



연결성과 소통구조 모듈을 통한 차세대 범용 농업시스템 시뮬레이터 (GASS2)의 개발

Development of GASS2 through Improving Inter-component Connection and Communication Modules

김태곤* · 이성용** · 이호재*** · 이정재**** · 서교*****,†

Kim, Tae Gon · Lee, Sung Yong · Yi, Ho Jae · Lee, Jeong Jae · Suh, Kyo

Abstract

The purpose of this study is to improve the Generic Agricultural Systems Simulator (GASS) which can simulate various rural systems based on object-oriented model. GASS provides the configuration platform of various system components to simplify integrated agricultural system models such as irrigation systems for rice farming. The new connection and communication modules of GASS improve applicability for modelling diverse systems. The geometric connection of GASS replaces topological connection, and communication protocols expand to analyze not only homogeneous system but also heterogeneous system. In this paper, we applied GASS2 to simulate the water heights of linked tanks and the simulation outputs were verified through comparing with analytical solutions of differential equations. The two new modules make it possible to analyze the 4-tank problem which includes topological and heterogeneous issues with GASS2.

Keywords: Communication; connection; GASS; object-oriented; simulator

1. 서론

시스템 시뮬레이션을 위한 모델을 만들기 위한 과정은 일반적으로 복잡하고, 기 구축된 모델의 유지보수 또한 쉽지 않다 (Jones et al., 2001). 이와 관련하여 많은 연구자들이 객체지향 프로그래밍 패러다임과 같은 소프트웨어 공학기술을 이용하여 문제를 해결하고자 노력해 왔음에도 불구하고 (Van Evert and Campbell, 1994; Acock and Reddy, 1997), 그 성과를 활용할 수 있는 연구자는 극소수에 불과하였다. 이는 시스템을 구성하기 위하여 다수의 서브시스템이 결합되면서 여러 시스템 간의 정보 전달을 효과적으로 정의하기가 어렵

고, 시스템이 확장될수록 시스템 사이의 연결수가 급격히 증가하여, 이에 따른 정보 전달의 복잡성이 증가하기 때문이다 (Yi, 2003).

이러한 복잡성을 해결하기 위하여, 일방향 정보전달 기법 (Unidirectional Information Fetch, UIF)과 요소 독립성을 기반으로 한 범용농업시스템 시뮬레이터 (Generic Agricultural Systems Simulator, GASS)가 개발되었다 (Yi, 2003). 일방향 정보전달 기법은 서브시스템 사이의 정보교환을 위하여 상호의존적인 정보교환 방식에서 탈피하여, 인접한 모델에서 필요한 정보만을 가져올 수 있도록 설계됨으로써, 복잡도를 크게 낮추었다. 여기서, 복잡도란 사용자가 모델의 결합과 정에서 속성값을 서로 연결시켜야 하는 횟수로 정의하였다. 또한 지나친 객체지향 패러다임의 사용을 지양하고, 시스템을 구성하는 단위모델이 독립적으로 구동될 수 있도록 요소 독립성이 유지하는 환경을 구축하였다 (Lee et al., 2012). GASS의 컴포넌트는 방어적 프로그래밍 개념 (Goodliffe, 2007)으로 구현되어, 기본값을 이용해서 독립적으로 구동하며, 새로운 요소가 추가될 때 연결 구조를 통해 새로운 값으로 갱신하도록 설계되었다. 이러한 GASS는 비주얼 프로그래밍을 적용하여 사용자가 직접 코딩을 하지 않고, 마우스를 이용하여 모델을 인접하게 위치시키는 작업만으로 시스템을 구성할 수 있다. 여러 종류의 시뮬레이션 모델이 점진적으로 결합할 때, GASS를 통하여 시스템이 주어진 구성에 따라 자기 조직화하는 과정을 모사하였으며, 이후 관개시스템 모의, 열전도 모의, 농산물 유통 모의와 같은 다양한 분야의 문제해결에

* Institutes of Green Bio Science Technology, Seoul National University

** Department of Landscape Architecture and Rural Systems Engineering, Seoul National University

*** Agricultural and Biological Engineering, Pennsylvania State University

**** Department of Landscape Architecture and Rural Systems Engineering, Research Institute of Agriculture and Life Sciences, Seoul National University

***** International Agriculture Technology Graduate School, Institutes of Green Bio Science Technology, Seoul National University

† Corresponding author

Tel.: +82-2-880-4715 FAX: +82-2-873-2087

E-mail: kyosuh@snu.ac.kr

Received: May 19, 2014

Revised: August 20, 2014

Accepted: August 26, 2014

적용되었다 (Yi, 2003; Suh et al., 2005; Kim, 2007). 하지만 시스템을 구성하는 요소의 연결을 기하학적 인접성에 기반을 둔 탓에 정보전달에 취약한 문제가 지적된 바가 있다 (Kim and Lee, 2007).

따라서 본 연구에서는 GASS가 갖는 연결방식 (connection) 과 구조에 따른 통신 방식 (communication) 측면에서 한계를 개선하고, 편의성을 향상시키기 위하여 프로그램을 재설계 하여 GASS2를 개발하였다. 연결방식과 통신 방식의 문제점을 분석하여 기능 개선방안을 기술하였고, 기존 GASS에서 부족했던 시뮬레이션 모듈의 개발을 위한 규약을 정의하고 설계하였다. 서로 연결된 저류조의 수위변화 해석을 통하여, 시뮬레이션 모델을 검증하고, 기존 GASS에서 불가능하였던 위상구조와 이질소통구조가 혼합된 문제를 모의하였다.

II. 기능 개선을 통한 GASS2의 개발

1. 위상구조 기반의 연결방식 개선

기존의 GASS는 컴포넌트의 연결성을 설정하기 위하여 기하학적 (geometry)으로 인접한 컴포넌트를 검색하는 방식으로 구현되었다. 이는 현실 세계에서 물리적 공간을 차지하는 실체에 대하여 모델링을 수행한다는 개념으로 접근했기 때문이다. 모델링으로 활용되었던 대상을 살펴보면, 관개시스템 모의에서는 논, 수로, 유역, 저수지, 작물(벼), 대기를 대상 컴포넌트로 정의하였으며, 구조시스템에서는 수문과 강, 바다로 정의하였고, 또 다른 구조시스템에서는 물, 사면, 포화토, 불포화토로 정의하였다 (Yi, 2003). 이들 컴포넌트는 연결성이 단순하여, 2차원 공간상에서 인접하게 위치시킬 수 있었기 때문에 시뮬레이션 모델을 표현하는 것이 가능했다. 그리고 농산물 유통을 모의하기 위하여 추상적인 모델인 생산자, 도매상인, 소매상인으로 시뮬레이션 컴포넌트를 구성하더라도 (Suh et al., 2005), 복잡하지 않은 관계는 기하학적 인접성에 기반한 컴포넌트 연결이 가능하였다.

하지만 다수의 수로 컴포넌트가 연결되고, 양 끝단에 위치하는 저수지 컴포넌트와 논 컴포넌트가 정보를 교환할 때, 기하학적인 연결 구조로는 단위시간에 한 단계의 정보전달이 이루어져 정보전달이 이루어져야 하는 저수지 컴포넌트와 논 컴포넌트 사이에 존재하는 수로 개수만큼의 정보 지연이 발생하였다 (Kim and Lee, 2007). 이와 같이 전달 속도가 다른 정보에 대하여 정보계와 물질계로 분리하여 속도가 빠른 정보계는 브로딩캐스트 기법을 이용하여 구현한 사례가 있으나, 본 연구에서는 근본적인 대안으로 컴포넌트 간 관계를 위상학적 관계 (topology)로 정의함으로써 정보계와 물질계 모

두를 표현할 수 있도록 설계하였다. 위상구조 기반의 연결방식은 기하학적으로 인접한 컴포넌트를 전처리에서 계산하여 산입함으로써, 기하학적 인접성과 위상학적 관계를 모두 담을 수 있다.

이러한 위상학적 관계는 그래프 이론에 따라 인접행렬로 정의할 수 있다. 공학 문제는 대부분 실세계에 실존하는 문제로서 기하학적인 관계에 기반을 두기 때문에 인접행렬로 관계를 서술할 경우에 희소행렬일 가능성이 높다. 대표적으로 유한요소법에서 전체 강성행렬 (global stiffness matrix)은 대부분 희소행렬이다. 따라서 효율적으로 인접행렬을 관리하기 위하여 리스트 구조를 이용하여 컴포넌트 관계를 관리하도록 시뮬레이터를 개선하였다.

2. 이질 소통 구조를 도입한 통신방식 개선

시스템 시뮬레이션은 다수의 동일한 컴포넌트가 연결되어, 서로 정보를 교환하며 유기적으로 기능을 수행하는 행태를 모의한다. 이때 정보통신 방법은 크게 동질소통구조와 이질소통구조로 나눌 수 있다. 동질소통구조는 동일한 컴포넌트끼리 정보를 교환할 때 한 가지 경로로만 정보를 소통하는 행태로서 Fig. 1과 같이 도시할 수 있다. 동질소통구조를 갖는 시스템에서는 컴포넌트 간의 연결 유무만 의미가 있으며, 계산을 위한 정보는 컴포넌트가 갖고 있는 속성으로 모의된다.

반면 이질소통구조는 동일한 컴포넌트 사이에 정보 교환 경로가 여러 가지 형태인 시스템에서 나타나는 구조로, Fig. 2와 같이 도시할 수 있다. 동질소통구조와 달리 이질소통구조는 정보 전달 경로가 2가지 이상을 포함하는 시스템 해석 문제로서, 정보를 전달할 때 동일한 클래스의 컴포넌트라고 하더라도, 대상 개체 (instance)에 따라 전달 체계가 달라진다.

이질소통구조를 반영한 정보전달을 구현하기 위하여, 컴

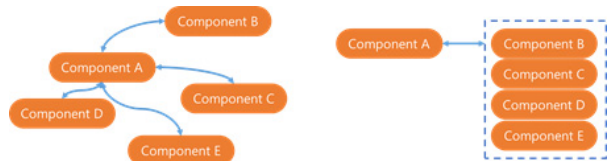


Fig. 1 Concept diagram of homogeneous communication

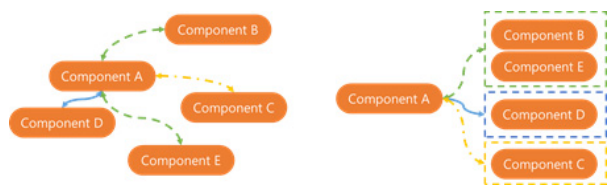


Fig. 2 Concept diagram of heterogeneous communication

포넌트 속성 정보 중 이질소통구조에 해당하는 속성은 연결 속성을 결합하여 저장하도록 구현하였다. Fig. 2의 컴포넌트 A에서 height와 이질소통구조로 연결된 weight라는 변수가 있을 때, height는 일반변수로 저장하여 관리하였다. 반면 weight 변수는 3번 컴포넌트와 5번 컴포넌트와 연결되어 있다고 가정할 때, (weight, 3), (weight, 5)와 같이 벡터변수로 저장하여 속성을 관리하도록 개발하였다.

3. GASS2의 구현

GASS2는 대표적 객체지향 언어인 java를 이용하여 구현하였으며, 주요한 클래스는 Fig. 3에 도시한 바와 같다. 시뮬레이션을 관장하는 GASSEnvironment 클래스에서 시뮬레이션에 관한 대부분의 기능을 구현하였으며, 이를 상속하여 계산결과를 데이터베이스에 저장하는 기능이 추가된 GASSEnvironmentDB 클래스를 구현하였다. IGASSComponent 인터페이스는 시뮬레이션 모델을 개발하는 사용자가 지켜야 할 규약을 정의하고 있으며, 기본적인 기능을 구현해 놓은 DefaultModel 클래스를 제공함으로써, 사용자가 주요 로직만 코딩 하면 되도록 설계하였다. 또한 계산과정을 확인할 수 있는 NetworkViewer와 GraphViewer 클래스를 구현하여 편의성을 도모하고자 하였다.

시뮬레이션을 구동하는 데 있어 가장 중요한 클래스인 GASSEnvironment 클래스의 주요 메소드로는 컴포넌트를 정의하는 setNodes(), 컴포넌트의 연결 구조를 정의하는 setAdjacencyList(), 연결 구조를 각 컴포넌트별로 알려주는 makeTree(), 시뮬레이션을 구동하기 위한 simulation() 메소드가 있다. Simulation() 메소드가 호출되면, 단위시간(tick)을 증가시키면서 각 컴포넌트가 preProcess(), process(), postProcess() 메소드를 수행하도록 제어한다. 결과를 데이터베이스로 저장하는 기능은 컴포넌트의 개수와 컴포넌트의 속성개수, 계산횟수에 따라 계산시간이 증가하므로, 빠르게 결과를 모의할 수 있도록 GASSEnvironment 클래스를 구현하고, 결과 저장을 위한 기능은 상속하여 추가하도록 하였다.

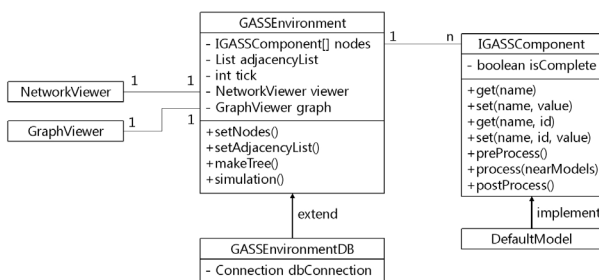


Fig. 3 Class diagram of the main class and component interface of GASS2

서로 다른 컴포넌트끼리 결합하여 모델을 구성하기 위해서는 컴포넌트 간에 통일된 규약이 필요한데, 본 연구에서는 IGASSComponent 인터페이스를 통하여 규약을 제한하였다. 모델을 개발하고자 하는 사용자는 IGASSComponent 인터페이스에서 정의한 메소드를 구현할 필요가 있으며, 중복되는 코드에 대하여 재활용할 수 있도록 DefaultModel 클래스를 제공하였다. 컴포넌트 개발 시 필요한 규약은 정보전달을 담당하는 get(), set() 메소드, 계산시간마다 각 컴포넌트가 고유하게 수행해야 하는 기능을 기술하는 preProcess(), process(), postProcess() 메소드로 구성된다. 마지막으로 각 컴포넌트의 종료 여부를 전달할 수 있는 isComplete 변수를 추가하였다.

시뮬레이션 결과를 저장하기 위하여 경량 데이터베이스를 도입하여 GASSEnvironmentDB를 개발하였다. 경량 데이터베이스는 데이터베이스 관리시스템 구동을 위하여 별다른 프로그램 설치 없이 기존 프로그램에 접착프로그램(glue program) 형태로 제공할 수 있다는 점에서 배포의 장점을 갖는다. 시뮬레이션된 결과는 경량 데이터베이스인 SQLite를 통하여 데이터베이스 파일로 저장되며, 시뮬레이션이 종료된 후, csv 파일로 변환하여, 엑셀로 쉽게 원하는 데이터를 참조할 수 있도록 구현하였다.

시뮬레이션이 구동되는 동안 시뮬레이션 모델의 구성을 파악할 수 있도록 NetworkViewer 클래스를 개발하였다. 해당 클래스는 JUNG 라이브러리를 이용하여 (JUNG, 2013), 네트워크 및 그래프 형태로 컴포넌트의 연결 상태를 시각화하며, 컴포넌트에서 지정한 색채 및 형태로 표시하도록 구현하였다. 컴포넌트의 특정 속성값에 따라 색채를 변경하면, 시뮬레이션 진행 상황을 직관적으로 확인할 수 있으며, 확산문제와 같은 유한차분 해석방법에서는 후처리 없이 바로 결과를 시각화한다는 점에서 장점이 있다.

실시간으로 시뮬레이션 결과를 확인할 수 있도록 그래프를 실시간으로 그려주는 GraphViewer 클래스를 개발하였다. 그래프는 JFreeChart 라이브러리를 이용하여 구현하였으며 (JFreeChart, 2013), 컴포넌트의 속성 중 키워드 "plot_item"으로 지정한 속성에 대하여 시뮬레이션이 수행되는 동안 값을 추적하여 시각화한다. 그래프 기능은 주요 관심 영역의 속성값 변화 및 시스템 시뮬레이션을 수행하는 동안 오차의 수렴을 관찰하는 데 이용할 수 있다.

4. GASS2의 이용 규약

사용자가 GASS2를 이용하여 시뮬레이션을 수행하기 위해서는 크게 모델 작성과 모델 이용이라는 두 가지 역할로 구분되며, 각 역할별로 이용 시 필요한 규약을 정의하였다. 먼저, 시뮬레이션을 수행하기 위해서 모델 작성이 필요하며, 모델은 java

언어를 이용하여 시스템에서 제시하는 IGASSComponent 인터페이스를 구현해야 한다. 정보를 주고 받는 get(), set() 메소드는 범용적으로 이용이 가능한 기능이므로, 공통 기능을 미리 구현한 DefaultModel 클래스를 상속하여, 작업량을 최소화하도록 설계하였다. 연구자는 주어진 문제를 분석하여, 문제를 구성하는 요소들의 거동을 설명할 수 있는 메커니즘을 preProcess(), process(), postProcess() 메소드에 기술하도록 규정하였다. 다른 컴포넌트와 정보를 주고 받는 작업은 process() 메소드에 기술하되, 단위시간이 변하기 전에 준비작업은 preProcess()에서, 연결된 컴포넌트와 정보를 교환한 이후 해당 컴포넌트의 속성값을 갱신하는 작업은 postProcess()에 기술하도록 하였다.

모델 개발이 완료되었거나, 이미 개발된 컴포넌트가 있는 경우, 사용자는 시스템 시뮬레이션을 수행하기 위하여, 다음과 같은 모델을 이용하기 위한 프로시저로 프로그램을 작성해야 한다. 먼저 컴포넌트의 인스턴스를 생성하고, 각 인스턴스에 속성값을 부여한다. 인스턴스의 속성은 분석 모델 구현 시 필요한 변수들과 GASS2에서 특수하게 사용되는 키워드로 구분된다. 필요한 변수는 사용자가 임의로 추가할 수 있다. 시스템에서 사용되는 키워드로는 "x", "y", "size", "color", "shape", "plot_item" 등이 있으며, 각각 화면상의 x, y 위치좌표, 화면상에 표시되는 컴포넌트의 크기, 컴포넌트의 색상, 컴포넌트의 모양, 그래프에서 추적하고자 하는 변수를 의미한다. 개별 컴포넌트 인스턴스에 대한 설정이 완료되면, 컴포넌트를 연결하는 인접리스트를 정의한다. 일반적으로 방향이 없는 시스템의 경우에는 한쪽 방향만 지정하면, 반대방향의 연결도 자동으로 설정되도록 설계하였다. 컴포넌트의 연결 정보는 시뮬레이션 구동시 NetworkViewer를 통하여 화면상에서 확인 가능하므로, 누락된 연결 및 잘못 연결된 관계를 수월하게 검사할 수 있다. 컴포넌트의 연결성을 정의하면, 기본적으로 시뮬레이션 구동을 위한 설정은 완료된다. 추가적으로 시뮬레이션 구동화면을 저장할 수 있도록 특정 시점을 정의할 수 있고, 시뮬레이션 데이터를 데이터베이스에 저장한 후, csv 파일로 추출할 수 있다.

III. 연결된 저수조의 수위변화 모델 개발

1. 미분 방정식을 이용한 해석해

물이 차 있는 원통을 파이프를 연결할 경우, 서로 연결된 물통 사이의 위치에너지 차이에 의하여 물이 흐르게 되며 중국에는 연결된 모든 물통의 수위가 같아진다. 이와 같은 현상을 분석하기 위하여, 미분방정식의 풀이를 통한 해석해를 계산

하고, 이를 GASS2를 이용한 시뮬레이션 결과와 비교하였다. 원통형의 저수조가 주어지고, 저수조 가장 아래 부분에 유출구가 있다고 가정하였다. 물이 유출됨에 따라 변화하는 각 저수조의 수위 함수 $h(t)$ 를 구하는 문제로서, 유출구의 단면적을 A , 원통 수조의 밑면 면적을 B 로 정의하였다.

위와 같은 조건에서 저수조의 유출구로 분출되는 물의 유출속도는 위치에너지가 운동에너지로 바뀌게 되므로, 식 (1)와 같이 정리된다.

$$v(t) = C\sqrt{2gh(t)} \quad (1)$$

여기서, C 는 유출구의 크기 및 형태에 따른 형상계수인데, 본 연구에서는 이상적인 상태로 가정하여 $C=1$ 로 정의하였다.

두 개의 저수조가 연결되어 있는 경우, Fig. 4와 같이 도시할 수 있다. 두 개의 저수조가 연결된 경우에는 위치에너지가 두 저수조의 수위 차에 의해 결정되므로, 앞서 유출속도와 수위 함수에서 수위를 두 저수조의 수위 차이로 서술한다. 수치 해석 결과와 비교하기 위하여, 각 저수조의 초기수위는 $h_1(0) = 50$, $h_2(0) = 200$ 으로 가정하였다.

유출속도에 따라 유출되는 물의 부피 V 와 저수조 안의 부피변화가 동일하고, 연결된 두 수조의 물의 총량은 보존되므로, 초기값을 대입하여 연결된 두 저수조의 수위함수를 다음 식 (2)와 같이 정리하였다.

$$h_1(t) = \frac{250 - \left(\sqrt{150} - \frac{A}{B} \sqrt{2g} \times t \right)^2}{2} \quad (2)$$

$$h_2(t) = 250 - h_1(t)$$

주어진 문제를 2개의 저수조에서 3개의 저수조로 확장하면 관계식이 1개에서 3개로 늘어나면서 관계가 복잡해진다. 초기조건 $h_1(0) = 50$, $h_2(0) = 150$, $h_3(0) = 100$ 으로 가정하

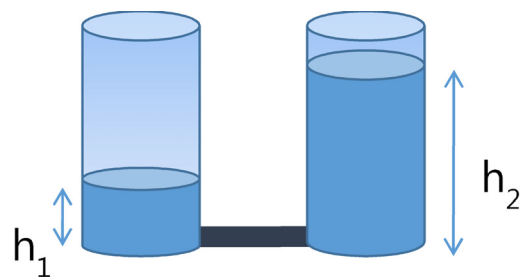


Fig. 4 Problem for water height function of two tanks

고, 각각의 단면적을 A_p , A_q , A_r 로 정의하여 지배방정식을 정리하면 다음과 같다. 식 (3)은 비선형 미분 연립 방정식으로 해석해를 구하기가 어렵다.

$$\begin{aligned}
 h_1(t) + h_2(t) + h_3(t) &= 350 \\
 \frac{dh_1(t)}{dt} &= \frac{\sqrt{2g}}{B} (A_p \sqrt{h_3(t) - h_1(t)} + A_q \sqrt{h_2(t) - h_1(t)}) \\
 \frac{dh_2(t)}{dt} &= \frac{\sqrt{2g}}{B} (-A_q \sqrt{h_2(t) - h_1(t)} - A_r \sqrt{h_2(t) - h_3(t)}) \\
 \frac{dh_3(t)}{dt} &= \frac{\sqrt{2g}}{B} (-A_p \sqrt{h_3(t) - h_1(t)} + A_r \sqrt{h_2(t) - h_3(t)})
 \end{aligned} \quad (3)$$

2. 저수조 컴포넌트의 구현

앞서 미분방정식의 풀이 과정을 통하여, 저수조의 수위함수를 계산하였다. 단순한 현상임에도 불구하고, 저수조의 개수가 늘어날수록 점차 해석하기 어려운 단계에 도달함을 확인하였다. 시스템 시뮬레이션은 이와 같이 복잡한 시스템 현상에 대하여, 앞서 미분방정식의 풀이와 같이 전체에 대하여 모델링하는 방식과 달리, 시스템을 구성하는 요소 단위로 분석하여 모델링하고, 개개의 요소를 통합함으로써 시스템의 거동을 모의한다.

저수조가 여러 개가 연결되는 시스템 문제는 저수조에서 수위량 변화 거동 규칙, 즉 위치에너지가 운동에너지로 변환되는 지배방정식은 동일하지만, 여러 개가 결합하는 과정에서 수위 함수가 서로 연관성을 가지게 되면서 복잡도가 증가하게 되었다. 따라서 저수조를 단위 요소로 모델링하여, 저수조 모델을 결합함으로써 문제를 손쉽게 해결할 수 있다.

저수조의 지배방정식은 연결된 수조와 상호간의 수위차에 의한 위치에너지를 운동에너지로 변환하는 과정이므로, 식 (1)을 기반으로 모델링하였다. 미분방정식은 시간 흐름에 따라 모의하는 GASS2 특성에 따라 차분식으로 정리하여 식 (4)와 같이 정리하여 적용하였다.

$$h_1(t) = h_1(t-1) + \sum_{k=2}^n \frac{\sqrt{2g} A}{B} \sqrt{h_k(t-1) - h_1(t-1)} \times \Delta t \quad (4)$$

GASS2 환경에서 컴포넌트 구현과정 중 주요 부분을 Fig. 5와 같이 도시하였다. 코드에서 process() 메소드에서 차분식을 이용하여, 단위 시간당 수위변화량을 구하며, postProcess() 메소드에서 이전 수위와 수위변화량을 더하여 값을 갱신하는 것을 확인할 수 있다. 이와 같이 단계를 분리한 이유는 process() 메소드는 각 컴포넌트가 동시에 수행되므로, 이 과정에서 서로 참조할 수 있는 값을 변경하게 되면 서로 다른 시

```

01: @Override
02: public void process(List<TreeNode> neighborhoods) {
03:     double h1 = getDouble("height");
04:     double transferTotal = 0;
05:     for (TreeNode n : neighborhoods) {
06:         IGASSComponent nm = n.getModel();
07:         double h2 = nm.getDouble("height");
08:         double trans = Math
09:             .sqrt(2 * 9.81 * Math.abs(h2 - h1)) * .01;
10:         if (h1 > h2) trans = -trans;
11:         transferTotal += trans;
12:     }
13: }
14: setDouble("trans", transferTotal);
15: }
16:
17: @Override
18: public void postProcess() {
19:     setDouble("height",
20:         getDouble("height") + getDouble("trans"));
21: }

```

Fig. 5 Program code for water height problem with linked tanks

점의 값을 참조하게 되므로 구분하였다.

행렬식을 이용하여 반복해석을 수행하는 모델은 GASS2 환경으로 비교적 쉽게 변환할 수 있는데, 이는 GASS2의 계산 과정이 행렬을 Jacobi 반복법으로 풀이하는 과정과 유사하기 때문이다. 여기에서 process()와 postProcess()를 분리하여 모델링을 수행하였는데, 분리하지 않고, 업데이트된 정보를 그대로 이용하게 되면, Gauss-Seidel 반복법과 유사한 형태를 지니게 된다.

IV. 연결된 저수조의 수위 해석

1. 수위 컴포넌트의 검증

연결된 저수조의 수위 해석을 위하여, 시스템을 구성하는 요소 컴포넌트를 검증하였다. 검증을 위하여 2개의 컴포넌트를 연결하여, 연결된 두 개의 저수조 수위변화를 모의하였다. 모의한 결과는 미분방정식의 해석해와 비교함으로써 검증하였다. GASS2 환경에서는 구현한 컴포넌트를 이용하여 연결 정보 및 초기 속성 정보만 입력하면, 시스템 시뮬레이션이 가능하므로, 초기 속성 정보로 각 저수조의 초기 수위를 200 cm와 50 cm로 입력하고, 방향성이 없이 완전 연결된 형태로 네트워크를 구성하였다.

모의 결과는 Fig. 6과 같이 시스템의 네트워크 구성도와 시뮬레이션 결과 그래프를 실시간으로 표시된다. 저수조 2개는 각각 2개의 붉은 원으로 표시되며, 두 원이 선으로 이어져 있는 것을 확인할 수 있다. 선이 연결된 컴포넌트 간에만 정보교환이 가능하며, 이후 다른 예제를 통하여 다루도록 하겠다. 시뮬레이션 결과 그래프는 컴포넌트에서 지정한 속성만 관찰할 수 있으며, 선택된 속성은 시뮬레이션 시간에 따라 변화되는 값을 동적으로 확인할 수 있다.

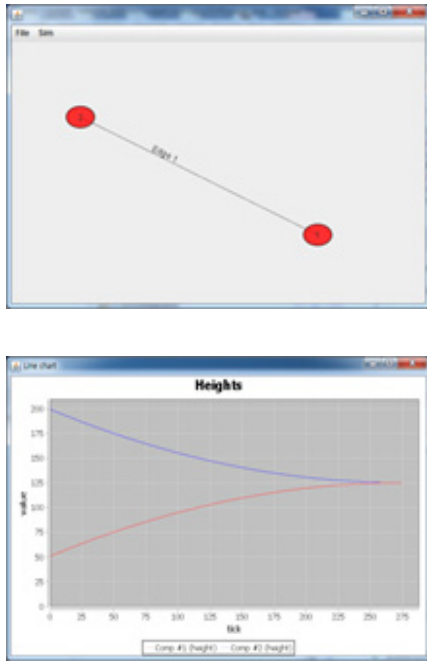


Fig. 6 Screenshots of GASS2 for two tanks problem

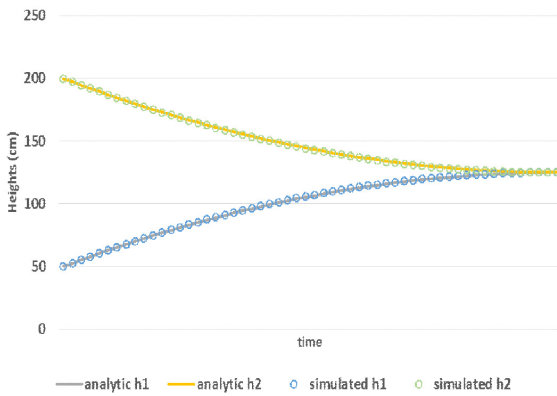


Fig. 7 Comparison of simulated result to analytic result

모의 결과를 검증하기 위하여, 미분방정식을 풀이한 해석해와 시뮬레이션 결과를 Fig. 7에 도시하였다. 그래프에서 확인할 수 있듯이 모의결과가 해석해와 동일하다고 판단할 수 있다. 하지만 차분식을 기반으로 모의하기 때문에 단위시간의 범위에 따라 오차가 발생하고, 특히 반복해법의 수렴반경보다 커지면, 발산할 우려가 있으므로 주의가 필요하다.

차분식에 사용되는 단위시간의 크기에 따라 모의 결과의 오차를 비교하기 위하여, 두 저수조의 수위가 동일해지는 시점을 기준으로 Fig. 8에 비교 정리하였다. 결과에서 x축은 단위시간을 log를 취하여 정리하였으며, y축은 해당 단위시간으로 시뮬레이션을 실행하였을 때, 평행상태가 이루어지는

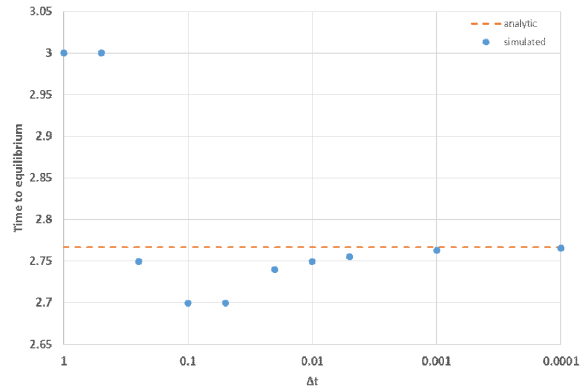


Fig. 8 Simulation model results by the delta time

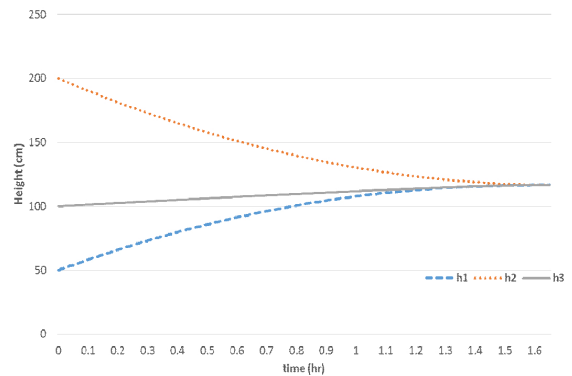


Fig. 9 Simulated water height for three tanks problem

시간을 도시하였다. 그리고 그래프 상에 점선은 해석해이며, 점으로 표시된 자료는 시뮬레이션을 통해 모의된 시간의 의미이다. 해석해의 결과는 2.766 시간에 동일한 수위를 갖게 되는데, 1시간과 0.5시간으로 모의했을 경우는 동일 수위가 되는 시점이 3시간인데 반해, 0.1시간 단위로 모의했을 경우, 2.7시간, 0.01시간 단위로 모의할 경우 2.75시간으로 산정되었다. 이는 단위 시간이 작아질수록 정해에 수렴해가는 과정으로 판단할 수 있다. 해석의 정확도를 위해서 단위시간을 작게 설정해야 하지만, 단위 시간의 크기는 전체 계산시간과 반비례하므로, 사용자가 적절한 단위 시간을 선정할 필요가 있다.

2. 모델 연결을 통한 시스템 시뮬레이션

서브시스템을 결합하여 시스템을 모의하는 시스템 시뮬레이션은 모델을 확장할 때 효용성이 드러난다. 앞서 검증한 모델을 바탕으로 저수조를 3개로 늘려서, 0.01시간 단위로 시뮬레이션을 수행한 결과를 Fig. 9와 같이 도시하였다. 초기 조건은 $h_1(0) = 50$, $h_2(0) = 150$, $h_3(0) = 100$ 로 설정하여, 저수조의 수위가 같아지기까지는 1.67 시간이 걸리는 것으로 모

의되었다. 저수조별 수위를 살펴보면, 1.45 시간이 경과한 시점에 첫 번째 저수조와 세 번째 저수조의 수위가 115.91 cm로 같아진 후, 전체 저수조의 수위가 116.67 cm로 동일할 때까지 수위 변화가 동기화된 행태를 확인하였다.

서로 연결된 저수조의 수위 변화를 모의한 결과를 통하여, 미분연립방정식을 풀이하는 방법에 비하여 연구자가 쉽게 현상을 분석할 수 있음을 확인하였다. 저수조 3개를 연결한 결과를 도시하였지만, 시뮬레이션 모델 구성을 고려하면 n개의 저수조로 확장 가능하다고 판단된다. 또한 위상학적으로 저수조의 연결을 정의하였기 때문에 2차원 격자 형태가 아닌 복잡한 형태의 네트워크로 구성된 연결을 갖더라도 문제를 정의할 수 있다.

3. GASS2의 개선효과 및 결과고찰

앞서 구성한 문제를 확장하여, GASS2의 개선효과를 살펴보고자 한다. 저수조 4개가 Fig. 10과 같이 연결되어 있는 구조에서 각 저수조의 초기 수위는 각각 50 cm, 300 cm, 150 cm, 30 cm로 정의하였다. 저수조의 밀면적 및 연결관의 면적은 동일하나, 일정 조건에 따라 연결관의 흐름량을 조절하도록 문제를 설정하였다. 저수조 4번의 수위가 50cm를 초과할 경우, 2번 저수조와 3번 저수조 사이의 연결관을 절반으로 닫고, 4번 저수조의 수위가 100cm를 초과할 경우에 2번 저수조와 3번 저수조 사이의 연결관을 완전히 폐쇄한다. 이러한 조건을 구현하기 위하여, 연결관의 개폐를 조절하는 컴포넌트를 개발하여 모델링하였다. 네트워크 뷰어에서 100번으로 표시하였다.

모의 결과는 Fig. 11과 같이 나타났으며, 0.46시간이 지난 시점에 4번 저수조가 50cm를 초과하면서, 저수조 2번과 3번 사이에 연결관의 단면적이 절반으로 줄어들고, 이에 따라 2번 저수조 수위변화의 변곡점이 나타났다. 2번 저수조에서 3번 저수조로 유입되는 물의 양이 줄어들면서, 3번 저수조에서 수위 강하 폭이 늘어나는 것을 확인할 수 있으며, 1.91시간에 3번 저수조는 최저 수위인 103.56 cm를 기록하고, 이후 2번 저수조에서 유입되는 양이 3번 저수조에서 유출되는 양보다 많게 되어 오히려 수위가 증가하는 모습을 나타냈다. 이후 2.66시간이 흐른 시점에 4번 저수조가 100 cm를 초과하게 되어 2번 저수조에서 3번 저수조로 연결된 관을 폐쇄하고, 2번 저수조는 218.99 cm로 수위가 유지되면서, 3번 저수조의 수위가 다시 떨어지면서, 1, 3, 4번 저수조의 수위가 3.46시간이 지난 시점에 103.67 cm로 동일하게 모의되었다.

구성한 문제는 식 (3)과 같이 연결관의 개폐 정도에 따라 면적이 달라지는 경우로서, 동일 시간 내에 전달량이 달라지게 되고, 이는 대상 저수조에서 어느 저수조와 연결되는 지에 따

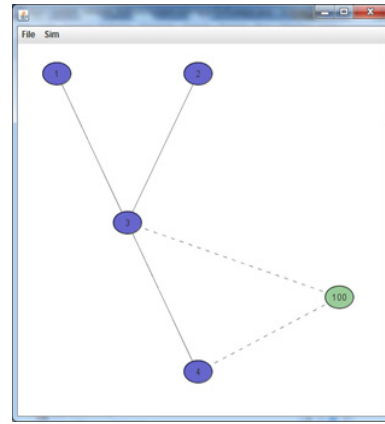


Fig. 10 Network viewer for 4-tank problem with control component

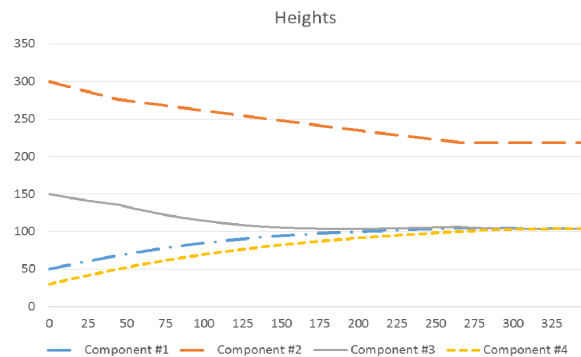


Fig. 11 Simulated heights for 4-water tank considering controlled pipe diameter

라 행위가 달라지는 형태가 된다. 다시 말해 연결되어 있는 컴포넌트 그룹 중 어느 한 경로를 선택해야 하는 이질소통문제로 치환되는데, 기존의 GASS는 컴포넌트 사이의 관계에 대한 속성을 정의할 수 없으므로, 이질소통문제를 모델링하기에 한계가 존재하였다. 또한 제어 컴포넌트를 위치시키기 위해서는 서로 떨어진 컴포넌트를 연결해야 하는데, 연결 구조가 복잡한 경우 이를 기하학적으로 정의하기 어려웠다. 이러한 문제를 본 연구에서 개선한 GASS2로 손쉽게 모의할 수 있음을 확인하였다.

V. 결론

본 연구에서는 시스템 시뮬레이터인 기존 GASS의 연결성과 소통구조 한계를 개선하여 GASS2를 개발하였다. 연결 구조와 통신방식을 개선하고, NetworkViewer, GraphViewer와 같은 편의기능을 추가하여 보다 쉽게 다양한 문제에 적용이 가능한 시뮬레이터를 구현하였으며, 이를 서로 연결된 저

수조의 수위 변화를 모의함으로써 활용가능성을 검토하였다.

GASS2는 생명체와의 비교를 통하여 연결 구조는 기하학적 연결과 위상학적 연결로, 통신방식은 동질소통구조와 이질소통구조로 구분하여 분석하였다. 기존의 GASS가 가지는 이질소통구조와 위상학적 연결에 대하여 한계를 극복하기 위하여, 이질소통구조는 벡터형식의 데이터 교환 규약을 설계하여 정의하고, 위상학적 연결은 컴포넌트 간의 연결 구조를 네트워크 기반으로 변경하여 모의하였다. 문제 설정 및 결과 확인을 위하여 네트워크와 그래프를 실시간으로 시각화하는 사용자 인터페이스를 구현하였다.

또한 개발한 GASS2를 검증하기 위하여 연결된 저수조의 수위 해석을 수행하였다. 저수조 1개에서 위치에너지와 운동에너지의 변환을 이용하여 지배방정식을 도출하고, 저수조가 2개 연결된 상태에서 일반화된 컴포넌트를 개발하였으며, 미분연립방정식의 해석해와 비교함으로써 모델을 검증하였다. 검증된 컴포넌트를 이용하여, 연결된 저수조 3개의 수위를 모의하였다.

해석해와 비교하여 유효숫자 내에서 동일한 결과를 확인하였으며, 시뮬레이션을 수행하는 시간 간격에 따라 수치해가 정해에 수렴하므로, 사용자가 적절한 시간간격을 결정할 필요가 있다. 모의 결과는 계산과 동시에 그래프에 반영되어 사용자에게 직관적인 피드백을 제공하였다. 또한 제어기능이 추가된 4개 저수조의 수위변화를 모의함으로써, GASS2의 개선된 기능을 확인하였다.

이상의 결과에서 시스템 시뮬레이터인 GASS2의 기본적 개념 및 구동과정을 제시하였으며, 저수조의 수위 모의라는 1계 미분 방정식을 풀이함으로써, 다양한 공학 문제에 적용하여 문제를 단순화하여 풀이할 수 있음을 보여주었다. GASS2는 복잡한 프로그래밍에 익숙하지 않은 연구자들이 쉽게 모델을 개발하고, 모의실험을 할 수 있는 플랫폼을 제공할 것으로 기대된다. 또한 향후에 2계 편미분 방정식으로 표현되는 보다 복잡한 공학문제로 확장할 수 있을 것으로 사료된다.

사 사

본 연구는 농림수산식품부 생명산업기술개발사업 기획과

제인 “농업생산기반시설 통합 관리시스템 개발”(과제번호: 311009-3) 연구비지원에 의해 수행되었습니다.

REFERENCES

1. Acock, B. and V. R. Reddy, 1997. Designing an object-oriented structure for crop models. *Ecological Modelling* 94(1): 33-44.
2. Goodliffe, P., 2007. Code Craft the practice of writing excellent code. ch. 1. (translator: Kim, Y. M., 2006). Hanbit Media Inc. pp. 3-22 (in Korean).
3. JFreeChart, 2013. <http://www.jfree.org/jfreechart> accessed 5 Oct. 2013.
4. Jones, J. W., B. A. Keating, C. H. Porter, 2001. Approaches to modular model development. *Agricultural Systems* 70(2-3): 421- 443.
5. JUNG, 2013. Java Universal Network/Graph Framework. <http://jung.sourceforge.net> accessed 20 Aug. 2013.
6. Kim, T., 2007. System simulator improvement using centralized component attribute control and diverse communication layers of material and information. MS. thesis, Seoul National University (in Korean).
7. Kim, T. and, J.J. Lee, 2007. GASS Improvement using diverse communication layers of material and information. *Journal of the Korean Society of Agricultural Engineers* 49(1): 101-109 (in Korean).
8. Lee, H. K., T. Kim, and J. J. Lee, 2012. Development of an object-oriented finite element model through iterative method ensuring independency of elements. *Journal of the Korean Society of Agricultural Engineers* 54(2): 115-125 (in Korean).
9. Suh, K., T. G. Kim, H. J. Lee and J. J. Lee, 2005. Simulation of agricultural marking based on GASS. *Journal of the Korean Society of Agricultural Engineers* 47(4): 3-12 (in Korean).
10. Van Evert, F. K. and G. S. Campbell, 1994. CropSyst: A Collection of Object-Oriented Simulation Models of Agricultural Systems. *Agronomy Journal* 86(2): 325-331.
11. Yi, H. J., 2003. Simplifying complexity and emerging self-organization of integrated agricultural system models. Ph.D. diss., Seoul National University.