

실세계 시스템의 퍼지 시뮬레이션에 관한 연구

A study on the fuzzy simulation for real world system

이은순 · 김성옥

Eun-Soon Lee · Sung-Ok Kim

Abstract

Fuzzy simulation predicts the behaviors of real system based on a model by qualitative reasoning methods and simulates the representation of ambiguous values on the real system variables using the theory of fuzzy sets. During the simulation, however, unnecessary behaviors due to the fuzzy representation are created, and the number of states of system variables changing temporally in the time axis is drastically increased.

In this paper, we present a new algorithm which eliminates the spurious behaviors from the great number of result values due to the results of the fuzzy operation, and reduces the number of the states by transforming the complex state transition rules. This paper also shows the easy implementation of the simulation by using the existing package while it is difficult on the PC due to the complexities of the calculation.

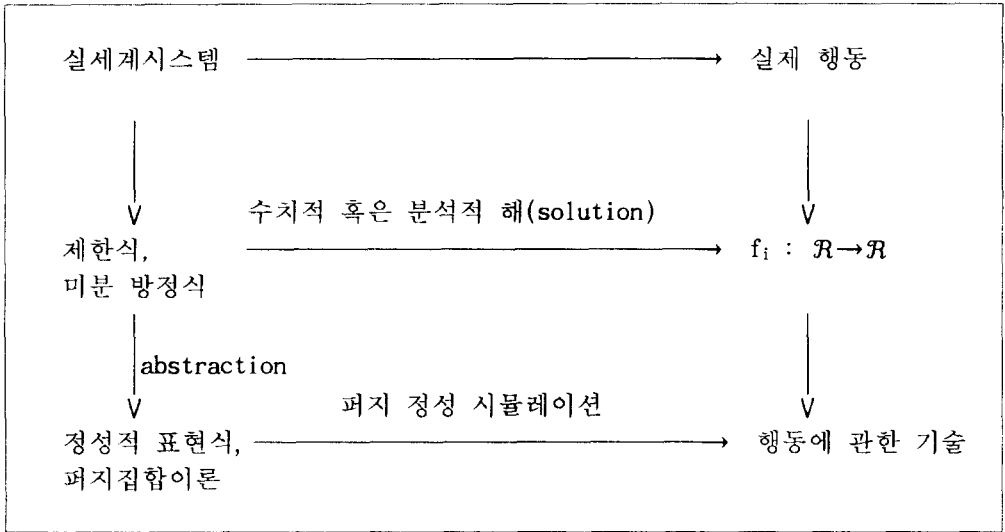
1. 서론

퍼지 정성 시뮬레이션(Fuzzy Qualitative Simulation: 이하 퍼지 시뮬레이션으로 표시)은 모델을 기초로 하여 실세계 시스템의 행동들에 관하여 정성적(qualitative) 추론 방법으로 행동들을 예측하고, 실제 시스템의 표현에 있어서 시스템 변수가 갖는 애매한 값들을 퍼지 집합 이론을 이용하여 시뮬레이션을 수행한다. 퍼지 모델을 기초로 한 진단 시스템에 활용될 수도 있으며 과학, 기술 등 여러 분야에서 적용될 수 있다[9][13][14][17]. 특별히 정성적 모델링과 퍼지 시뮬레이션은 계수 값들을 완전히 알지 못하는 복잡한 기계장치의 행동들을 예측하는데 아주 유용하게 사용된다. 즉, 수학적, 혹

은 분석적으로 모델링하기 어려운 복잡한 기계장치에 대하여 모니터링할 수 있고, 잘 알려지지 않은 불완전한 상태의 장치에 대한 진단시스템에도 유용하게 사용될 수 있다. 정성적 추론에 관한 연구는 시스템 행동의 정성적 기술을 위하여 절충적 기법들로 유도되는데 이러한 연구는 오늘날 기계공학적 시스템에서 발견되는 복잡성을 극복하는 방법을 제공[15]하고 여러 응용 분야에 확장하여 자동화 시스템의 기초가 되고 있다. 특별히 정성적 시뮬레이션 알고리즘은 시스템이 외부 동요에 의해 주관되거나 혹은 시스템에 대하여 초기에 영향을 주는 시스템 요소들을 추론하기 위한 기법들을 설명하는데 사용된다[4]. 그러나 이러한 정성적 추론 기법은 실제의 시스템에 응용했을 때 정성적 개념 자

체가 갖는 애매함때문에 시간에 따라 끊임없이 변화하는 실제 상황의 여러 가지 변화를 잘 나타내 주지 못한다[15].

서 많은 경우의 상태로 전이할 경우 매우 복잡하고 지루한 계산과정을 반복해야만 한다. 본 연구에서는 이러한 가능한 상태전이에서 상태수를 줄이고 계산상의



〈그림 1〉 실제행동에 대한 퍼지 정성 시뮬레이션과 상태방정식에 의한 추상화

〈그림 1〉에서는 시스템 상태 변수들의 집합과 관계식들에 의하여 실세계시스템을 미분방정식의 형태로 표현해 주고, 그 방정식에 의한 해는 시간에 따라서 시스템의 행동을 표현해 주는 하나의 함수형태로 표현하고 있으며 퍼지 정성적 표현기법에 의하여 시스템의 실제 행동을 기술해 준다[3][4]. 퍼지 집합 이론에 의한 시뮬레이션은 시스템의 주어진 구조적 기술과 초기 조건들에 의하여 여러 가지 가능한 행동이 예측된다. 그러나 이러한 여러 가지 예측된 행동들에는 퍼지 연산의 결과에 의하여 실제 값의 범위를 벗어나는, 즉 실제 시스템의 행동과는 무관한 행동들도 포함된다. 이러한 불필요한 행동들을 제거하기 위하여 D.Waltz[5] 알고리즘이 사용되었다.

본 연구에서는 첫째로 위에서 언급한 바와 같이 퍼지연산의 결과로 인한 수많은 결과값들 중에서 불필요한 행동을 제거하고 유일한 다음 상태로 전이하기 위하여 각각의 변수들에 대한 변수들간의 관계를 표현한 제한식, 미분방정식, 함수적 관계식들에 대하여 변수들간의 퍼지연산이 수행되어야 한다. 때문에 상태전이에

비효율성을 극복하기 위하여 상태전이를 하는 과정에서 이전의 퍼지값과 다음상태의 퍼지값에서 중복되는 부분을 제외시킨 값으로만 전이를 시켜 반복계산 과정을 훨씬 줄일 수 있도록 하였다. 또한 퍼지 시뮬레이션을 수행하는 과정은 각각의 과정마다 매우 복잡한 반복계산을 수행하므로 PC환경에서의 시뮬레이션 실행시간상의 문제가 많았다. 이 논문에서는 이를 단계적으로 구분하여 퍼스널컴퓨터 환경에서 스프레드 시트 패키지인 엑셀[2]을 이용하여 시뮬레이션하였다.

본 논문의 구성은 2장에서는 퍼지 시뮬레이션에 시스템 변수의 값을 표현하기 위한 퍼지 정량 공간(fuzzy quantity space)을 설명하였고, 3장에서는 기존의 연구 방법들의 구현에 있어서 발생하는 복잡한 계산과정을 단순화하기 위하여 상태 전이 규칙과 상태도착시간 및 상태지속시간을 구하기 위하여 좀 더 간단한 방법을 제시하였고, 4장에서는 사례를 들어 제안한 방법으로 시뮬레이션을 수행하여 보았다. 마지막으로 5장에서는 결론 및 향후과제에 대하여 살펴본다.

2. 퍼지 정성 시뮬레이션

2.1 정성적 모델링(qualitative modeling)과 정량공간(quantity space)

“정성적(qualitative)”이란 개념은 여러 가지 방법으로 설명될 수 있으나 일반적으로 “비수치적(non-numerical)”인 개념을 말한다. 그러므로 “정성적 모델링(qualitative modeling)”은 관련된 시스템 변수들이 갖는 값을 추상화시키고, 시간에 따라 연속적으로 변하는 시스템 변수들이 계속적으로 이 추상적인 값을 갖도록 하는 것이다. 대부분의 정성적 모델링에서 실제값의 적절한 추상화 작업은 추론할 시스템의 행동에 대하여 중요한 특징이나 경계(landmarks)를 포함해야 한다. 이것은 유한한 특징들에 대하여 실수선상의 값들로의 정량화(quantization)가 필요하며, 이러한 값을 정량공간(quantity space)라고 한다[11]. 즉, 변수들이 가질 수 있는 값들의 집합을 포함한 공간을 말하며, 각각의 정량공간은 본래의 시스템이 갖는 값을 추상화시키는 과정에서 포함되지 않는 경우도 있어, 시스템을 가장 정확하고 올바르게 표현하기 위하여 정량공간을 정하는 일은 아주 중요하다[6][7]. 또한 복잡한 실세계 시스템의 표현에 있어서 애매한 표현을 나타내기 위하여 퍼지집합[1]을 이용한 정량공간을 사용한다.

실세계 시스템을 정성적 기법으로 표현할 때 주어진 t 시간 동안에 시스템 상태변수의 정성적 상태를 다음과 같이 한 쌍(크기값, 변화율)의 값으로 표현한다. 즉,

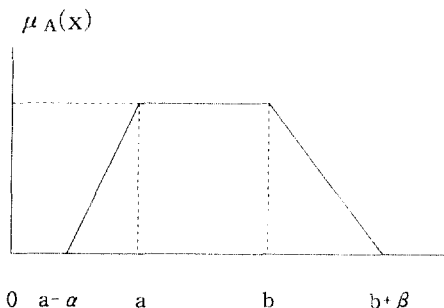
▶ 크기 값(amount) : 시스템 변수의 값

▶ 변화율(derivative) : 주어진 시간동안에 시스템 변수가 갖는 크기값의 변화율

이들 두 가지 요소가 갖는 값의 정성적 값은 퍼지 정량 공간(fuzzy quantity space)값으로 표현되며, 퍼지 정량 공간은 부분적인 순서를 갖는 숫자들의 작은 집합이다[3]. 퍼지 정량 공간 값으로 표현된 모델은 실세계의 일반 상식적인 이해가 가능하고, 단순하며, 또한 컴퓨터로의 구현을 위하여 효율적이어야 한다[17]. 퍼지 정량 공간을 나타내는 퍼지숫자는 계산상의 복잡함을 극복하기 위하여 사각 퍼지 숫자를 사용한다[1]. 퍼지 정량 공간은 <그림 2>와 같이 표현되며 그의 소속함수는 다음과 같이 정의한다.

2.2 상태 전이 규칙과 시스템 상태의 정성적 표현

상태전이 규칙에 의하여 각 시스템 변수들에 대한 정성적 상태 전이(transition)가 일어남으로 시스템의 행동(behaviors)이 결정되는데, 여기에서 현재 상태의 변수의 크기값과 변화율을 (M_i, R_i), 다음상태의 변수의 크기값을 (M_j, R_j)로 하여 표현하며, 시스템 변수들의 상태는 퍼지숫자로 표현이 되고 변수의 상태가 갖는 값이 일시적 순간에 변수값과 변화율의 변동으로 인한 다음 일시적 순간으로의 상태 전이가 일어나게 된다. 다음과 같이 퍼지 시뮬레이션[15]의 방법에서는 상태 전이를 현재상태의 변화율에 따라 변화율이 0보다 크거나 작은 경우와 0과 같은 경우, 즉 변화율의 변화가 있는 경우와 변화율의 변화가 없는 경우로 나누어 크기값의 변화를 유도하였다. 또한 다음상태의 크기값의



$$\mu_A(x) = \begin{cases} 0 & x < a - \alpha \\ \alpha^{-1}(x - a + \alpha) & x \in [a - \alpha, a] \\ 1 & x \in [a, b] \\ \beta^{-1}(b + \beta - x) & x \in [b, b + \beta] \\ 0 & x > b + \beta \end{cases}$$

<그림 2> 정규블록퍼지숫자의 표현

전이를 위하여 크기값이 실수값(고정된 실수값)을 갖는 경우와 퍼지값을 갖는 경우와 구분하여 다음상태의 크기값의 범위를 현재상태의 크기값을 포함한 다음크기의 퍼지값을 선택함으로써 인하여 다음상태로의 값의 범위가 더욱 커지는 결과를 가져오게 된다.

※ 가능한 상태 전이 규칙

- 규칙① If $R_1 >_{\alpha} 0$ ($R_1 <_{\alpha} 0$), then
 if $M_1 \in R$, then $M_2 >_{\alpha} M_1$ ($M_2 <_{\alpha} M_1$),
 else $M_2 \geq_{\alpha} M_1$ ($M_2 \leq_{\alpha} M_1$).
- 규칙② If $R_1 = 0$, then
 if $M_1 \in R$, then
 if $M_2 >_{\alpha} M_1$ ($M_2 <_{\alpha} M_1$, or $M_2 = M_1$),
 then $R_2 >_{\alpha} 0$ ($R_2 <_{\alpha} 0$, or $R_2 = 0$);
 else
 if $M_2 \geq_{\alpha} M_1$ ($M_2 \leq_{\alpha} M_1$),
 then $R_2 >_{\alpha} 0$ ($R_2 <_{\alpha} 0$),
 if $M_2 = M_1$,
 then $R_2 \in \{ 0, X, Y \}$.
- (여기에서 X, Y는 0의 α adjacent qualitative values이다)

시뮬레이션 하고자 하는 대상 시스템의 상태표현은 실제 시스템의 상태방정식을 추상화시킨 시스템의 행동을 기술해 주는 변수들간의 관계를 나타내는 정성적 제한 방정식(qualitative constraint equations)에 의하여 설명될 수 있다[10]. 시뮬레이션은 이들 변수에 시스템의 행동들을 적용하여 시뮬레이션 하게 되며, 이들 정성적 제한 방정식들은 여러 가지 가능한 정성적 행동들을 제한해 준다[16]. 이들 정성적 제한 방정식들은 다음 세 가지 유형으로 한다.

(1) 산술적 제한식(arithmetic constraints)

시스템 상태 방정식에 의한 매개변수들간의 관계를 표현할 때 사용되는 기본적인 산술 연산자는 아래와 같으며, 예를들어 시스템 매개변수 x, y, z가 있을 때 $Q(x)$, $Q(y)$, $Q(z)$ 는 퍼지 정량 공간에서의 값을 나타내며,

$$Q(z) = f(Q(x), Q(y)), \quad Q(x), Q(y), Q(z) \in Q_F$$

(f는 퍼지 연산자+, -, ×, ÷ Q_F 는 퍼지정량공간)

이들의 연산은 변수들의 값이 퍼지숫자로 표현되기 때

문에 일반적인 퍼지연산의 규칙에 따르기로 한다. 또한 퍼지 숫자의 연산에서 결과값이 하나의 퍼지 정량 공간값으로 결정되지 않을 때 근사 원리(approximation principles)[1]를 이용하여 두 퍼지 숫자 값의 근사값을 사용한다.

(2) 미분 제한식(derivative constraints)

미분 연산자는 동적 시스템(dynamic system)의 행동을 기술하는데 있어서 순간적인 시스템의 행동을 정의하기 위하여 꼭 필요하다. 퍼지 시뮬레이션에서 정성적 미분 제한식의 형태는 다음과 같으며

$$Q(y) = derivQ(x), \quad Q(x), Q(y) \in Q_F$$

이의 연산은 퍼지 정량 공간값으로 표현되어야 하고, 변수 y의 크기값은 변수 x의 변화율의 값과 같음을 의미한다. 이 연산은 새로운 값의 계산을 위함이지 아니고 퍼지 시뮬레이션을 수행하는 동안 주어진 변수값들의 일관성을 검사하는 값으로 사용된다.

(3) 정성적 함수 제한식(qualitative function constraints)

퍼지 정량 공간값을 사용하는 퍼지 시뮬레이션은 퍼지 논리 관계(fuzzy logical relation)로써 정성적 함수 관계를 표현할 수 있다. 즉 제한을 주는(constraining) 변수와 제한을 받는(constrained) 변수 사이의 관계를 퍼지 논리 규칙(fuzzy logical rules)으로 나타낼 수 있다. 즉 두 변수들간에 원인-결과와 관계가 존재하는 경우, 임의의 제한을 주는 변수값(constraining)이 또 다른 제한을 받는 변수값(constrained)과 관계가 성립하면 1, 그렇지 않으면 0으로 표시한다.

3. 상태 전이와 필터링

3.1 상태전이 규칙의 단순화

기존의 연구 방법에서 시뮬레이션의 실행은 방대한 양의 계산 과정과 시스템의 값이 변하는 순간마다 많은 가능한 상태전이들에 대하여 매순간 올바른 상태인가를 검사해야 하기 때문에 워크스테이션(SUN, Prolog

Language)이상의 컴퓨터에서 실행이 가능하였다. 예를 들면 시스템 상태 변수가 3개인 경우 각각의 변수들에 대한 가능한 상태전이가 각각의 변수에 대하여 3개, 4개, 3개의 경우로 전이한다고 가정하면, 주어진 제한식을 모두 만족하는 유일한 하나의 상태로 전이하려면 하나의 제한식에 대하여 최소한 $3 \times 4 \times 3$ (번) 비교연산을 수행하여야 각각의 제한식을 모두 만족하는 하나의 상태로의 전이가 이루어지게 된다. 따라서 이들 각각의 상태들에 대한 전이의 결과가 유일한 하나의 상태로 전이하고 복잡한 시뮬레이션 수행을 단계적으로 쉽게 시뮬레이션하기 위한 가능한 상태전이 규칙이 필요하다. 기존의 연구 방법에서는 현재상태에서 다음상태로의 전이가 일어날 때 현재상태의 크기값이 퍼지값을 갖는 경우 다음상태의 크기값은 현재상태의 크기값을 포함한 다음 크기의 퍼지값에 대하여 상태 전이를 하게 하였다. 그러나 현재상태의 크기값에 대한 변화율의 값이 0이 아닌 경우 조금이라도 변화가 일어난 경우이기 때문에 다음상태의 크기값에 현재상태의 크기값을 포함시켜 전이가 일어나지 않는다. 따라서 이런 이유로 현재상태에서 다음상태로의 전이하는 과정에서 필터링의 횟수를 기존의 방법보다 반 이상 줄일 수 있는 수정된 상태전이 규칙을 제안한다. 상태전이 규칙을 단순화한 수정된 상태전이 규칙은 다음과 같다.

※ 수정된 상태 전이 규칙

규칙① If $R_1 >_s 0$ ($R_1 <_s 0$), then

if $M_2 \geq_s M_1$ ($M_2 \leq_s M_1$),
then $R_2 >_s R_1$, $R_2 <_s R_1$, $R_2 = R_1$

규칙② else $R_1 = 0$, then

if $M_2 >_s M_1$ ($M_2 <_s M_1$, or $M_2 = M_1$),
then $R_2 >_s 0$ ($R_2 <_s 0$, or $R_2 = 0$);

예를들면 임의의 시스템이 갖는 퍼지 정량공간이 $(-t, 0.6, -s, 0, s, 0.6, t)$ 라고 할 때 임의의 상태에서 다음상태로의 전이는 수정된 상태전이규칙에 의하여 다음과 같이 전개된다.

t_1 에서 다음상태 t_2 로의 상태전이가 <표 2>에서 보는 바와 같이 <표 1>의 기존의 알고리즘에 의한 방법보다 경우의 수가 많이 감소되었음을 알 수 있으며, 값이 갖는 퍼지 정량 공간에서의 범위도 같은 범위에 속하게 됨을 알 수 있다. 결국은 기존의 방법에서는 불필요한

<표 1> 기존 알고리즘에 의한 상태전이

	t_0	t_1	t_2
x	(0.6,0)	(t,s) (s,-s) (0.6,0)	(s,-0.6) (s,-s) (s,0) (0,-0.6) (0,-s) (0,0)
v	(0,-0.6)	(-s,-s) (-s,-t) (-s,-0.6)	(-s,-0.6) (-s,-s) (-s,0) (-0.6,-0.6) (-0.6,-s) (-0.6,0)
a	(-0.6,0)	(-s,s) (-t,-s) (-0.6,0)	(-s,0) (-s,s) (-s,0.6) (0,0) (0,s) (0,0.6)

<표 2> 수정된 알고리즘에 의한 상태전이

	t_0	t_1	t_2
x	(0.6,0)	(t,s) (s,-s) (0.6,0)	(0,-0.6) (0,-s) (0,0)
v	(0,-0.6)	(-s,-s) (-s,-t) (-s,-0.6)	(-0.6,-0.6) (-0.6,-s) (-0.6,0)
a	(-0.6,0)	(-s,s) (-t,-s) (-0.6,0)	(0,0) (0,s) (0,0.6)

※ 현재상태에서 다음상태로의 상태전이(t_0 에서 t_1 으로)

① 변화율이 0인 경우

$x : (0.6,0) \rightarrow (t,s)$ (0.6을 기준으로 다음 크기의 값)
(s,-s) (" " 이전 크기의 값)
(0.6,0) (0.6) (" ")

② 변화율이 0이 아닌 경우

$v : (0,-0.6) \rightarrow (-s,-s)$ (변화율이 0보다 작기 때문에 크기값은 -s)
(-s,-t) (" " " ")
(-s,-0.6) (" " " ")

범위의 값까지 포함하여 상태 전이를 하였기 때문에 훨씬 많은 경우의 상태 전이들에 대하여 반복계산을

함으로 계산과정이 복잡하게 되고 필터링을 위하여 더 많은 노력을 한 것을 알 수 있다. 위의 비교표는 수정된 상태전이 규칙에 의하여 전이가 일어나며, 예를 들면 현재상태에서 ① 변화가 일어나지 않은 경우(if $R_i = 0$)에는 다음상태의 크기값 M_2 은 현재상태의 크기값을 기준으로 큰 경우, 작은 경우, 같은 경우로 값을 전개하고 그에 대한 다음 상태의 변화율 R_2 은 역시 크기값의 경우와 같이 변화율의 값을 큰 경우, 작은 경우, 같은 경우로 전개하였다. 다음으로 ② 변화가 일어난 경우(if $R_i \neq 0$)에는 현재상태의 변화율의 방향에 따라(양수 혹은 음수) 다음상태의 크기값 M_2 을 변화율이 양수이면 다음 크기의 값을, 변화율이 음수이면 이전 크기의 값으로 전개하고, 다음상태의 변화율 R_2 은 ①의 경우와 같이 전개한다.

3.2 상태 지속 시간과 도착시간

주어진 정성적 상태에서 하나의 변수가 그 상태에 머무르는 시간을 상태지속시간(persistence time: ΔT_p)이라 하고 이 시간도 퍼지숫자로 표현한다. 하나의 시스템 변수 x 에 대하여 상태전이는 $QS(x, \Delta T_p) = (M_1, R_1)$ 에서 상태 $QS(x, \Delta T_p) = (M_2, R_2)$ 로의 전이를 말하며, 상태 지속 시간은 아래와 같이 계산한다.

$$M = [\rho_1, \rho_2, \rho_3, \rho_4], \quad R = [q_1, q_2, q_3, q_4]$$

(1) 만일 변화율(R_{α})이 0이 아니면

$$\Delta T_p \in \alpha \{ W(M) / |R_{\alpha}| \}$$

$$\in [\rho_2 - \rho_1 + (1 - \alpha)(\rho_3 + \rho_4)] / |R_{\alpha}|$$

$$|R_{\alpha}| = ABS(q_2 - q_1 + (1 - \alpha)(q_3 + q_4))$$

$W(\cdot)$: 퍼지 정성 값(퍼지 정량 공간)의 α -cut에서의 넓이 (폭)

R_{α} : R의 α -cut

(2) 만일 변화율이 0이면

ΔT_p 는 정의되지 않음

(왜냐하면, 상태변화가 없었으므로)

(1)식에서 $|R_{\alpha}|$ 의 값은 기존의 연구 방법[15]으로는 시간구간의 각 구간값이 초기상태부터 마지막상태까지 전체구간의 시간값을 모두 포함하는 결과가 된다. 따라서 이를 보완하여 크기값 M 과 같이 $|R_{\alpha}|$ 의 값을 정량화 된 퍼지값을 갖게 하였다.

시스템 변수가 하나의 상태에서 다음상태로 전이할 때 걸리는 시간을 도착시간(arrival time)이라 하고 두

퍼지숫자의 교차점(u, v)을 구하여 교차점의 v 값과 임계값 α 와 비교하여 도착시간을 구한다. 퍼지숫자 $M_1 = [p_1, p_2, p_3, p_4], M_2 = [q_1, q_2, q_3, q_4]$ 일 때 교차점과 도착시간은 다음 식과 같다.

$$\bullet \text{ 교차점}(u, v) : \begin{cases} u = (q_1 \times \rho_1 + \rho_2 \times q_3) / (q_3 + \rho_1), \\ v = -(q_1 - \rho_2 - q_3 - \rho_4) / (q_3 + \rho_1). \end{cases}$$

• 도착시간(ΔT_{α}) :

(1) null-전이나 혹은 변화율-전이, 즉 크기값의 변화가 없는 상태전이

$$\text{도착시간}(\Delta T_{\alpha}) = 0$$

(2) 크기값-전이 혹은 크기값-변화율-전이, 즉 크기값의 변화가 있는 상태전이

$$\text{if } \alpha \leq v \text{ then } \text{도착시간}(\Delta T_{\alpha}) = 0$$

else

$$\Delta T_{\alpha} \in (\Delta(M_1, M_2))_{\alpha} / |R_{\alpha}|$$

$$(\Delta(M_1, M_2))_{\alpha} = q_1 - \rho_2 + (\alpha - 1)(q_3 + \rho_4)$$

(여기에서 $(\Delta(M_1, M_2))_{\alpha}$ 는 M_1, M_2 퍼지값의 최단거리로 계산함)

$$(|R_{\alpha}|)_{\alpha} = ABS(R_2 - R_1)$$

(2)식에서 $|R_{\alpha}|$ 의 값은 상태 지속 시간과 같이 계산하고, 단 도착시간은 이전상태에서 현재상태로의 변화이기 때문에 현재상태의 변화율값에서 이전 상태의 변화율값을 뺀 값으로 $|R_{\alpha}|$ 을 계산하였다. 참고로 기존의 연구방법과 제안한 방법에서의 도착시간과 상태 지속 시간을 계산하여 <표 3>에서 비교하였다. 제안한 방법에 의하여 시간을 전개해 보면 다음과 같으며,

$$t_0 : 0$$

$$t_1 = t_0 + \Delta T_{\alpha 1} + \Delta T_{p1} = 0 + 1.1 + 1 = 2.1$$

$$t_2 = t_1 + \Delta T_{\alpha 2} + \Delta T_{p2} = 2.1 + 1.1 + 0 = 3.2$$

$$t_3 = t_2 + \Delta T_{\alpha 3} + \Delta T_{p3} = 3.2 + 1.1 + 1 = 5.3$$

$$t_4 = t_3 + \Delta T_{\alpha 4} + \Delta T_{p4} = 5.3 + 1.1 + 0 = 6.4$$

$$t_5 = t_4 + \Delta T_{\alpha 5} + \Delta T_{p5} = 6.4 + 0.1 + 1 = 7.5$$

$$t_6 = t_5 + \Delta T_{\alpha 6} + \Delta T_{p6} = 7.5 + 0.1 + 0 = 7.6$$

$$t_7 = t_6 + \Delta T_{\alpha 7} + \Delta T_{p7} = 7.6 + 0.1 + 1 = 8.7$$

$$t_8 = t_7 + \Delta T_{\alpha 8} + \Delta T_{p8} = 8.7 + 0.1 + 0 = 8.8$$

기존의 방법에 의한 시간구간은 아래와 같이 구해진다. 각각의 시간구간을 비교해 보면 제안한 방법에 의한 시간값이 기존의 방법(아래)에 의한 시간구간값에 포함됨을 알 수 있으며, 반면에 기존의 방법에 의한 시간구간값은 각 구간마다 필요 이상의 넓은 범위의 값을 갖게 됨을 볼 수 있다. 따라서 시스템 마지막 상태

〈표 3〉 변화율에 의한 상태지속시간과 도착시간

시간주기	변수	수정된 방법에 의한 시간구간(Width(R _i))		기존의 방법에 의한 시간구간(R _i)	
		도착시간 (ΔT_a)	상태지속시간 (ΔT_p)	도착시간 (ΔT_a)	상태지속시간 (ΔT_p)
$t_0 \Rightarrow t_1$	x	1.1	1	[1, 11]	[0.91, 10]
	v	1.1	1	[1, 11]	[0.91, 10]
	a	0.1	1	{0.091, 10}	[0.91, 10]
$t_1 \Rightarrow t_2$	x	1.1	0	[0.916667, 0.916667]	[0, 0]
	v	1.1	0	overflow	overflow
	a	0.1	0	overflow	overflow
		0.05	0	{0.091, 1}	{0, 0}
		0.1	0	[0.0833, 0.0833]	[0, 0]
$t_2 \Rightarrow t_3$	x	1.1	1	[1, 11]	[0.909, 10]
	v	0.1	1	{0.0909, 1}	[0.909, 10]
	a	0.1	1	[0.0909, 1]	[0.909, 10]
		0.05	0.5	{0.0909, 1}	[0.909, 10]
		0.1	1	{0.0909, 1}	[0.909, 10]
$t_3 \Rightarrow t_4$	x	1.1	0	overflow	overflow
	v	0.1	0	[0.0833, 0.0833]	[0, 0]
	a	0.1	0	overflow	overflow
$t_4 \Rightarrow t_5$	x	0.1	1	[0.0909, 1]	[0.909, 10]
	v	0.1	1	[0.0909, 1]	[0.909, 10]
	a	1.1	1	[1, 11]	[0.909, 10]
$t_5 \Rightarrow t_6$	x	0.1	0	[0.0833, 0.0833]	[0, 0]
	v	0.1	0	overflow	overflow
	a	1.1	0	[0.916667, 0.916667]	{0, 0}
		0.55	0	[1, 11]	[0, 0]
		1.1	0	overflow	overflow
$t_6 \Rightarrow t_7$	x	0.1	1	{0.0909, 1}	[0.909, 10]
	v	1.1	1	[1, 11]	{0.909, 10}
	a	1.1	1	[1, 11]	[0.909, 10]
		0.55	0.5	[1, 11]	[0.909, 10]
		1.1	1	{[1, 11]}	[0.909, 10]
$t_7 \Rightarrow t_8$	x	0.1	0	overflow	overflow
	v	1.1	0	{0.916667, 0.916667}	[0, 0]
	a	1.1	0	overflow	overflow

의 시간구간값이 초기상태로부터의 시간구간값을 거의 포함하게 된다. 이는 퍼지연산에 의한 결과로 결과값이 많은 범위의 값을 포함하게 되기 때문이다. 따라서 변수의 변화율값을 크기값과 같이 퍼지 정량 공간의 α -cut에서의 넓이(width)값을 사용함으로써 시간구간을

좀 더 실제적으로 의미있게 구현하였다.

$$t_0 : 0$$

$$t_1 = t_0 + \Delta T_{a1} + \Delta T_{p1} \in [1, 11]$$

$$t_2 = t_1 + \Delta T_{a2} + \Delta T_{p2} \in [1.09, 12]$$

$$t_3 = t_2 + \Delta T_{a_3} + \Delta T_{p_3} \in [2.09, 23]$$

$$t_4 = t_3 + \Delta T_{a_4} + \Delta T_{p_4} \in [2.18, 24]$$

$$t_5 = t_4 + \Delta T_{a_5} + \Delta T_{p_5} \in [3.18, 35]$$

$$t_6 = t_5 + \Delta T_{a_6} + \Delta T_{p_6} \in [3.27, 36]$$

$$t_7 = t_6 + \Delta T_{a_7} + \Delta T_{p_7} \in [4.27, 47]$$

$$t_8 = t_7 + \Delta T_{a_8} + \Delta T_{p_8} \in [4.36, 48]$$

위의 시간구간 진행은 <표 3>의 변수 x에 관하여 전개하였으며 4장의 예를 통하여 각 시스템의 시간에 따른 변수의 상태변화를 (그림 3)에서 보여준다.

3.3 제한식의 필터링

시스템 변수들이 갖는 값들의 범위는 정성적 퍼지 정량 공간에서의 값을 갖고 이의 퍼지연산을 수행함에 따라 계산된 값의 범위는 실제 값의 연산의 범위보다 넓은 범위의 값을 갖게 된다. 따라서 이의 가능한 유일한 상태로의 전이를 위하여 계산된 값을 필터링하여야 하는데 그 방법은 Waltz 알고리즘[5][8]에 따르며 다음과 같다.

```

procedure REVISE(Q(X1, ... , Xn))
  begin CHANGED ← 0
  for each argument Xi do
    begin S ← REFINED(Q, Xi)
    if S = 0 then halt /* the original constraints were inconsistent */
    else if S ≠ Si then
      begin Si ← S;
      add Xi to CHANGED
    end
  end
  return CHANGED
end

procedure WALTZ
  begin QUE ← a queue of all constraints
  while QUE ≠ 0 do
    begin remove constraints Q from QUE;
    CHANGED ← REVISE(Q)
    for each Xi in CHANGED do
      for each constraints Q' ≠ Q which has Xi in its domain do
        add Q' to QUE
    end
  end
end
  
```

입의 제한식(산술 혹은 미분)을 Q(x)라 하고, 제한식 Q(x)에 포함되어 있는 변수들을 X₁, X₂, ..., X_k . 어

떤 변수 X_i에 대한 퍼지 정량 공간값들의 집합을 S_i 라 할 때 다음과 같은 과정에 의하여 각각의 시스템 변수에 대한 정성적 값이 퍼지 정량 공간에서 값을 가지게 된다.

4. 시뮬레이션의 실행

기존의 연구 방법에서 시뮬레이션의 실행은 방대한 양의 계산 과정과 시스템의 값이 변하는 순간마다 많은 가능한 상태전이들에 대하여 매순간 올바른 상태인가를 검사해야 하기 때문에 워크스테이션(SUN, Prolog Language)이상의 컴퓨터에서 실행이 가능하였다. 또한 퍼지 시뮬레이션 실행을 위한 프로그램의 구현이 용이하지 않고, 매 시간구간에서의 상태전이에 따른 시간 계산이 실시간으로 이루어져야 하는데 이를 일반적인 프로그램 언어(C언어 etc.)로 구현할 경우 실시간 프로그램을 구사하기가 매우 어렵고, 또 구현한다고 하더라도 하나의 상태전이를 위한 시간계산과 각각의 경우에 대한 필터링을 하여야 하는데 1분에서 2분까지의 시간이 걸리게 된다. 이러한 점을 고려하여 좀 더 쉬운 방법으로 퍼지 시뮬레이션을 실행시키기 위하여 실행 속도도 매우 빠르고 비주요한 개발 툴들이 많이 보급되어 있는 퍼스널 컴퓨터를 이용하여 시뮬레이션하는데, 스프레드 시트 패키지인 엑셀을 이용하여 시뮬레이션의 각 단계를 모듈별로 구분하여 실행하고 최후에 모듈들을 통합하여 결과를 유도하였다. 또한 엑셀에 의한 각 모듈별 계산 결과를 직접 그래프를 구현하는데 데이터로 사용할 수 있었다. 우선 프로그램 단계를 크게 4단계로 구성하였으며, 각 단계별로는 (1)초기 시스템 정의 과정 (2)가능한 상태 전이 과정 (3)제한식에 의한 필터링 과정 (4)시간진행에 의한 상태전이과정으로 구분하여 실행하였다.

4.1 예제(스프링에 달린 추의 시뮬레이션)

(1) 시스템 정의

스프링에 달린 추에 대한 행동을 퍼지 시뮬레이션을 이용하여 실행해 보고자 한다. 우선 시스템 변수 및 퍼지 정량 공간, 변수의 초기값, 제한식, 퍼지 논리 관계

표를 보면 아래와 같다.

① 시스템 변수

x : 스프링에 매달린 추의 위치로부터의 거리

v : 추의 속력

a : 추의 가속도

② 퍼지 정량 공간

$$Q_F = \{n_top, n_medium, n_small, zero, p_small, p_medium, p_top\}$$

$$= \{-1, -0.6, -s, 0, s, 0.6, t\}$$

$$= \{(-1, -0.7, 0, 0.1), (-0.6, -0.6, 0, 0), (-0.5, -0.1, 0.1, 0.1), (0, 0, 0, 0), [0.1, 0.5, 0.1, 0.1], [0.6, 0.6, 0, 0], [0.7, 1, 0.1, 0]\}$$

③ 변수들의 초기값

$$(x, v, a) = ((0.6, 0), (0, -0.6), (-0.6, 0))$$

④ 제한식

$$deriv\ x = v$$

$$deriv\ v = a$$

⑤ 퍼지 관계표

a~x	t	-0.6	-s	0	s	0.6	t
-1	0	0	0	0	0	1	1
-0.6	0	0	0	0	1	1	0
-s	0	0	0	0	1	1	0
0	0	0	0	1	0	0	0
0.6	0	1	1	0	0	0	0
s	0	1	1	0	0	0	0
t	1	1	0	0	0	0	0

(2) 상태 전이 규칙과 필터링에 의한 상태 전이

현재상태의 가능한 상태들의 집합에서 유일한 다음상태로의 전이는 예에서 주어진 제한식에 의하여 "x의

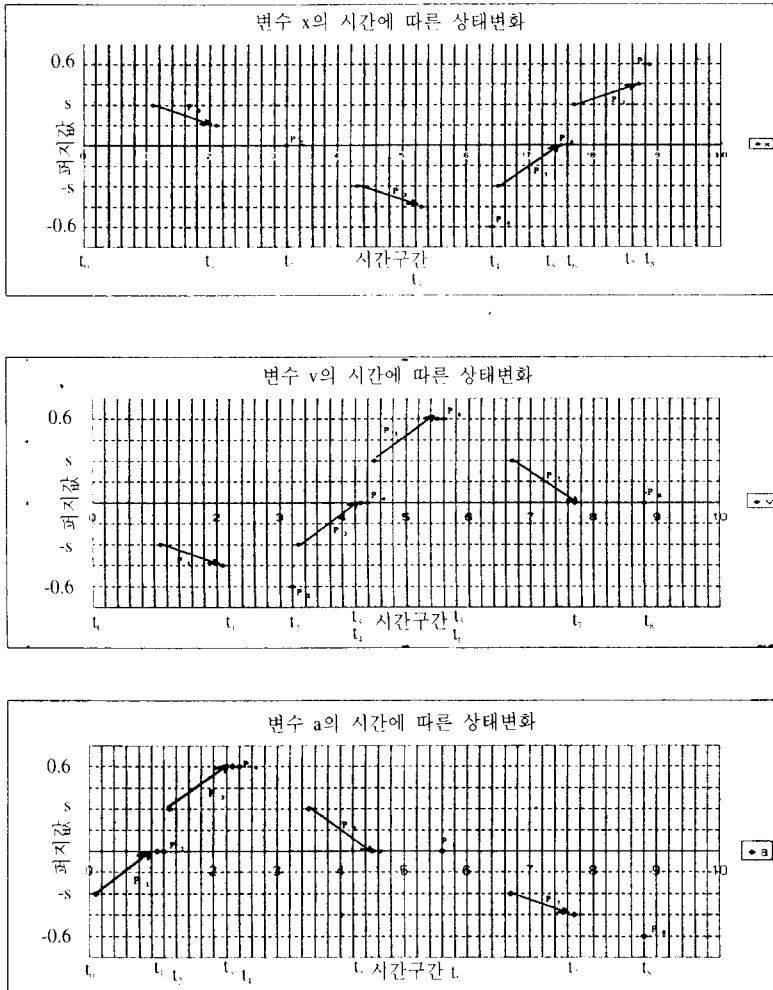
변화비 = v의 크기값"이어야 하므로 v의 크기값은 모두 -s이므로 x의 변화비가 -s가 아닌 상태는 모두 제거되므로 x의 상태는 (s,-s)만 남으며, a의 값에서 (-0.6,0)은 초기상태와 같은 상태가 전개되므로 제거되고 동시에 v의 (-s,-0.6)도 제거된다. 마지막으로 v의 (-s,-t), a의 (-t,-s)값은 v에 대한 두 개의 제한식에서 중복되므로 v에 대한 값은 (-s,-s), a에 대한 값은 (-s,s)로 결정된다. 이와 같이 각각의 가능한 상태들의 집합들에서 유일한 다음상태로 전이하기 위하여 시스템에서 정의된 여러 가지 제한식들을 사용하여 제한식을 만족하지 않는 경우를 제거하고 다음 상태를 결정한다. 마지막 t_k 상태에서는 각각의 변수가 처음의 초기상태와 같은 상태로 전개되기 때문에 더 이상 다음 상태를 전개하지 않는다[12]. 즉 초기상태로부터 8단계를 한 주기로하여 계속 반복하여 시스템이 전개됨을 알 수 있다. 단계별로 시뮬레이션한 결과를 <표 4>에 시간진행에 따라 표시하였다.

(3) 시간진행에 의한 상태전이

<표 3>과 <표 4>의 결과에 의하여 각 시간구간마다 결정된 시스템 변수의 상태전이를 (그림 3)에서 보았다. (그림 3)에서 p_1, p_2, \dots, p_k 는 각각의 시스템 변수가 임의의 시간구간내에서 임의의 퍼지값을 유지하며 다음상태로의 전이가 진행됨을 의미하며, 상태 지속 시간(p_1, p_2, \dots, p_k)들 사이의 간격은 현상태에서 다음상태에 도달하기 위한 도착 시간 간격을 의미한다.

<표 4> 시간 진행에 따른 상태전이

시간/변수	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
x	(0.6,0)	(t,s) (s,-s) (0.6,0)	(0,-0.6) (0,-s) (0,0)	(-s,-s) (-s,-t) (-s,-0.6)	(-0.6,-0.6) (-0.6,-s) (-0.6,0)	(-s,s) (-t,-s) (-0.6,0)	(0,0) (0,s) (0,0.6)	(s,s) (s,0.6) (s,t)	(0.6,0) (0.6,s) (0.6,0.6)
v	(0,-0.6)	(-s,-s) (-s,-t) (-s,-0.6)	(-0.6,-0.6) (-0.6,-s) (-0.6,0)	(-s,s) (-t,-s) (-0.6,0)	(0,0) (0,s) (0,0.6)	(s,s) (s,0.6) (s,t)	(0.6,0) (0.6,s) (0.6,0.6)	(t,s) (s,-s) (0.6,0)	(0,-0.6) (0,-s) (0,0)
a	(-0.6,0)	(-s,s) (-t,-s) (-0.6,0)	(0,0) (0,s) (0,0.6)	(s,0) (s,s) (s,0.6) (s,t)	(0.6,0) (0.6,s) (0.6,0.6)	(t,s) (s,-s) (0.6,0)	(0,-0.6) (0,-s) (0,0)	(-s,-s) (-s,-t) (-s,-0.6) (-s,0) (s,s) (0,0)	(-0.6,-0.6) (-0.6,-s) (-0.6,0)



〈그림 3〉 시스템 변수의 시간에 따른 상태변화

4. 결론 및 향후과제

실세계 시스템의 여러 가지 현상들에 대한 퍼지 시뮬레이션을 수행함에 있어서 기존의 방법들로는 퍼지 집합이론에 의한 연산으로 인하여 너무 많은 경우의 상태전이 발생하게 됨으로 이들의 연산 및 시뮬레이션 수행에 있어서 매우 복잡한 계산과정을 거치게 된다. 이러한 점을 극복하기 위하여 본 논문에서는 수정된 알고리즘을 사용하여 상태전이를 단순화 시킴으로써 상태수를 줄이고, 진행 시간구간을 좀 더 명확하게 구분하였다. 또한 이의 수행을 PC 환경에서 가능하도록 스포레드 쉬트 패키지를 이용하여 단계적으로 수행

하는 방법을 보이고, 또 실제로 PC에서 수행하여 보았다. 앞으로 더 보완해야 할 점으로는 모델링 과정이 좀 더 기계적으로 될 수 있도록 비주얼 SLAM II와 같은 모델링 도구의 개발이 필요하고 또 시간의 변화에 따라 그래프로 볼 수 있도록 하는 일도 의미가 있다고 생각한다.

참고문헌

- [1] 이광형, 오길록, "퍼지이론 및 응용", 1권, 2권, 흥릉과학출판사, 1991.
- [2] 신동준, "엑셀매크로", 기전연구소, 1997.

[3] B. Kuipers, "The limit of qualitative simulation", MIT Laboratory for Computer Science, Cambridge, 1985, pp.128-136.

[4] B.J.Kuipers, "Qualitative simulation", Artificial Intell., Vol.29, 1986, pp.289-338

[5] D.Waltz, "Understanding line drawings of scenes with shadows", The Psychology of Computer Vision, P. Winston Ed., New York: McGraw-Hill, 1975

[6] D.Weld, "Exaggeration", Proc. 7th Nat. Conf. Artificial Intell., 1988, pp.291-295

[7] D.Weld, "Exaggeration", Artificial Intell., Vol.43, 1990, pp.311-368.

[8] E.Davis, "Constraint propagation with interval labels", Artificial Intell., Vol.32, 1987, pp.281-331

[9] G.R.Guo and Q.Shen, "Fuzzy algorithm for error-correcting estimation of radar signal", Electron. Lett., Vol.26, No.16, 1990, pp.1321-1324.

[10] J.de Kleer and D.G.Bobrow, "Qualitative reasoning with high-order derivatives", Proc.3th Nat. Conf. Artificial Intell., 1984, pp.86-91

[11] P.Fouche and B.J.Kuipers, "Reasoning about energy in qualitative simulation", IEEE Trans. Syst. Man Cybern., Vol.22, 1992, pp.47-63

[12] P.Struss, "Global filters for qualitative behaviours", Proc. 7th Nat. Conf. Artificial Intell., 1988, pp. 275-279.

[13] Q.Shen, "Fuzzy intraframesmoothing of a Noisy image", Electronics Letters, Vol.26, No.13, 1990, pp.908-910

[14] Q.Shen and R.R.Leitch, "Application studies of fuzzy qualitative simulation", Mathematics of the Analysis and Design of Process Control, P.Borne and S.G. Tzafestas Eds., New York: Elsevier, 1992.

[15] Qiang Shen and Roy Leitch, "Fuzzy qualitative simulation", IEEE Trans. Syst. Man and Cybern., Vol. 23, No. 4, 1993, pp.1038-1062.

[16] R.G.Simmons "Common-sence arithmetic reasoning" Proc. 5th Nat. Conf. Artificial Intell. 1986, pp. 118-124

[17] R.R.Leitch and Q.Shen, "Model dimensions within fuzzy qualitative simulation", Proc. 13th IMACS World Congress on Communication and Applied Mathematics, 1991, Vol.3, pp.1212-1213, Dublin

● 저자소개 ●



김성욱

1966년

연세대학교 수학과(학사)

1976년

University of Minnesota 전자계산학과(석사)

1986년

연세대학교 수학과 (박사)

1983년~현재

한남대학교 컴퓨터공학과(교수)

관심분야

시뮬레이션, 수치해석, 교육공학



이은순

1983년

한남대학교 계산통계학과 (학사)

1991년

한남대학교 전자계산학과 (석사)

1996년

한남대학교 컴퓨터공학과 (박사과정 수료)

관심분야

시뮬레이션, 퍼지, 교육공학