

## 동시성 제어의 제어이론적 접근방법

○송 해상\*, 김 탁곤\*

\*한국과학기술원 전기및전자공학과

### A Control-theoretical Approach to Concurrency Control

Hae Sang Song\* and Tag Gon Kim\*

\*Electrical and Electronic Engineering Dept, KAIST

#### Abstract

This paper propose a control-theoretical approach to concurrency control in a database system. The approach is based not only on the separation between the data processing and the controller of a database system, but on the control-theoretical design of the concurrency controller. The analysis and design of the system become much easier than the conventional ones and the theoretical approach could play an important role to the other high level controller design. This approach is adopted in our database machine, COREDB, which is running with a great deal of satisfaction.

#### 1. 서론

지금까지 데이터베이스 시스템을 설계 구현하는 과정에서 제어부와 자료처리부는 서로 복합되어 설계되거나 잘 설계된 것은 모듈별로 다루어져왔음에도 불구하고 실제로 제어와 자료처리를 구분하여 다루는 제어이론적인 면에서는 잘 인식하고 있지 않다고 보인다. 실제로 많은 구현 예에서 명확한 구분이 지어지지 않았으며 설계자 또는 구현자가 이의 형태를 암시적으로 결정지어 왔다 [5].

동시성 제어란 데이터베이스 시스템내의 일의 단위인 여러 트랜잭션들이 동시에 같은 자료에 대한 작업을 행할때 서로간의 간섭으로 자료가 올바르게 읽은 값을 가지는 경우를 방지하는 것을 목적으로 여러 트랜잭션의 활동을 제어하는 것으로 정의된다. 동시성 제어의 용고 그룹은 순서화 가능성(serializability)으로 나타내며 이를 구현하는 알고리즘은 많은 연구가 되어왔으며 2단계잠금(2PL), 시간표지 순서화(TSO), 그리고 낙관적 방법(OOC) 등이 대표적인 알고리즘들이다[5].

트랜잭션은 자료를 읽고 처리하고 쓰는 일들을 해주는 자료처리 모듈을 이용하여 필요한 자료를 뽑아내거나 저장하는 일을 일괄 하는 프로세스, 또는 그 일 자체를 의미한다. 자료 처

리 모듈은 대개 사용자의 요구(예,SQL)를 받아들여 파싱(parsing)을 한 후 저장구조를 통해 수행하는 역할을 한다. 이는 데이터베이스 관리 시스템의 주목적이나 여러개의 트랜잭션이 동시에 같은 자료를 공유하므로 자료의 일관성을 유지하기 위해 동시성제어가 필요하게 된다. 이러한 데이터베이스 시스템의 하나로서 병렬 데이터베이스 컴퓨터인 COREDB가 개발되고 있다[6].

지금까지는 소프트웨어 공학적인 면에서 자료처리모듈과 동시성 제어모듈로 모듈화함으로써 어느정도 분석 및 설계의 용이성을 가져왔으나 좀 더 적극적으로 제어 이론을 수용하여 이를 들여다볼 생각은 못한 것 같다. 본 논문에서는 데이터베이스 시스템의 동시성 제어와 자료처리부분을 제어이론적인 관점에서 재조명하여 장단점을 비교분석해 보았다. 제어이론적 접근방법을 쓰게 되면 자료와 제어가 명확히 구분되고 필요한 제어요소 및 두 부분간의 인터페이스가 이론적으로 명확히 정의된다.

이 접근방법의 가장 큰 장점은 주어진 플랜트에 대해 목적은 같되 성능향상등을 위해서 제어를 마음대로 바꿀 수 있는데 있다. 만약 제어기와 입력 처리 플랜트가 서로 섞여 있다고 할 때 좀 더 나은 제어를 위한 방법이 제안되었을지라도 쉽게 제어 방법을 바꾸기가 힘들며, 기존의 제어방법조차도 이해하고 찾아내기가 힘들다. 그러나 제어기와 플랜트가 분리되어 있을 경우는 플랜트는 그대로 두며 단지 제어기를 바꾸기만 하면 되므로 시스템의 성능향상이 쉽게 이루어질 수 있다. 예를 들어 동시성 제어 알고리즘으로서 2PL, TSO, OOC등은 모두 같은 목적을 가지고 개발된 것이므로 주어진 플랜트와 연결경로는 그대로 두고 알고리즘, 즉 제어 알고리즘만 바꾸어 끼우는 것은 어렵지 않은 것이다.

제어와 자료 처리 과정을 분리하는 것은 마이크로 프로세서의 설계시 자료가 흘러가서 처리되는 부분과 자료의 흐름을 제어하는 제어부분을 독립적으로 설계하므로써 얻어지는 이득, 예를 들면 오류 정정의 용이성이라든가 구조를 명확히 이해할 수 있는 점 등, 많은 이득을 제어이론에서도 마찬가지로 얻을

수가 있는 것이다[4].

제 2장에서는 기존의 제어 이론에 대해 살펴보고 3장에서는 동시성 제어에 이 방법을 적용한 예를 보겠다. 마지막으로 4장에서 결론 및 향후과제를 기술하였다.

## 2. 기존의 제어 이론

시스템은 개개의 부분으로서 이룰 수 없는 어떤 목적을 달성하기 위해 부분이 모여서 함께 상호작용하면서 동작하는 것이라고 정의된다[2]. 여기서 시스템은 연속변수를 갖는 역동적 시스템만을 다룬다.

하나의 시스템은 크게 제어부분과 설비부분으로 나누어 생각할 수 있다. 설비부분은 주어진 입력에 대해 지정된 출력을 내는 부분이며 제어 부분은 설비부분을 주어진 목적에 맞게 움직이도록 입력을 조정하는 부분으로 본다. 설비부분은 수학적 모델로서 모델링되며 모델링이 실제에 근사할때는 설비와 설비모델을 구분하지 않아도 무방하다. 제어부분은 주어진 설비모델을 가지고 주어진 바람직한 행동을 하도록 입력을 조정하도록 설계되며 역시 수학적 모델로서 나타내어지고 이 모델을 바탕으로 구현된다.

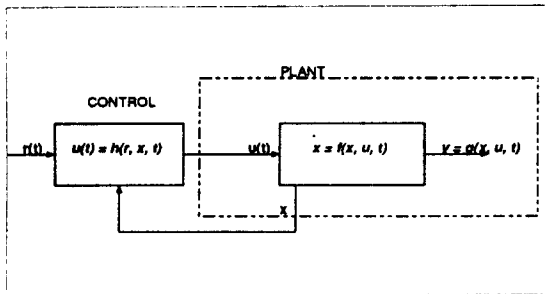


그림 1 일반적인 페루프 제어 시스템도

주어진 시스템의 목적을  $r(t)$  라고 하고 여기서는 기준신호라고 부른다. 설비부의 입력은 기준신호를 적절히 제어하여 나온 신호  $u(t) = h(r, t)$ 가 되며 설비부는 설비부내의 상태변수와 입력에 대해서 자신의 내부 상태를 변화시키며 이를  $\dot{x} = f(x, u, t)$  로 표현한다. 출력은 입력값과 상태변수의 함수로 주어지며  $y = g(x, u, t)$  로 표현한다. 여기서  $t$ 는 시간변수이며 시변 시스템까지 표현하기 위함이며 시불변 시스템의 경우는  $t$ 가 생략된다.

적절한 제어를 위해서 단지 주어진 기준값만을 가지고 제어하는 것을 개루프 제어라고 하고 제어를 위해서 설비부의 내부 상태값을 이용하는 경우를 페루프 제어라고 부른다. 보통의 제어는 설비부의 오차등을 보정하기 위해서 페루프 제어를 행하며  $u(t) = h(x, r, t)$  로 나타내어진다. 즉 제어기의 설계시 설비부의 내부 정보를 적극적으로 이용하는 것이다. 그림 1은 일반

적인 페루프 제어 시스템을 나타낸다.

시스템의 내부 상태를 나타내는 변수들의 집합을 상태변수집합이라고 하고 각각의 상태 변수가 가질 수 있는 범위를 정의역이라고 부르며, 상태변수에 각각의 정의역내의 어떤 값이 부여되어 있을 때 이를 하나의 상태라고 부른다. 일어날 수 있는 모든 상태의 집합을 상태공간이라고 부른다.

여기서 설비부 외부에서 관찰하여 그 값을 볼 수 있는 상태변수를 관찰가능변수라고 부르고 설비부 외부에서 값을 부여하여 내부의 상태를 변화시킬 수 있는 변수를 제어가능변수라고 부른다. 페루프 제어를 설계하는데 있어서 내부정보를 적극적으로 이용하기 위해서는 상태변수가 관찰가능해야 하며 제어가능변수를 가지고 있어야 제어가 가능하다.

PID 제어[1]는 페루프 제어의 대표적인 예로서 기준 신호는 보통 계단 입력을 주고 시스템이 기준 신호로 주어진 목적을 달성하는데 응답시간, 정확도, 과도응답시 오버슈트가 얼마나 작은지 등을 제한 사항으로 하여 제어기를 설계한다. 여기서 주목할 사항은 시스템의 목적은 정확히 계단 파형을 출력이 따라오는 것이며 이것은 논리적인 목적이 된다는 것과, 그러나 목적은 100% 만족할 수 없기 때문에 부가적으로 성능 평가 기준으로서 응답시간, 정확도 등을 어느 범위내에서 만족할 것인가 제한 사항을 가지고 PID 계수들을 조정하여 설계를 한다는 것이다.

제어는 기본적으로 감지(sensing)와 판단(decision making), 그리고 명령(actuation) 세가지로 이루어진다. 감지는 설비부의 어떤 변수값을 얻는 방법이며 이는 관찰변수 또는 이를 독립변수로 갖는 함수에 해당한다. 판단은 주어진 정보와 기준신호로 알고리즘적으로 이루어지며 이는 제어기의 알고리즘에 해당한다. 명령은 제어기의 판단을 실제 설비부가 움직이도록 명령을 발생시키는 것을 말하며 이는 제어변수에 값을 부여하는 작용이라고 볼 수 있다. 예를 들면 PID 제어제에서 감지는 설비부의 출력과 기준신호 사이의 오차를 감지하며 이 오차를 줄이기 위해 주어진 파라미터로 PID 제어 즉 PID 알고리즘으로 다음 명령 즉 설비부의 입력을 재조정한다.

본 논문에서는 위와 같은 개념을 데이터베이스 시스템에도 마찬가지로 적용해 보고자 한다.

## 3. 동시성 제어의 제어이론적 설계예: 2PL

동시성 제어는 데이터베이스 시스템에서 트랜잭션간 상호 간섭을 막고 트랜잭션의 일관성(consistency)을 유지하여 주기 위해 트랜잭션들이 자료를 읽거나 쓰는 것을 제어하는 중요한 역할을 한다.

트랜잭션은  $r, w, b, a, c$  의 다섯 가지 함수의 연속으로 정의되며, 항상  $b$ 로서 시작하고  $a$  혹은  $c$ 로서 끝이 난다. 이 사이에는  $r, w$ 의 임의의 연속으로 이루어지며 갯수에는 제한이 없다.

여기서  $r(w)$ 은 임의의 자료 읽기(쓰기)를 의미하고  $b$ 는 트랜잭션의 시작,  $a$ 는 트랜잭션의 취소,  $c$ 는 트랜잭션의 정상적인 끝맺음을 의미한다.

트랜잭션의 일관성은 트랜잭션의 목적에 따라 다르나 보통 여러 트랜잭션이 동시에 동작을 하지만 자료의 접근은 어느 시점이나 단 한 트랜잭션만이 행하는 것같이 가상적인 환경을 제공하는 것을 의미한다. 또한 한 트랜잭션이 수행도중 만들어낸 결과를 다른 트랜잭션이 접근하지 않도록 하여야 하는데 이 두 가지, 즉 일관성과 독립성, 가 동시성 제어의 목적이 된다.

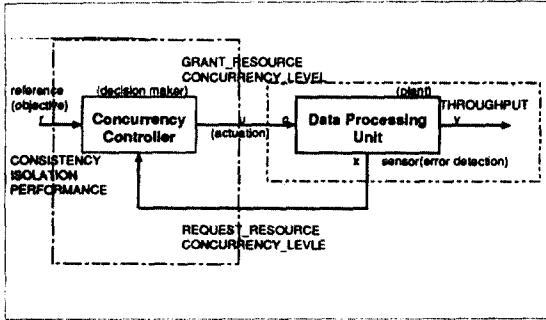


그림 2 제어이론적 관점에서의 동시성제어

데이터베이스 시스템의 중요분야인 자료 처리부는 사용자로부터 필요한 정보를 요구하는 질의를 받아 가지고 있는 데이터베이스로부터 필요한 정보를 꺼내주는 역할을 담당한다. 우리는 제어이론적 관점에서 이 자료처리부를 설비부라고 생각할 수 있다.

각 트랜잭션은 자료 처리부를 이온하히 필요한 정보를 읽기도 하고 쓰기도 한다. 그러나 같은 자료를 공유하여 쓰기 때문에 일관성 및 독립성을 보장 받아야 하여 그러기 위해서 자료 처리부는 동시성 제어기에 의해 자료의 읽고 쓰미 제어된다. 이 부분은 제어 이론적 관점에서 제어부에 해당하며 일관성과 독립성은 제어부의 목적이 된다.

동시성 제어부는 자료 처리부의 일관성 및 독립성을 보장하기 위해서는 자료 처리부의 내부 상태, 특히 처리부의 어느 부분에서 읽거나 쓰는가를 알지 않으면 제어를 할 수 없다. 즉 관찰 가능한 변수가 존재해야 한다. 다시 말하면 펌프 제어가 아니면 주어진 목적을 달성할 수 없다는 것을 알 수 있다.

만약에 이러한 변수가 없다면 자료처리부내의 읽거나 쓰는 부분에 소프트-감지기(soft-sensor)를 설치하여 이를 찾아내어야 한다. 여기서 감지된 부분들은 잠재적인 오류로 볼 수 있으며 제어기에서 이를 목적에 맞도록 조정하여주어야 한다.

오류 감지를 하여 이를 목적에 맞는 올바른 것으로 조정하였다 하더라도 자료 처리부에 이를 시정할 수 있도록 제어할 수 있는 방법이 없다면 제어기의 의미는 역시 없다. 따라서 우리는 자료 처리부에 대해 제어가능한 변수를 가져야 하며 만약에 자료 처리부에 이러한 것이 제공되지 않았다면 이를 추가하여

야 한다.

일단 제어변수와 관찰변수가 설정이 되면 우리는 제어기를 목적에 따라 바꾸는 것은 어렵지 않다. 동시성제어에서는 현재 2PL로 구현되어 있는 알고리즘을 TSO, OOC등으로 대체하여도 명령이나 상태변수, 입력은 일정하기 때문에 자료처리부는 독립되어 처리할 수 있다.

또하나 주목할 것은 자료 처리부는 자료 처리 그 자체의 목적에 더욱 충실해 질 수 있다는 것이다. 제어기능과 자료 처리기능이 복합되어 설계되는 경우는 두 가지가 섞여 명확히 시스템을 이해하기도 어렵고 설계 또한 복잡하지만 각자가 따로 설계되면 각각의 목적에 따라 더욱 충실한 결과를 얻을 수 있다. 표 1은 기존의 제어이론과 동시성제어부의 상관관계이다.

표 1. 기존의 제어이론과 동시성제어의 비교

| 항목              | 제어이론   | 동시성제어                |
|-----------------|--------|----------------------|
| objective       | $r(t)$ | consistency          |
| error detection | sensor | $r, w, b$ req. point |
| control alg.    | PID    | 2PL, TSO, OOC        |
| actuation       | $u$    | allow $r, w, c, a$   |
| output          | $y$    | THROUGHPUT           |
| feedback        | $x$    | $r, w, b$ req.       |

#### 4. 결론

소프트웨어 시스템을 제어이론적 관점에서 바라다보아 제어부와 설비부로 구분하였고 제어이론적 관점에서 제어부를 설계함으로써 다음과 같은 장점을 얻을 수 있었다.

- i) 시스템의 이해 및 설계가 명확하다.
- ii) 기존의 제어이론을 적용하여 제어기의 문제를 해결할 수 있다.

향후 일반적인 소프트웨어에 대한 자료처리부, 제어부, 그리고 상호간의 연결에 대해 제어 이론적 형식론적 모델링과 검증(validity)을 위한 뼈대(framework)를 구축하고자 한다.

#### 참고문헌

- [1] B. C. Kuo, *Automatic Control Systems*, ch.6, Prentice-Hall, 1992.
- [2] C. G. Cassandras, *Discrete Event Systems*, ch.1, Irwin, 1993.
- [3] H. S. Song, *Implementation of 2PL with Adaptive Load Control*, KAIST M.S.E.E. thesis, 1991.
- [4] J. L. Hennessy, et. al., *Computer Architecture: A Quantitative Approach*, ch.5, Morgan Kaufmann, 1990.
- [5] P. A. Bernstein, et. al., *Concurrency Control and Recovery in Database Systems*, ch.2-3, Addison-Wesley, 1987.
- [6] 한국과학기술원, *하이퍼큐브형 데이터베이스 컴퓨터 개발(III)*, ch.6, 채신부 연구보고서, 1993.