

# 화학공정 동적모사기 개발에 있어서 시스템구조 및 전략

이강주\*, 한경택\*\*, 윤인섭\*

\* 서울대학교 화학공학과    \*\* 한양화학 중앙연구소

## System Architecture and Simulation Strategy for Dynamic Process Simulation

Kang Ju Lee\*, Kyung Taek Han\*\*, En Sup Yoon\*

\* Department of Chemical Engineering, Seoul National University

\*\* Hanyang Chemicals Central Research Center

### Abstract

This paper presents the simulation architecture and strategy for dynamic simulation of chemical process and describes key features of developed dynamic simulation system, MOSA (Multi-Objective Simulation Architecture).

A plant structure may be partitioned into several strong coupling units, called cluster. If this cluster is solved simultaneously, it is possible to simulate whole plant without introducing convergence problem of tear streams. In this study, a flexible modular approach based on clusters was proposed as a promising architecture for dynamic chemical process simulator.

### 1. 서론

공정의 설계, 최적화 문제 등에 있어서 정상상태 모사가 성공적으로 많이 이용되고 있다. 그러나 정상상태 모사만으로는 공정이 비평형 또는 비정상상태에 있을 때의 동특성에 대한 정보는 얻을 수가 없으며, 현대의 공정은 경제적인 이유 등으로 점차 복잡해짐에 따라 그 동특성을 예측하기라 더욱 어렵게 됨에 따라 제어 문제, 조업 문제 등의 공정의 동특성을 기본으로 하는 분야의 연구에는 많은 제약이 있었다.

동적모사는 공정의 동특성을 분석할 수 있는 비교적 값싼 방법으로 컴퓨터의 발달과 수치해석의 발전으로 이제는 가능한 수단이 되고 있다.

동적모사에 대한 연구는 주로 화학공정을 수학적으로 모델링하여 얻어지는 미분방정식을 효과적으로 적분하기 위한 수치해석적 측면과 실제로 이를 시스템으로 구현하기 위한 모사구조 및 전략에 대한 연구로 나누어져 행해져 왔다. 수치해석적인 측면에서는 stiff한 미분방정식계를 효과적으로 적분하기 위한 연구로서 Gear(1978)의 DIFSUB 코드를 기본으로 한 Multistep 방법이 성공적으로 적용되고 있다. 모사구조 및 전략은 크게 순차적 모듈방식과 방정식 중심방식으로 나눌 수 있는데, 정상상태 모사의 경우와 유사하다.

실제로 상업적 정상상태 모사기의 대부분은 순차모듈방식을 사용하고 있으나, 학문적 관심은 없어지고 있는 실정이다. 기본적으로 방정식 중심구조는 정상상태모사, 동적모사, 최적화 등의 다양한 기능을 모두 수행할 수 있으나, 그 개발에 많은 비용이 들고 아직까지는 대규모 공정에 적용하기에는 많은 어려움이 있다. 정상상태 모사나 최적화 등은 그 동안 독립적으로 많은 연구가 되어왔고, 이용할 수 있는 좋은 프로그램도 많이 존재한다. 따라서 동적모사의 기능을 주 목적으로 할 경우에는, 두 구조의 우월성을 비교하기에는 아직 이르다. 본 연구

에서는 두 방법의 장점을 적절히 조화시키는 모사 구조를 제안하였고, 이러한 구조하에서 동적모사의 여러 응용 분야에의 적용을 쉽게하기 위한 계산전략을 개발하였다. 제안된 구조는 순차적 모듈 방식에 기본을 두고 해당 공정의 특성에 알맞는 모사문제를 선택할 수 있는 유연한 구조로 생각된다. 본 연구에서는 모사전략으로써 유연한 모듈방식에 대하여 제시하였고, 이 방식을 채택한 원형 동적모사기의 주요 구조 및 기능에 대하여 설명하였다. 또한 간단한 예제를 통하여 구현된 전략 및 시스템의 성능 평가를 하였다.

### 2. 모사방법론

화학공정의 동적모사를 위한 모델식은 다음식과 같이 미분식과 대수식으로 나타내게 된다.

$$dy/dt = f(x, y, t) \tag{1}$$

$$0 = g(x, y, t) \tag{2}$$

$$y = y_0 \text{ at given time, } t_0$$

이때  $y$ 는 미분변수 또는 상태변수,  $x$ 는 대수변수,  $t$ 는 독립변수로서 주로 시간을 나타낸다.

화학공정은 기본적으로 잘 알려진 여러 개의 단위조작의 결합으로 이루어져 있고, 이러한 단위조작들은 하나 이상의 모듈(방정식시스템)로 적절히 모델링 될 수 있다. 모사전략 및 구조는 여러 개의 모듈로 이루어진 공정 문제를 푸는 방법을 제시한다. 먼저 다음과 같은 5개의 모듈로 이루어진 화학공정 모사문제를 생각해 보자.

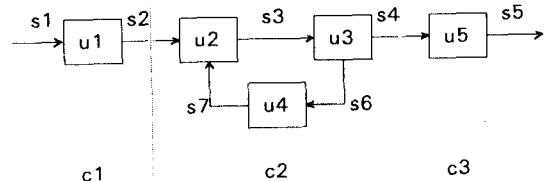


Fig. 1. 3-cluster process after partitioning

이 공정을 풀기위한 partitioning을 행하면 (1), (2,3,4), (5)와 같은 세개의 그룹을 얻는다. 괄호는 그룹을 의미하며 수는 단 위모듈을 나타낸다. 이 그룹을 cluster로 정의하면 위 공정은 세 개의 cluster로 이루어져 있고, cluster를 계산의 단위로 생각하면 순환루프가 없는 순차공정이 된다.

Fig.1과 같은 공정은 C2만이 순환루프가 존재하므로 C2에 속한 단위 모듈 2,3,4를 동시에 수렴시키는 방법을 사용하고 나머지 단위 모듈은 다른 모듈과 커플링 되지 않고도 독립적으로 모사가 수행되게 된다. 이러한 방식을 사용함으로써 계산의 정확성과 동시수렴법을 사용할 문제가 축소됨으로써 기존의 두 방식보다 효율성에서 우수하게 된다. 또한 하나의 cluster는 하나의 모듈에서 부터 전 공정에 이르는 모든 모듈을 포함할 수 있으므로 문제에 적합한 구조를 사용할 수 있다. 본 연구는 서브루틴화 되어있는 독립적인 모듈을 사용하여 cluster에서의 동시수렴법을 사용하는 방법과 전체 공정의 수렴성을 향상시키기 위한 구조 및 전략에 대하여 제시하였다.

## 2.1 적분 전략

모듈 중심 모사의 장점중의 하나는 각각의 모듈은 자신의 동특성에 알맞은 수치해법을 사용할 수 있어 계산 효율을 높일 수 있다는 것이다. Cluster는 공정에서 상당히 독립적인 성격을 띠고 있으므로, 다른 수치해법, 특히 적분방법을 사용할 수 있을 것이다. 그러나 상이한 적분방법을 사용했을 때는 각각의 계산블럭 사이에 통신을 할때 시간을 맞추기 위한 특별한 coordinator가 필요하고, extrapolation, interpolation 횟수가 증가함에 따라 오차가 도입될 수 있고, 또한 상당한 부하가 걸리는 작업이 될 수 있다. 그래서 본 연구에서는 모든 cluster가 동일한 코드를 사용하고 동일한 시간 간격을 가지도록 하는 방법을 사용하였다. 이를 위하여 IMSL DGEAR 코드를 기본으로 하여 여러 모듈이 동시에 적분될 수 있는 적분기를 개발하여 사용하였다. 식(1)과 (2)에서 다음과 같은 표준적인 일차 상미분 방정식을 얻을 수 있다.

$$dt/dt = f(y, t) \quad (3)$$

식(3)은 BDF 방법을 사용하면 다음과 같이 나타낼 수 있다.

$$y_n = \varphi + hb_0 f(y_n, t_n) \quad (4)$$

$$\text{where, } \varphi = \sum_{i=1}^q a_i y_{n-i}$$

이때 아래첨자 n은,  $t_n$ 에서의 값을 나타내고,  $a_i, d_i, b_0, e_0$ 는 적분차수 q에 의존하는 method 상수이다. 식(4)는  $y_n$ 에 대하여 음함수이고, 함수 f가 비선형이므로 일반적으로 Newton type의 반복 계산법이 사용된다. 식(4)를 quasi-Newton 방법을 적용하면 다음과 같은 식을 얻는다.

$$y_n^{m+1} = y_n^m + (I - hb_0 J^*)^{-1} [hb_0 f(y_n^m, t_n) + \varphi - y_n] \quad (5)$$

$$y_n^{(0)} = \sum_{i=1}^q d_i y_{n-i} + he_0 f(y_{n-1}, t_{n-1}) \quad (6)$$

이때 I는 identity 행렬, m은 반복 계산 횟수,  $J^*$ 는 과거에 구해진 Jacobian을 나타낸다. 식(6)은 반복 계산의 초기값을 구하는데 사용되며,  $y_n$ 에 대해 양함수 형태이며 이러한 식을 예측자 (Predictor)라고 한다. 위와 같은 다단계 적분방법은 몇 개의 과거값을 가지고 있어야 하는데, 이보다는 Nordsieck (1962) 행렬을 사용하면 적분간격 및 오차제어에 훨씬 유리하다. Nordsieck 행렬, Z를 사용하여 식(5), 식(6)은 다시 정리하면 다음과 같다.

$$hy^{m+1} = hy_n^m + (I - hb_0 J^*)^{-1} (hf(y_n^m, t_n) - hy_n^m) \quad (7)$$

$$y_n^{m+1} = y_n^m + I_0 (hy_n^{m+1} - hy_n^m) \quad (8)$$

$$Z_n^0 = DZ_{n-1} \quad (9)$$

식(9)에서 D는 Pascal 삼각행렬, Z는 Nordsieck 행렬이다.

$$D_{ij} = \binom{i-1}{j-1} = \frac{(i-1)!}{(i-j)(j-1)!} \quad \text{if } j \leq i$$

$$D_{ij} = 0 \quad \text{if } i < j \quad (10)$$

$$Z_n = [y_n, hy_n^1, \frac{h^2 y_n^{(2)}}{2!}, \frac{h^3 y_n^{(3)}}{3!}, \dots, \frac{h^k y_n^{(k)}}{k!}] \quad (11)$$

모사하고자 하는 공정이  $nc$  개의 cluster로 이루어져 있다면 다음과 같은 계산절차를 따라 동적모사에 따른 적분계산을 행한다.

**1. 예측단계 :** 모든 cluster에 대하여 식(9)를 이용하여  $D_n$ 을 예측한다. 이때 모든 tear stream이 예측되는 효과를 가진다. 예측된 값을 가지고 모든 모듈에 대하여 식(2)의 대수식을 계산한다. 대수식 계산은 뒤에서 다룬다.

**2. 수정단계 :** 하나의 선택된 cluster에 대하여 식(7),(8)의 비선형 방정식을 푼다. 수렴성과 오차시험을 거친다. 수렴성과 오차시험에 성공하면 Nordsieck 행렬을 갱신하고 다음 cluster의 수정계산을 시작한다. 만약 실패하면 이 cluster는 fail-cluster로 되며 성공할 수 있는 적절한 적분차수 q와 적분간격 h를 구한다. 현재 시간 간격에 행하여졌던 모든 계산이 취소되고 축소된 새로운 h를 가지고 1.단계로 간다.

**3. 적분간격 및 적분차수 최적화 :** 현재 시간 간격의 모든 cluster가 성공하면 다음 시간에 적용할 최적 h와 q를 구한다. 현재의 적분차수가 q일때 q-1, q, q+1의 각각에 대하여 적용할 적분간격,  $h_{q-1}, h_q, h_{q+1}$ 를 계산한다.

$$h_{q-1} = h_n \frac{1}{1.3} \left[ \frac{\varepsilon}{C_{q,q}!} \frac{1}{\left\| \frac{a_q}{\omega} \right\|_2} \right]^{1/q} \quad (12)$$

$$h_q = h_n \frac{1}{1.2} \left[ \frac{\varepsilon}{C_{q+1,q}!} \frac{1}{\left\| \frac{\nabla a_q}{\omega} \right\|_2} \right]^{1/(q+1)} \quad (13)$$

$$h_{q+1} = h_n \frac{1}{1.4} \left[ \frac{\varepsilon}{C_{q+2,q}!} \frac{1}{\left\| \frac{\nabla^2 a_q}{\omega} \right\|_2} \right]^{1/(q+2)} \quad (14)$$

이때  $\varepsilon$ 는 사용자가 정한 오차한계이며,  $\nabla a_q$ 는  $Z_n$ 의 마지막 원소의 backward difference이다. 일반적으로 방정식 시스템에서는 L2-norm을 사용한다.

서로 다른 차수에 대하여 계산된 h 중에서 가장 큰 h 값을 선택하며 그때의 차수를 다음 시간 간격에 적용한다. 그러나 여러 개의 cluster가 있을 때는 각각의 cluster에 대하여 동일한 h를 선택하기 위하여 다음과 같은 방법을 사용할 수 있다.

### 방법 1: MAXMIN(Maximum of Minimums)

모든 cluster에서 같은 h와 q를 사용하는 방법으로 Fagley 등이 사용하였다. 식(15.4)에 의해 결정된 h와 q를 전체 시스템에 사용한다.

$$h_{q-1} = \min ( h_{q-1,k} ), \text{ for } k = 1, nc \quad (15.1)$$

$$h_q = \min ( h_{q,k} ), \text{ for } k = 1, nc \quad (15.2)$$

$$h_{q+1} = \min ( h_{q+1,k} ), \text{ for } k = 1, nc \quad (15.3)$$

$$h = \max ( h_{q-1}, h_q, h_{q+1} ) \quad (15.4)$$

### 방법 2: MINMAX(Minimum of Maximums)

본 연구에서 제안한 방법으로 전체 시스템의 h는 식(16.2)에 의해 결정하고, 각 cluster의 q는 식(16.1)에 의해 결정한다.

방법 2는 항상 방법 1보다 좀더 큰 h를 제안할 수 있으며, 각 cluster의 특성에 적합한 차수를 사용함으로써 계산 효율을 높일 수 있다.

$$h_k = \max ( h_{q-1,k}, h_{q,k}, h_{q+1,k} ), \text{ for } k = 1, nc \quad (16.1)$$

$$h = \min ( h_k ), \text{ for } k = 1, nc \quad (16.2)$$

## 2.2 모사 전략

동적모사의 방법론은 주로 미분식과 대수식의 풀이 방법에 따라 구분된다. 예를 들어 Fig.1과 같은 공정을 생각하면, 순차모듈 방식은 각 모듈은 식(1)을 적분하기 위해 독립적으로 적분기를 호출하고, 모듈 내의 대수식을 계산한다. 이 모듈이 수렴하면, 적절한 계산순서에 따라 다음 모듈을 같은 방법으로 계산한다. 계산순서는 정상상태 모사의 경우와 비슷한 Precedence Ordering에 따른다. Fig.1의 경우에는 3개의 스트림, s3, s6, s7중에 하나를 tear-stream으로 한다. 예를 들어 s7을 tear-stream으로 하면, (1)-(2)-(3)-(4)-(5)과 같은 계산 순서를 생각할 수 있다. Hillestad(1986) 등은 tear-stream의 값을 예측하는 모듈을 prediction 모듈이라고 하였으며, 이를 적절히 선택하여 공정의 순차구조를 얻은 다음 순차모듈 방식을 적용하였다. 이 방식은 장점은 계산이 비교적 간단하고, 해법이 강건하다는 것이다. 또한 모듈 단위의 동특성을 구분하여 이용할 수 있다는 것이고, 병렬처리 컴퓨터를 이용하기 쉽다는 것이다. 단점으로는 복잡한 순환구조를 가진 공정은 다루기가 쉽지 않다는 것이다. 그러나 이때 단순한 물질흐름의 순환구조는 공정의 capacity나 시간지연 등의 동특성으로 수렴에는 큰 어려움은 없지만 물질흐름과 역방향으로 전달되는 정보흐름이 있을 때 모사 성능이 떨어진다.

반면 방정식 중심방식은 모든 모듈에 대한 방정식을 모아서 하나의 수치해석 루틴을 사용하여 수렴시킨다. 식(4)와 식(2)를 합쳐서 하나의 수식으로 쓰면,

$$Dy_n = D\phi + hb\phi f(y_n, t_n) \quad (17.1)$$

Newton-Raphson 방법을 적용하면,

$$(D-hb\phi J^*)\Delta y_n^{m+1} = hb\phi f(x_n^m, y_n^m, t) + D(\phi - y_n^m) \quad (17.2)$$

where,

$$J_{ij} = \frac{\partial f_i}{\partial y_j}$$

$$\Delta y_n^{m+1} = y_n^{m+1} - y_n^m$$

$$D = I \text{ for ODEs}$$

$$D = 0 \text{ for AEs}$$

$$D = I \text{ or } 0 \text{ (DAE system)}$$

이 방식은 문제의 구성에 있어서 유연하고, 수학적으로 매우 우수한 방법이어서 최근 많은 연구가 행해지고 있다. 그러나 대규모 DAE 시스템의 수렴에는 많은 비용이 들고, 신뢰할 만한 충분한 강건성이 부족할 편이다. 특히 대규모 복잡한 공정의 경우 식(17)을 동시에 수렴시키기 위한 초기값을 구하는 문제도 매우 어렵다.

본 연구에서는 플랜트 구조를 partitioning을 하여 cluster를 얻은 후에 cluster를 하나의 계산 단위로 정의하는 유연한 모듈 방식을 사용하였다. Fig.1의 공정은 3개의 cluster로 풀 수 있는데, (1)-(2,3,4)-(5)의 순서에 의해 동적모사를 행하면 효율성과 정확성 면에서 기존의 두 방법보다 유리하다. 이때 c2는 u2, u3, u4 모듈의 DAE를 연립하여 풀게 되므로 전체 공정을 연립하여 푸는 방정식 중심법 보다는 작은 문제를 다루게 될 뿐 아니라 분할된 공정의 특성에 적합한 수치해법을 적용할 수

있다. 이 방식의 가장 큰 장점은 완전한 모듈 구조에 있다. 즉 독립적인 서브루틴 형태의 모듈을 사용하여 cluster 안에서 동시해법이 이루어지도록 한다. 일반적으로 다음 Fig.1과 같은 두개의 모듈을 가진 cluster를 푸는 방법은 다음과 같다.

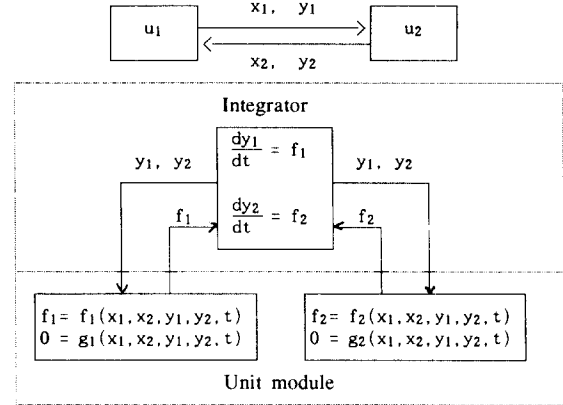


Fig.2. Cluster having two interacting units

Fig. 2에서 보이는 바와 같이 적분기는 cluster의 미분방정식을 풀고, 단위 모듈은 대수식을 풀어 미분방정식의 RHS 항을 계산한다. 이 때 모듈의 호출 순서를 u1, u2로 하고, 적분기로부터  $y_{1,n}^m, y_{2,n}^m$ , 즉 시간  $t_n$ 에서의 m번째 반복 수정값을 계공받았을 때, 모듈 u1은  $x_1$ 을 모듈 u2는  $x_2$ 를 구해서 각각  $f_1, f_2$ 를 적분기로 넘겨주는 역할을 해야 한다. 그러나 u1, u2를 순차적으로 한번 호출하면 u1은 아직  $x_2$ 를 알 수 없으므로 m-1번째의 수정값을 사용하게 된다. 그러나 아직 같은 시간대에서 계산이 이루어지고 있으므로, 전통적인 순차모듈 방식보다는 우수한 풀이 기법이 된다. 이와 같은 순환루프의 문제점을 극복하기 위하여 본 연구에서는 3단계 모듈호출 방식을 개발하였다. 모듈의 대수식은 그 형태에 따라 다음과 같은 3가지로 구분할 수 있다.

### Type 1: Internal Explicit Algebraic Equations

$$x_e = g_e(y, t) \quad (18.1)$$

### Type 2: Internal Implicit Algebraic Equations

$$x_i = g_i(x_e, x_i, y, t) \quad (18.2)$$

### Type 3: External Explicit(or Implicit) Algebraic Equations

$$x_c = g_c(x, y, z, t) \quad (18.3)$$

$$\text{where } x = \{ x_e, x_i, x_c \}$$

$$z = \{ z_e, z_i, z_c \}$$

즉 type 1은 모듈내부의 상태변수 값이 주어지면 계산이 되는 가장 간단한 형태의 방정식이다. 예를들어, 장치의 수위(level) 같은 변수이다. Type 2는 type 1과 같이 내부 변수만의 함수이며  $x_i$ 는 반복계산법에 의해 구해진다. Type 3은 자신의 모듈내의 변수 외에 다른 모듈의 변수, z를 포함하고 있으며, 2개 이상의 모듈이 연결되어 있으면 나타날 수 있는 식이다.

다시 type 3에서  $z_e, z_i, z_c$ 와 마찬가지로  $z_e, z_i, z_c$ 로 구분된다. 주어진 y로부터 cluster의 RHS 항을 정확하게 계산하기 위해서는 cluster를 이루는 단위 모듈의 호출방식을 다음과 같이 행한다.

### 3-Phase 호출방식

**Phase 1:** Cluster를 이루는 모든 모듈에 대하여 식(18)의 type 1과 type 2의 방정식을 푼다. 한 모듈의 내부변수  $x_e, x_i$ 는 타 모듈의 외부변수,  $z_e, z_i, z_c$ 에 해당한다.

**Phase 2:** Cluster를 이루는 모든 모듈에 대하여 식(18)의 type 3의 방정식을 푼다. 이 때  $z_c$ 를 제외한 모든 외부 변수 값이 phase 1에서 구해졌으므로 모든 계산은 특별한 가정없이 수행된다. 그러나 type 3식 중에  $z_c$ 가 존재하면 식(19)과 같은 미해결의 식이 남는다.

$$x_c = g_1(z_c) \quad (19.1)$$

$$z_c = g_2(x_c) \quad (19.2)$$

이와 같은 식을 type 4로 분류하였다. 다행스럽게 대부분의 화학공정 동적모사 문제에서는 식(19)와 같은 모델식은 거의 없다. 예를들어 증류탑과 같은 다단탑 모델식에서도 type 4의 형태는 없고, vapor flow에 대한 식 만이 type 3으로 나타난다. 만약 복잡한 설계사양이 주어지거나 제어알고리즘으로 식(19)가 나타난다 하더라도 수렴에는 큰 어려움이 없을 것이다.

**Phase 3:** 식(18)과 식(19)를 수렴시킨후,  $f$  값을 구한다. 이렇게 구해진 RHS 항을 가지고, 적분기는 새로운 수정값을 계산한다.

### 3. 시스템구조

#### 3.1 데이터 구조

모사기의 데이터 구조는 시스템의 유연성과 확장성에 큰 영향을 미치며, 계산효율에도 영향을 미친다. 본 연구에서는 이러한 사항들을 고려하여 일차원 형의 배열에 포인터를 사용하는 구조와 객체지향적인 데이터 구조를 고안하여 사용하였다. 이러한 구조는 데이터 입출력에 부하가 걸리지만, 시스템의 확장성과 유연성 면에서는 유리한 방법이다. Fig. 3에 두 구조의 간략한 개념도를 나타내었다. 주로 수치해석과 관련되는 데이터는 pointer 형을 사용하고, 기타의 데이터 구조는 object 형을 사용하였다.

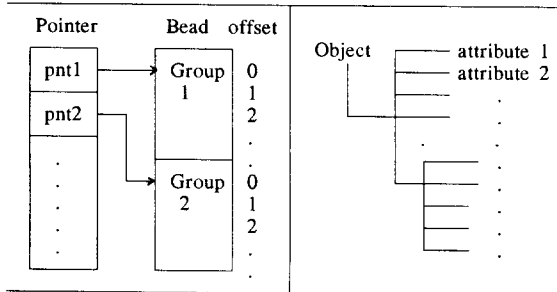


Fig. 3. Pointer type and object type data structure

#### 3.2 시스템 구조

MOSA는 크게 입력화일을 분석하여 문제를 생성하는 입력해석기(Input Translator)와 전체 모사를 수행하는 집행부분(Executive), 여러 개의 cluster에 대한 한 스텝의 적분을 감독하는 적분감독부분(Supervisor), 실제 하나의 cluster를 적분하는 적분기(Integrator)와 단위 공정의 모델식을 가지고 있는 단위 모듈, 기타 열역학 루틴, 최소행렬 계산 및 여러 인터페이스로 이루어져 있다. 전체적인 시스템의 구성도는 Fig.4와 같다.

#### -입력언어(Input Language)

모사하고자 하는 문제또는 공정을 정의 하는 부분으로 보통의 정상상태 모사기와 유사하나, 특징적인 것은 공정의 partitioning을 할 수 있는 CLUSTER 부분과 그래픽 출력을 위한

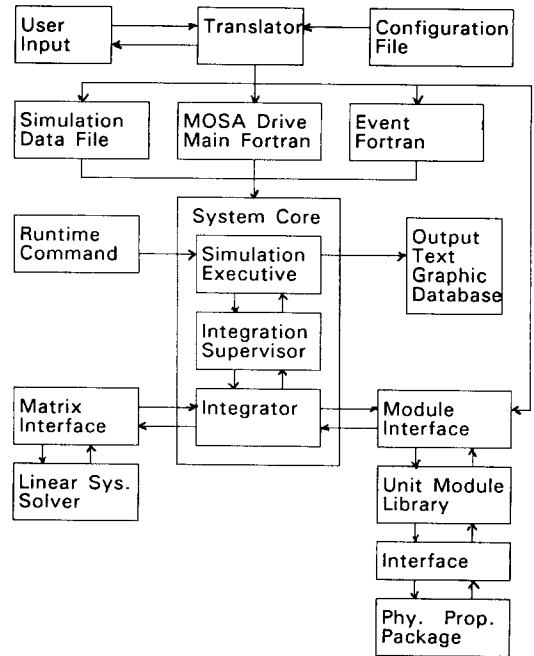


Fig. 4. Overall system structure of MOSA

PLOT 부분, 특히 동적모사의 가장 특징적인 사건을 다루기 위한 EVENT 부분이 있다. 사용자가 입력한 ASCII 화일을 읽어들이고 후, 정의된 예약어 중심으로 해석을 한 후, 실제 모사를 위한 주 프로그램, 모사 데이터 화일, 사건처리 프로그램, 모듈 호출 인터페이스 루틴 등을 생성한다. 시스템이 인식할 수 있는 단위 모듈의 변수이름, 스트림의 변수 등, 모사 시스템이 사용하는 모든 변수는 SETLIB.CFG와 SETSTR.CFG 라는 화일에 정의되어 있다. 즉 새로운 모듈을 작성하거나 새로운 변수를 도입하려면 이 화일에 첨가 시키면 된다.

#### - 집행부분(Executive)

입력언어에 의해 작성된 메인 프로그램에 의해 시스템의 각종 배열의 크기 등이 결정되고, 이 프로그램은 집행부분을 호출한다. 실제로 모사 시스템의 모든 기능은 이 부분에 집중되어 있다. 시스템의 변수들을 초기화 시키고, 필요한 데이터 화일을 읽거나, 데이터 베이스를 작성한다. 모든 모사 준비가 완료되면 명령어 수준에 머물러 사용자의 명령을 기다린다. 모사 시작을 알리는 명령을 받으면, 정해진 한 스텝의 적분을 실행한다. 이때 각종 사건(Event)을 처리하며, 또한 적절한 시간에 출력을 하고, Runtime interrupt 명령이 들어오면 이를 해석하고 수행하는 기능을 한다. Fig.5에 집행 루틴의 구조를 나타내었다. 모사를 수행하는 동안 어느 때라도 interrupt에 의해 명령을 수행시킬 수 있다. 이 부분은 실제로 동적모사에 매우 유리하며, 완전한 interactive simulation을 가능하게 한다. 또한 장치 혼련용 시스템을 구축하는데 기본이 되는 기능이다. 집행부분의 또 하나 중요한 기능은 사건을 처리하는 기능이다. 사건은 수학적으로는 불연속의 입력과 처리를 말하는 데, MOSA 시스템에서는 특별한 데이터 구조를 사용하여 사건을 처리하고 있다. 즉 사건은 크게 Explicit event와 Implicit event로 구분된다. 전자는 사건이 일어나는 시간이 미리 알려져 있는 것이고, 후자는 그렇지 못한 경우이다. Explicit event의 처리는 미리 시간을 알기 때문에 정확히 그 시간에 적분간격을 일치시켜 한 스텝을 마친후 불연속이 시작되는 시간부터는 다시 초기값 문제를 다시 푸는 방식으로 해결

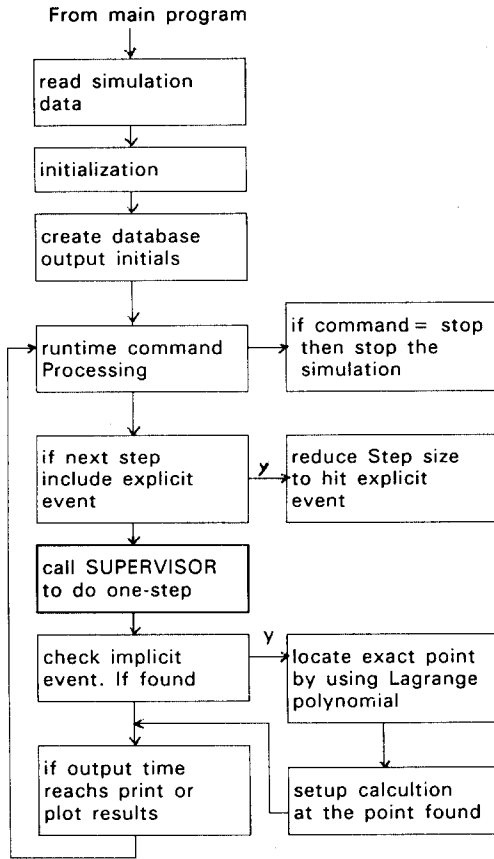


Fig.5. Structure of MOSA executive

할 수 있다. 문제는 implicit 한 경우인데, 어떻게 효과적으로 정확한 점을 찾느냐 하는 것이다. 본 연구에서는 불연속을 유발시키는 조건의 값을 history에 저장하여 보관하고 있다가 불연속이 활성화 되면 history에 저장된 값에 Lagrange interpolation을 적용하여 불연속을 찾는 방법을 개발하였다.

#### 4. 모사기 시험

현재 제안된 모사전략의 특성을 여러 응용 예제를 통하여 시험하고 있는 중이므로, 모든 면이 아직 규명되지는 않았다. 하지만 2장에서 살펴본 바와 같이, 모듈구조의 장점은 다양한 목적에 맞는 모듈을 비교적 쉽게 구성할 수 있으며, 이렇게 구성된 모듈은 안정되고 정확한 해를 구할 수 있다. 또한 빈번한 불연속이 존재하는 공정이거나, batch 공정과 같이 공정의 특성이 시간에 따라 또는 생산계획에 따라 급변하는 공정의 동적모사에는 모든 모듈을 동시에 푸는 방정식 기법보다는 단위 모듈 단위의 모사가 훨씬 유리하다. 또한 전 공정이 아닌 커플링이 심한 공정만을 동시해법으로 구함으로써 계산상의 효율을 기대할 수 있다. 제안된 3-phase 모듈 호출방식은 일반적인 물질흐름에 의한 순환루프가 존재하는 공정에 대해서도 순차 모듈 방식이 유리하다.

Fig. 6과 같은 공정을 대상으로 3-phase 호출방식이 우수함을 보였다. 편의상 각 블록을 간단한 혼합탱크로 모델링 하였으며, 임의의 초기 holdup을 가지도록 하여 정상상태까지의 start-up 모사를 행하였다. Table 1에 3가지 방식에 대하여 비

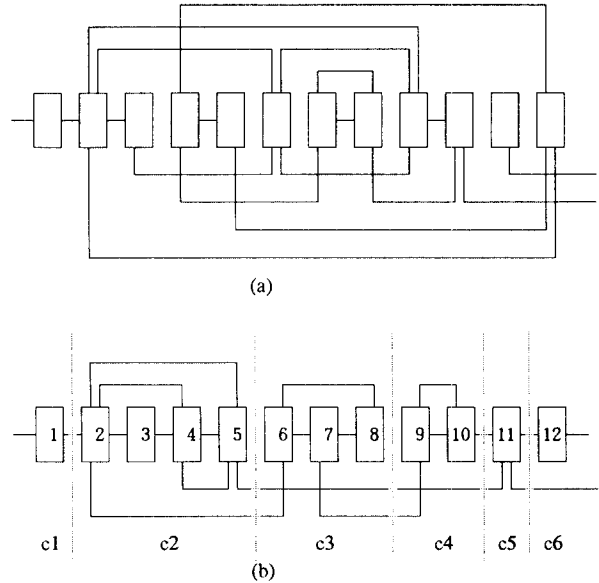


Fig. 6. 12-unit process

(a) before partitioning, (b) after partitioning

교하여 나타내었다. 상당히 복잡한 공정이라도 커플링 되어 있는 공정간에 식(19)와 같은 type-4의 커플링이 존재하지 않고 또한 미분변수들 간에 직접적인 커플링 없기 때문에 단순한 순차 모듈 방식이 유리하다. 그러나 일반적으로 계산 단위인 cluster 수가 늘어나면 LU-decomposition 등과 같은 선형 방정식을 풀때, 부하가 더 걸리게 되어 cpu 시간 면에서는 불리할 수 있다.

Table 1. Performance comparison on Fig 6. Process

eps = 0.0001, tfinal = 10, output step = 0.2

Method	% NFE	% CPU
12-cluster(all sequential)	100	100
6-cluster	138	98
1-cluster(all simultaneous)	362	110

그러나 Fig. 7은 두개의 상호작용 탱크로 이루어져 있으며 물질흐름만을 살펴보면 순차적 공정이나 정보흐름에 의하여 루프를 형성한다. 물질흐름에 비해 정보흐름은 공정에 의한 시간 지연이 아주 작거나 없기 때문에 서로가 민감하게 상호작용을 하게 된다. 이러한 공정은 cluster를 도입하는 것이 바람직하다. Fig. 7의 문제는 정상상태에서 출발하여 t=0.4일때 F2의 밸브상수가 10에서 5로 계단변화를 시키고, 다시 시간 t=1.0 일때 10으로 환원시킨 가상 시나리오를 모사하였다. 즉, F2 파이프의 leaking 또는 저항으로 F2 유량이 줄어들었다가 정상으로 회복되는 것과 같은 효과를 모사한 것이다. Table 2에 나타낸 것과 같이 C1, C2를 tearing한 1의 경우는 해를 얻을 수 없었고, 모든 공정을 하나의 cluster로 한 경우 (III)가 가장 바람직한 결과를 보여준다. Fig. 8에 응답을 나타내었다.

#### 5. 결론

본 연구에서는 복잡하게 커플링 되어 있는 경우에도, 모듈구조의 모사 방법으로 해결할 수 있는 모사전략을 제시하고, 이를

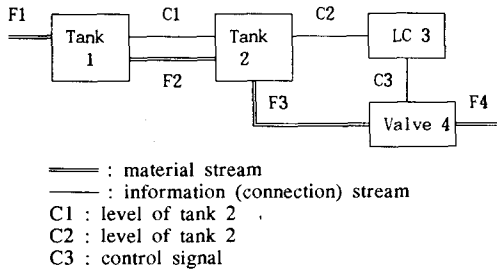


Fig. 7. Two-interacting tank problem

Table 2. Performance comparison on Fig.7 process

eps = 0.0001, tfinal = 2, output step = 0.01

	Method	% NFE	% CPU
I	(1), (2), (3 4)	incorrect solution	
II	(1) (2 3 4)	100	100
III	(1 2 3 4) (all sim)	91	30

비탕으로 원형 동적모사기를 개발하였다. 제안된 구조는 단위 모듈로 부터 전 공정에 이르기까지의 커플링을 도입할 수 있는 유연한 모듈구조로서 공정의 partitioning 정보를 이용하여 최대의 계산효율을 발휘할 수 있다. 특히 3-phase 모듈호출 방식은 공정의 동특성을 분할하여 계산의 정확도를 높일 수 있으며, 모듈중심 기법으로 모듈간의 동시계산(simultaneous solving)을 가능하게 한다. 동적 모사기의 확장성과 유연성 그리고 응용성을 고려하여 데이터 및 프로그램의 구조를 확정하였다.

#### References

1. Fagley, J.C. and B. Carnahan, *Comput. chem. Engng.* 14, 2(1990).
2. Gear, C.W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, New Jersey(1971).
3. Hillestad, M and T. Hertzberg, *Comput. chem. Engng.* 1, 4(1986).
4. Kuru, S. and A.W. Westerberg, *Comput. chem. Engng.* 9, 2(1985).
5. Liu, Y.C. and C.B. Brosilow, *Comput. chem. Engng.* 11, 3(1987).
6. Pantelides, C.C., *Comput. chem. Engng.* 12, 7(1988).

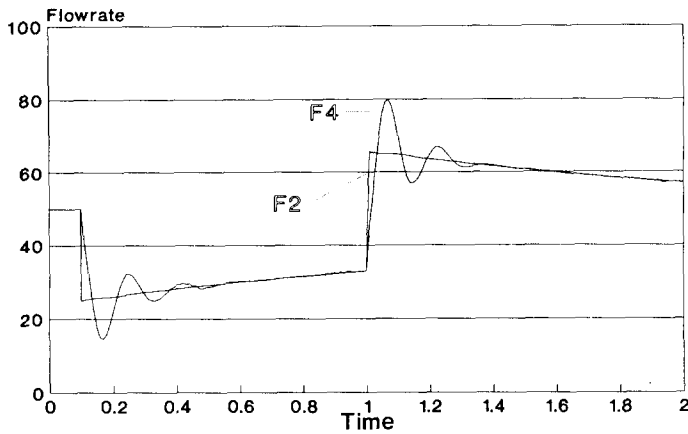


Fig. 8. Dynamic response of process in Fig. 7