어떤 이벤트와 어떤 프로세스를 거쳐서 최종 목적지까지 도착하는지를 좀 더 쉽게 확인할 수 있으며, 이를 검증함으로써 보다 높은 수준의 모델 검증이 가능해진다.

특정 tact time을 갖게 되는 로드는 특정한 이벤트를 거쳐서 최종 목적지에 다다르게 된다. 즉 동일한 tact time을 갖는 로드는 동일한 이벤트를 거친 것으로 볼 수 있다. 여기서 동일한 이벤트라 함은 단순히 같은 작업과 공정 순서, 이동경로를 가지는 것뿐만 아니라 모든 과정에서 동일한 시간을 거쳐서 처리된 것을 말한다. 이산 이벤트 시뮬레이션(discrete event simulation)은 이벤트를 기반으로 이루어지기 때문에 클래스의 구분 또한 이러한 이벤트를 바탕으로 이루어지는 것이 합리적이다. 임의의 한 로드가 거친 모든 이벤트들의 조합을 하나의 이벤트 집합이라 하자. 클래스를 구분 지을 때, 동일한 이벤트 집합을 가지는 데이터들을 하나의 클래스 안에 포함한다. 즉, 서로 다른 이벤트 집합을 갖는 출력데이터마다 별도의 클래스를 생성하는 것이다. 최종적으로 서로 다른 이벤트 집합의 수만큼 클래스가 생성이 된다.

### 3.3 모델의 검증을 위한 가중 F 척도

출력데이터를 그 값에 따라 실제 데이터는  $r_1, r_2, \dots, r_n$ 의 클래스에, 시뮬레이션 데이터는  $s_1, s_2, \dots, s_n$ 의 클래스에 할당을 하고 그림 2와 같이 혼동행렬을 구성한다. 여기서  $r_i$ 의 값은  $s_i$ 의 값과 같다.

여기서  $f_{i1}$ 은  $r_i$ 클래스에 해당하는 실제 데이터에 대해 시뮬레이션 데이터가  $s_1$ 클래스에 해당하는 값이 나온 횟수를 말하며, 같은 방식으로  $f_{i2}$ 는  $r_i$ 클래스에 해당하는 실제 데이터에 대해 시뮬레이션 데이터가  $s_2$ 클래스에 해당하는 값이 나온 횟수를 말한다. 위와 같은 혼동행렬을 구성하고 나면, 첫 번째 클래스에 대한 나온 횟수  $R_1$ 과 는 값

		시뮬레이션 데이터			
		$s_1$	$s_2$	...	$s_n$
실제 데이터	$r_1$	$f_{11}$	$f_{12}$	...	$f_{1n}$
	$r_2$	$f_{21}$	⋮		
	⋮	⋮			
	$r_n$	$f_{n1}$			$f_{nn}$

그림 2. 모델 검증을 위한 혼동행렬

$P_l$ 은 식 (5), (6)과 같이 얻어진다.

$$R_1 = \frac{f_{11}}{\sum_{j=1}^n f_{1j}} \quad (5)$$

$$P_1 = \frac{f_{11}}{\sum_{i=1}^n f_{i1}} \quad (6)$$

식 (5)와 (6)에 의해서 F 척도는 각각의 클래스 당 하 나씩 총 n개가 얻어지며, 이때 a번째 클래스의  $F_a$ 값은 다음 식 (7)과 같이 구할 수 있다.

$$F_a = \frac{2f_{aa}}{\sum_{j=1}^n f_{aj} + \sum_{i=1}^n f_{ia}} \quad (7)$$

한편, 여기서 한 가지 주목할 것은 F 척도는 정확하게 구분 하였는가 아닌가에 대해서만 평가를 한다는 것이다. 따라서 시뮬레이션 데이터 값이 실제 시스템의 데이터 값과 어느 정도의 차이를 가지고 다른지와 상관없이, 단순히 값이 다른 부분에 대해서는 동일하게 평가가 된다. 기존에는 F 척도를 주로 두 개의 클래스에 대해 평가를 하였기 때문에 이러한 현상이 발생하지 않았지만, 여러 개의 클래스에 대한 평가를 하기 위한 경우에 F 척도는 이러한 부분을 감안하여 수정되어야 할 필요가 있다.

기존의 재현성을 구하는 식 (5)를 시뮬레이션에서 다른 클래스로 관측된 도수들에 대해 다시 정리하면 식 (8)과 같다.

$$R_1 = \frac{\sum_{j=1}^n f_{1j} - \sum_{j=2}^n f_{1j}}{\sum_{j=1}^n f_{1j}} \quad (8)$$

여기서 다른 클래스로 관측된 도수, 즉 실제의  $r_l$ 클래스에 대해서는  $f_{12}, f_{13}, \dots, f_{1n}$ 의 도수들에 대해 각각 가중치를 부여하여 그만큼 더 빼주도록 한다. 가중치  $\omega_{1j}$ 는 식 (9)와 같이  $s_j$ 와  $s_1$ 의 차이를  $E(s_n)$ 으로 나누어 주는 표준화 과정을 거쳐서 구하며, 이를 통해 가중 재현성 (weighted recall: WR)은 식 (10)과 같이 얻어질 수 있다.

$$\omega_{1j} = \frac{|s_j - s_1|}{E(s_n)} \quad (9)$$

$$WR_1 = \frac{\sum_{j=1}^n f_{1j} - \sum_{j=2}^n (1 + \omega_{1j})f_{1j}}{\sum_{j=1}^n f_{1j}} \quad (10)$$

같은 방식으로 가중 정밀도(weighted precision: WP)에 대해 식 (11)의 가중치를 사용하여 식 (12)를 얻어낼 수 있으며, 식 (10)과 (12)를 이용해 가중 F 척도 (weighted F-measure: WF-measure)를 구하면 다음의 식 (13)과 같다.

$$\omega_{i1} = \frac{|r_i - r_1|}{E(r_n)} \quad (11)$$

$$WP_1 = \frac{\sum_{i=1}^n f_{i1} - \sum_{i=2}^n (1 + \omega_{i1})f_{i1}}{\sum_{i=1}^n f_{i1}} \quad (12)$$

$$WF_a = \frac{2 \left\{ \sum_{j=1}^n f_{aj} - \sum_{j \neq a}^n (1 + \omega_{aj})f_{aj} \right\} \left\{ \sum_{i=1}^n f_{ia} - \sum_{i \neq a}^n (1 + \omega_{ia})f_{ia} \right\}}{2 \sum_{i=1}^n \sum_{j=1}^n f_{aj}f_{ia} - \sum_{i \neq a}^n \sum_{j=1}^n (1 + \omega_{ia})f_{aj}f_{ia} - \sum_{i=1}^n \sum_{j \neq a}^n (1 + \omega_{aj})f_{aj}f_{ia}} \quad (13)$$

이를 통해, 각각의 클래스에 대해 가중치를 고려한 가중 F 척도를 구할 수가 있다.

### 3.4 검증 과정 및 결과의 해석

앞서 설명한 것과 같이 가중 F 척도는 클래스의 개수 만큼 n개가 얻어지며, 얻어진 값을 통해 해당 클래스에 대한 평가가 가능하다. 특정 클래스가 얼마나 잘 맞는가, 즉 시뮬레이션 모델에서 발생하는 특정 이벤트가 실제 시스템에서의 해당 이벤트를 얼마나 잘 반영하였는가를 평가할 수 있다.

F 척도에 관한 논문들을 살펴보면, F 척도가 어느 정도의 수준이 되어야 특정 분류기가 좋다는 기준은 정해져 있지 않다. 다만, 상대적으로 F 척도 값을 비교함으로써 새로 제시한 분류기가 다른 분류기보다 더 좋은 성과를 얻어낸다는 결론을 제시한다<sup>19)</sup>. 그러나 시뮬레이션 모델을 평가함에 있어서 상대적인 기준을 통한 판단은 위험한 결과를 초래할 수 있다. 따라서 가중 F 척도를 통해 모델의 검증 여부를 결정할만한 절대적인 기준이 필요하다. 본 논문은 모델 검증의 신뢰성을 높이기 위해서 개별적인 클래스에 대한 평가와 전체 클래스에 대한 평가를 위한

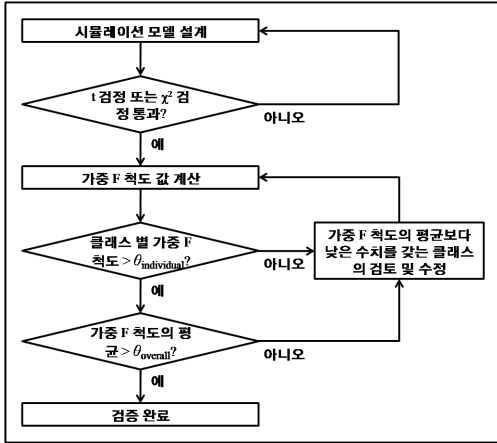


그림 3. 가중 F 척도를 이용한 모델의 검증 과정

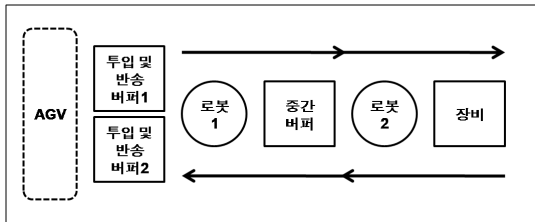


그림 4. 유연 생산 시스템 모델의 배치도

두 가지 평가 기준을 제시한다.  $\theta_{individual}$ 을 통해 각각의 이벤트 집합을 바탕으로 구성된 개별 클래스들에 대해 평가하고, 추가적으로 전체 모델의 검증을 위해 모든 클래스들의 가중 F 척도 값들의 평균에 대해서  $\theta_{overall}$ 의 기준으로 평가한다.  $\theta_{individual}$ 과  $\theta_{overall}$ 은 의사결정자가 만족할 만한 수준으로 그 값을 정할 수 있다.

한편, 기존의 검증 방법론은 대부분 단순히 시뮬레이션 모델과 실제 시스템이 얼마나 유사한가에 대한 평가에 그치고 있다. Chung(2004)은 검증 결과 수정이 필요할 경우에 기존의 검증 방법들은 수정해야 할 부분을 알려주지 못한다고 서술했다<sup>8)</sup>. 그러나 본 논문에서 제안하는 방법은 클래스 기반으로 모델에 대한 평가를 진행하기 때문에, 이를 토대로 문제가 되는 이벤트들을 판단하고, 수정 작업 시에 참고자료로 활용이 가능하다.

가중 F 척도를 이용한 모델의 전체적인 검증 과정에 대해 그림 3에 나타내었다. 가중 F 척도를 이용한 검증 과정은 보다 정밀한 모델의 검증을 위한 과정이며, 이에 따라 많은 작업량을 수반한다. 또한 평균의 유사성을 보장하지는 않기 때문에 평균값의 유사 정도가 중요한 영향을 미치는 공정에 대해서는 충분한 신뢰성을 보장한다고 할

수 없다. 따라서 초기에는 t-검정이나 카이제곱 검정 등의 기존의 방법들을 통해 개략적인 평가를 진행하고 이러한 과정을 거친 모델에 대해서 추가적으로 가중 F 척도를 이용하여 평가하도록 한다.

모든 클래스가  $\theta_{individual}$  이상의 가중 F 척도 값을 가지고 전체 클래스의 가중 F 척도 값의 평균이  $\theta_{overall}$  이상의 값을 가지면 모델의 검증은 완료된다. 그러나 이 조건을 만족시키지 못하면 전체 가중 F 척도 값들의 평균보다 낮은 수치를 갖는 클래스들을 선정하고 이를 검토하게 된다. 초기 모델의 경우 대부분의 클래스들이  $\theta_{individual}$  이상의 가중 F 척도 값을 가지기 어렵다. 복잡한 모델의 검증 시에는  $\theta_{individual}$ 을 넘지 못하는 클래스들을 모두 검토할 대상으로 선정할 경우 상당히 많은 수의 클래스들에 대해 일일이 검토해야 하며 이는 많은 시간을 필요로 한다. 따라서 검토할 클래스의 범위를 줄이기 위해  $\theta_{individual}$  이하의 가중 F 척도 값을 갖는 클래스들 중에서 전체 가중 F 척도의 평균값과의 비교를 통해 클래스들을 선정한다.

## 4. 사례 연구(Case Study)

앞서 제시한 검증 방법론의 효과를 검증하기 위해서 이 장에서는 유연 생산 시스템 예제를 통해 기존의 검증 방법론들과 가중 F 척도의 검증 결과의 차이에 대해 살펴볼 것이다. 또한, 가중 F 척도는 시뮬레이션 모델이 적합하지 않다고 판정할 때, 이후에 수정 단계에서 특정 이벤트 집합에 대해 검토하도록 제안할 수 있으며 그 과정에 대해서도 살펴볼 것이다.

### 4.1 유연 생산 시스템 모델

모델은 로봇 2대와 버퍼 3개, 그리고 장비 1대로 구성이 되어있다. 로봇은 각 로드를 운반하는 역할을 하며, 버퍼에는 해당 시스템으로 로드의 투입을 위한 버퍼 2개와 선·후처리 공정의 역할을 하는 버퍼가 있다. 각 로봇과 장비의 용량은 1 EA이며, 모델의 배치는 그림 4와 같다.

투입 및 반송 버퍼는 총 25단으로 구성이 되고 중간버퍼는 2단으로 구성이 되며 여기에서 선·후처리 공정이 이루어진다. 장비는 총 2가지 공정을 수행하고 모든 로드는 2가지 작업을 차례로 거치게 되며 장비에는 한 번에 두 개의 로드가 실행될 수 있다. 전체적인 공정의 순서는 다음과 같다. 무인운반차(AGV)를 통해서 로드들이 투입 버퍼에 쌓이고 로봇1을 거쳐 중간 버퍼에 투입된다. 선처리 공정을 마친 후, 로봇2를 통해 장비로 이동하여 공정1을 마친 후 공정2를 수행한다. 공정을 모두 마친 로드는 로

봇2를 통해 중간 버퍼에 투입되며 후처리 공정을 끝내고 로봇1을 거쳐 다시 반송(투입) 버퍼에 쌓인다.

공정이 진공 상태에서 진행이 되어야 하기 때문에 중간 버퍼를 기준으로 오른쪽에 배치되어있는 로봇2와 공정 1, 2가 포함되어 있는 지역은 진공 상태이고 왼쪽에 배치되어있는 지역은 대기 상태이다. 이에 따라, 선·후처리 공정은 중간 버퍼 내부의 기압을 바꾸어주는 작업이며, 이는 버퍼 2단 모두에 동시에 적용된다.

#### 4.2 Trace-driven 시뮬레이션의 적용

Trace-driven 시뮬레이션은 시뮬레이션 프로그램에 시간적인 순서로 실제 시스템에서 발생 하는 입력 데이터를 넣는 것이다<sup>10)</sup>. 즉, 시뮬레이션의 모든 환경을 실제와 동일하게 하여 실제의 출력 데이터와 비교함으로써 모델 자체에 대해서만 정확하게 평가할 수 있도록 하는 것이다. 이는 현재 프록시 서버(proxy server)의 스케줄링 알고리즘의 평가<sup>15)</sup>와 디스크 캐쉬(disk cache) 알고리즘 평가<sup>27)</sup> 등의 분야에서 널리 쓰이고 있다.

본 논문에서는 시뮬레이션 모델링 툴로 AutoMod II를 사용하고 입력 및 출력 데이터를 Excel과 연동하여 입력하고 출력한다. 이 데이터들은 시간 단위의 데이터이며 모델은 무수히 많은 종류의 입력 데이터를 필요로 한다. 따라서 모든 입력 변수에 대해 실제 시스템과 동일한 데이터를 적용하지 않고, 무시해도 될 정도의 미세한 차이만 갖는 입력 변수에 대해서는 일정한 값을 주며 그 외의 변수에 대해서만 실제의 데이터를 입력하도록 한다. 한편, 다음은 예제 모델에서 사용될 입력데이터들이다.

- 무인운반차의 도착 시간
- 무인운반차에서 투입버퍼로의 상차(Loading) 시간
- 로봇1의 투입 버퍼에 대한 상차, 하차(Unloading)

시간

- 로봇1의 중간 버퍼에 대한 상차, 하차 시간
- 중간 버퍼의 기압 변경 시간(대기→진공, 진공→대기)
- 로봇2의 공정1에 대한 상차 시간
- 로봇2의 공정2에 대한 하차 시간
- 로봇1의 이동 시간(투입 버퍼, 중간 버퍼 간)
- 로봇2의 이동 시간(중간 버퍼, 공정1, 공정2 간)

로봇의 상·하차 시간 및 이동시간은 무시해도 될 정도의 미세한 차이만 존재할 뿐 거의 일정하기 때문에 이 값들에 대한 입력변수는 일정한 값으로 설정한다. 그러나 무인운반차의 도착 시간은 상황에 따라 다양한 값들을 가지게 되므로 실제 시스템에서의 도착 시간을 그대로 입력 데이터로 사용할 수 있도록 한다. 또한 중간 버퍼의 기압 변경 시간의 경우 시간 기준으로 작업이 완료되는 것이 아니라 특정 기압으로 변경될 때까지의 시간을 측정하게 되므로 시간의 변동성이 다른 작업들에 비해 크다. 따라서 기압 변경 시간 역시 실제 시스템에서의 시간과 동일하게 입력한다.

#### 4.3 시뮬레이션 모델링 및 출력 데이터

위에서 나열한 사항들을 바탕으로 AutoMod 12.1.3을 사용하여 시뮬레이션 모델을 구축하였으며, 그림 5는 모델의 실행 화면이다.

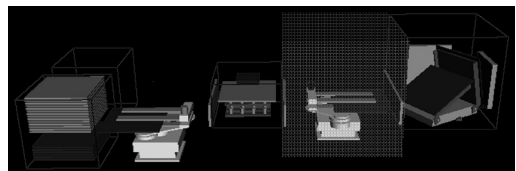


그림 5. 시뮬레이션 모델 실행 화면

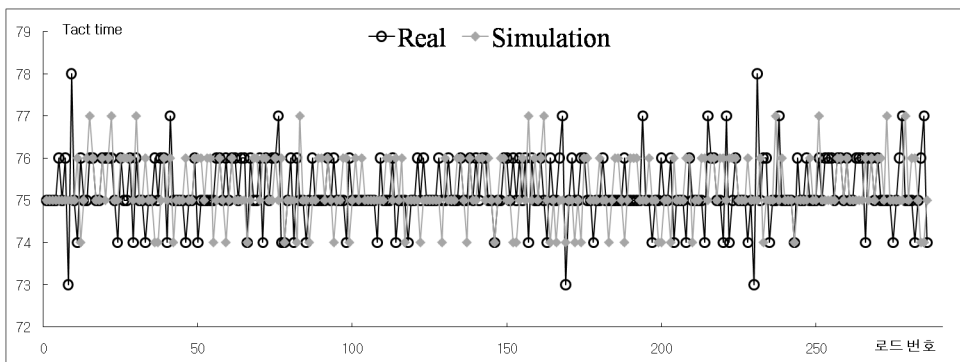


그림 6. 초기의 출력 데이터 비교

실제 시스템에 대해 291개의 데이터 값이 존재하므로 시물레이션 모델 또한 해당 개수만큼의 데이터가 필요하며, 데이터 수집을 위해 시물레이션 시간으로 약 2시간 동안 모델을 실행하였다. 그림 6은 이에 따른 시물레이션 모델의 출력 데이터와 실제 시스템의 데이터를 비교한 그래프이다.

현재의 시스템은 전체적으로 73초와 78초 사이의 정수 값의 tact time을 가지며, 이 범위 내에서의 차이에 따라 다른 이벤트 집합을 갖는 로드로 간주할 수 있다. 그러나 위의 그래프를 통해서 두 시스템간의 차이를 눈으로도 확인할 수 있으며, 이러한 상태로는 두 시스템이 유사하다고 하기에는 무리가 있다.

#### 4.4 검증 방법의 비교

시물레이션 모델에서 얻어진 데이터와 실제의 데이터

표 3. 타 방법들에 의한 초기 모델의 검증

검증 방법	통계량 값	기각역	유의 확률
t-검정	0	-	1
월복순 순위합 검정	-0.028	-	0.978
카이제곱 검정	5.253	-	0.148
K-S 검정	0.010	-	1
TIC 검정	0.0004	0.3	-

표 4. 초기 모델의 혼동행렬

R \ S	73	74	75	76	77	78
73	0	1	2	0	0	0
74	0	5	13	16	1	0
75	0	14	98	40	5	0
76	0	15	39	27	4	0
77	0	1	7	1	0	0
78	0	0	2	0	0	0

표 5. 초기 모델의 여러 가지 척도

	73	74	75	76	77	78
가중 재현성	-0.02	0.125	0.619	0.306	-0.03	-0.04
가중 정밀도	0	0.121	0.602	0.310	-0.02	0
가중 F 척도	-	0.123	0.610	0.308	-0.02	-

에 대해 기존의 방법들로 모델 검증을 해 본 결과가 표 3과 같이 나타났다. 모델 검증을 위해 SPSS 12.0k와 Excel을 이용했다.

대부분의 검정을 통해서 거의 1에 가까운, 매우 높은 유의확률이 얻어졌으며, TIC 검정과 같은 경우 0.3보다 작은 값을 가지면 충분히 잘 설계되었다고 할 수 있는데, 얻어진 TIC 값은 0.3에 비해 매우 작은 값이다. 카이제곱 검정의 경우에는 다른 방법에 비해서 유의확률이 작게 나왔지만 유의수준 0.05보다는 크기 때문에 두 시스템이 유사하다는 결론을 내린다. 따라서 기존의 방법들은 현재의 모델과 실제 시스템은 같다는 가설을 어떠한 근거로도 기각하지 못한다.

다음으로 가중 F 척도를 통해 두 시스템을 비교해 보도록 하겠다. 우선 클래스에 대한 정의가 필요하며, 출력 값이 73에서 78 사이의 정수 값이므로 각각의 값을 동일한 이벤트 집합을 갖는 하나의 클래스로 하여 총 6개의 클래스를 정의한다. 표 4는 해당 모델의 검증을 위한 혼동행렬이다.

가중 F 척도를 통한 모델의 검증을 위한 다음 단계로 개별적인 클래스와 전체 클래스에 대한 평가 기준에 대해서 정의할 필요가 있다. 의사결정자는 개별클래스에 대한 평가 기준  $\theta_{\text{individual}} = 0.6$ , 전체 클래스에 대한 평가 기준  $\theta_{\text{overall}} = 0.8$ 의 수준을 제시한 것으로 가정한다. 모델의 검증 과정은 이와 같은 평가 기준을 만족할 때까지 반복이 되며, 평가 기준을 만족하면 검증 과정을 완료한다.

이 혼동행렬을 통해 각 클래스 별 측정치를 구한 결과 표 5와 같은 수치를 얻을 수 있었다. 아래의 표에서 음의 값이 나온 것들은 해당하는 값에 대해 모델과 실제의 데이터가 일치하는 데이터가 없기 때문이다. 가중치를 주었기 때문에 가중치에 따라 음수 값이 나올 수 있다. 클래스 73과 78에 대한 F 척도 값이 없는 것은 가중 정밀도의 값이 0이기 때문에 조화평균을 구하기 위해 이에 대한 역수를 취할 수 없기 때문이다.

전체적으로 가중 F 척도의 값은 현저히 낮게 나타나며 이는 다른 검증 방법들에 의한 결과와 상당히 대비된다. 다른 방법들은 현재의 모델이 실제 시스템과 다르다는 판단을 하지 못하지만 본 논문에서 제안한 방법은 두 시스템이 차이가 있다고 검증하였다.

한편, 클래스 75를 제외하고 가중 F 척도가 앞서 가정한  $\theta_{\text{individual}} = 0.6$ 보다 낮은 5개의 클래스들 중에 가중 F 척도들의 평균 값 0.169보다 낮은 수치를 갖는 클래스 73, 74, 77, 78에 대해서 각각의 이벤트가 제대로 반영되었는지 검토한다. 이에 따라, 해당 클래스들에서 중간 버퍼의



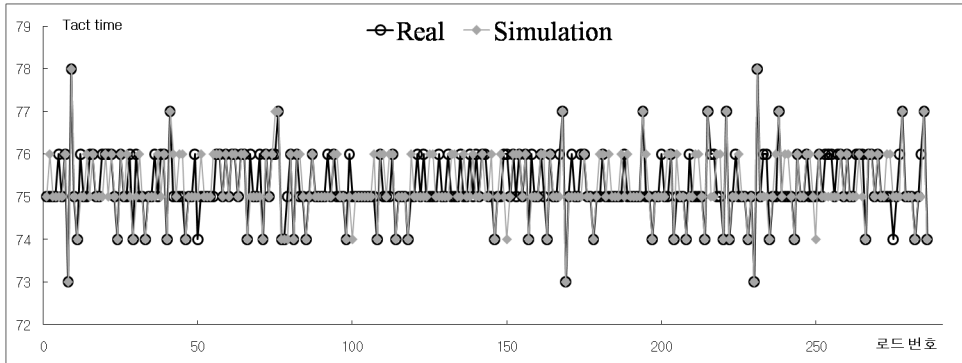


그림 7. 수정 후 출력 데이터 비교

표 6. 타 방법들에 의한 수정된 모델의 검증

검증 방법	통계량 값	기각역	유의 확률
t-검정	-0.107	-	0.914
윌콕슨 순위합 검정	-0.145	-	0.885
카이제곱 검정	0.279	-	0.960
K-S 검정	0.083	-	1
TIC 검정	0.0002	0.3	-

표 7. 수정된 모델의 혼동행렬

R \ S	73	74	75	76	77	78
73	3	0	0	0	0	0
74	0	33	2	0	0	0
75	0	3	110	44	0	0
76	0	1	44	39	1	0
77	0	0	0	0	9	0
78	0	0	0	0	0	2

표 8. 수정된 모델의 가중 F 척도

	73	74	75	76	77	78
가중 F 척도	1	0.915	0.699	0.457	0.947	1

표 9. 최종 모델의 가중 F 척도

	73	74	75	76	77	78
가중 F 척도	1	0.915	0.935	0.898	0.947	1

기압 변경 과정의 처리에 문제가 있는 것을 발견하고 모델을 수정하였으며, 그 때의 실제 시스템과 시뮬레이션 모델의 출력데이터를 비교한 그래프가 그림 7에 나타나 있다. 초기 모델보다 확연히 두 개의 그래프가 더 유사하게

겹치는 모습을 확인할 수 있으며, 이때 기존의 방법들로 모델의 검증을 한 결과 표 6과 같은 유의 확률을 얻었다.

수정 후에도 대부분의 검정을 통해서 1에 가까운 유의 확률을 얻어낼 수 있으며, 카이제곱 검정의 유의 확률도 눈에 띄게 증가한 것을 알 수 있다. 다음의 표 7은 수정된 모델의 혼동행렬을 나타낸다.

이때의 가중 F 척도는 표 8과 같이 얻어지며 대부분의 클래스에 대해 1에 가까운 값이 나오는 것을 확인할 수 있다. 그러나 클래스 76의 가중 F 척도가  $\theta_{\text{individual}} = 0.6$  보다 낮기 때문에 해당하는 각각의 이벤트에 대해 검토하였으며 그 결과 공정1의 작업 시간에 미세한 지연 시간이 포함되어 있어 그 시간이 중복으로 적용되어 있는 것을 발견하였다.

이를 수정하여 최종적으로 얻어낸 가중 F 척도는 다음의 표 9와 같다. 모든 클래스들이 평가 기준  $\theta_{\text{individual}} = 0.6$  이상의 가중 F 척도 값을 가지며 이들의 평균값도 0.949로 평가 기준  $\theta_{\text{overall}} = 0.8$ 보다 큰 값을 가지므로 모델의 검증 과정을 완료한다.

## 5. 결론 및 향후 연구 계획

시뮬레이션 분석에 있어서 모델의 검증은 상당히 중요한 부분을 차지한다. 입력데이터를 실제 시스템과 동일하게 사용하는 Trace-driven 시뮬레이션을 통하여 모델 자체에 대해서만 검증을 할 수 있으며 이를 통해 보다 신뢰성 있는 모델을 설계할 수 있다. 한편, 기존의 검증 방법들은 특정 통계량을 바탕으로 평가하는 것이 대부분이기 때문에 개별적인 데이터들에 대한 평가가 어려웠다. 그러나 가중 F 척도를 이용하여 모델을 검증하면 각각의 이벤트 집합을 바탕으로 구성된 개별 클래스들에 대해 평가를 할 수 있으며 이 평가 결과는 수정 작업을 할 때 피드백

작용을 하여 작업을 용이하게 한다.

그러나 각각의 이벤트 집합에 대해 개별적인 클래스를 설정하게 되면 복잡한 시스템의 경우 고려해야 할 클래스의 개수가 너무 많아지며 그에 따라 평가 과정이 상당히 복잡해질 수 있다. 따라서 복잡한 시스템을 고려하여 클래스의 구분법을 구체화하는 방법에 대해 추가적인 연구가 필요하다. 이러한 점을 보완한다면 가중 F 척도를 사용하여 보다 정밀한 시뮬레이션 모델의 검증이 가능할 것이다.

## 참 고 문 헌

1. 강석복, 우정수 (2002), *통계적 추정과 가설 검정*, 경문사, pp. 156-318.
2. 강석복, 우정수 (2008), *수리통계학의 이해*, 경문사, pp. 291-358.
3. 김대학, 박숙영 (2004), *SPSS를 활용한 분산분석*, 교우사, pp. 39-66.
4. 김해경, 김태수 (2003), *시계열 분석과 예측 이론*, 경문사, pp. 3-16.
5. 최규정, 권용만, 장인홍, 이장재, 이정화 (2007), *이공계를 위한 확률과 통계학*, 자유아카데미, pp. 265-277.
6. Balci, o. (1994), "Validation, verification, and testing techniques throughout the life cycle of a simulation study", *Annals of Operations Research*, Vol. 53, pp. 121-173.
7. Balci, o. (1995), "Principles and techniques of simulation validation, verification, and testing", *Proceedings of the 1995 Winter Simulation Conference*.
8. Chung, C. A. (2004), *Simulation modeling handbook : a practical approach*, CRC Press, pp. 8-4.
9. Harrison, S. R. (1990), "Regression of a model on real-system output: An invalid test of model validity", *Agricultural systems*, Vol. 34, No. 3, pp. 183-190.
10. Kleijnen, J. P. C. (1999), "Validation of models: statistical techniques and data availability", *Simulation Conference, 1999. Proceedings of the 1999 Winter*, pp. 647-654.
11. Murray-Smith, D. J. (1998), "Methods for the external validation of continuous system simulation models", *Mathematical and computer modelling of dynamic systems*, Vol. 4, No. 1, pp. 5-31.
12. Ross, S. M. (2006), *Simulation 4th Ed.*, Elsevier Academic Press, pp. 294-306.
13. Sargent, R. G. (2008), "Verification and Validation of Simulation Models", *Proceedings of the 2008 Winter Simulation Conference*.
14. Banks, J., Carson, S. J., Nelson, L. B., and Nicol, M. D. (2005), *Discrete-event system simulation 4th ed.*, Pearson Prentice Hall, pp. 305-346.
15. El A. K. and Rachid (2005), "Class-based weighted fair queueing: validation and comparison by trace-driven simulation", *International journal of communication systems*, Vol. 18, No. 10, pp. 975-994.
16. Hu, D., Xie, C., and CaiBin, C. (2005), "A Study of Parallel Prefetching Algorithms Using Trace-Driven Simulation", *Simulation Conference 1996. Proceedings of the 1996 Winter*, pp. 352-359.
17. Kleijnen, J. P. C., Bettonvil, B. and Van, G. W. (1996), "Validation of trace-driven simulation models: regression analysis revisited", *Simulation Conference 1996. Proceedings of the 1996 Winter*, pp. 352-359.
18. Law, A. M. and Kelton, W. D. (2000), *Simulation modeling and analysis 3rd Ed.*, McGraw-Hill, pp. 283-290.
19. Lazarevic-McManus, N., Renno, J. R. and Makris, D. (2008), "An object-based comparative methodology for motion detection based on the F-Measure", *Computer vision and image understanding : CVIU*, Vol. 111, No. 1, pp. 74-85.
20. Liu, F. and Yang, M. (2009), "An Optimal Design Method for Simulation Verification Validation and Accreditation Schemes", *Simulation*, Vol. 85, No. 6, pp. 375-386.
21. McFarland, J. and Mahadevan, S. (2008), "Multivariate significance testing and model calibration under uncertainty", *Computer methods in applied mechanics and engineering*, Vol. 197, No. 29/32, pp. 2467-2479.
22. Smith, E. P. and Rose, K. A. (1995), "Model goodness-of-fit analysis using regression and related techniques", *Ecological modelling*, Vol. 77, No. 1, pp. 49-64.
23. Tan, P., Steinbach, M., and Kumar, V. (2006), *Introduction to data mining*, Pearson Addison Wesley, pp. 294-306.
24. Vasermanis, E. K., Nechval, K. N., and Nechval, N. A. (2003), "Statistical validation of simulation models of observable systems", *Kybernetes*, Vol. 32, No. 5/6, pp. 858-869.
25. Wei, X. C. and Li, E. P. (2006), "Reflection of transmitting antenna in reverberation chamber and its effect on chi-square validation", *Antennas and propagation society international symposium 2006*, pp. 3573-3576.
26. Witten, I. H. and Frank, E. (2005), *Data Mining : Practical Machine Learning Tools and Techniques 2nd Ed.*, Morgan Kaufmann series in data management systems, pp. 161-176.
27. Zhu, Q., Gelenbe, E., and Qiao, Y. (2008), "Adaptive prefetching algorithm in disk controllers", *Performance evaluation*, Vol. 65, No. 5, pp. 382-395.



**황 보 훈** (gnrkd1qk@korea.ac.kr)

2008 고려대학교 산업시스템정보공학과 학사  
2008~현재 고려대학교 정보경영공학부 석사 과정

관심분야 : Discrete Event Simulation, Management Science



**천 현 재** (slash@korea.ac.kr)

1997 인천대학교 산업공학과 학사  
1999 고려대학교 산업공학과 석사  
2006 고려대학교 산업공학과 박사  
2009~현재 고려대학교 정보보호연구원 연구교수

관심분야 : SCM, 데이터마이닝, Discrete Event Simulation



**이 흥 철** (hcllee@korea.ac.kr)

1983 고려대학교 산업공학과 학사  
1988 Univ. of Texas 산업공학과 석사  
1993 Texas A&M Univ. 산업공학과 공학박사  
현재 고려대학교 정보경영공학부 교수

관심분야 : SCM, 생산 및 물류 정보시스템, PLM