'96 춘계 학술발표회 논문집
한국원자력학회

Supervisory control 기법을 이용한 비상운전지원시스템 개념설계

김명기, 홍승열
전력연구원,한국전력공사

정명진
전기및전자공학과,한국과학기술원

요약

최신 제어이론의 한 분야로 주목을 받고있는 이산사건시스템((Discrete Event System) 제어이론을 원자력발전소의 비상운전지원시스템 설계에 도입하였다. 이산사건이론을 비상운전절차에 적용하면 보다 체계적이고 조직적으로 비상운전지원시스템을 구축할 수 있다는 장점이 있다. 본 논문에서는 발전소 운전상태 및 운전원의 조치사항을 이산사건시스템으로 모델링하여 제약조건에 맞는 수퍼바이저를 구성하고 이를 바탕으로 비상운전지원시스템을 구축하는 방법을 제시하였다.

1. 서론

미 원자력규제위원회에서는 중대사고 종결계획으로 개별발전소에 대하여 PSA/IPE 수행결과를 바탕으로 사고관리수립을 요구하고 있다. 중대사고는 여러가지 현상학적 요소들이 상호작용하고 있고 사고진행 과정도 복잡하기 때문에 기존 운전절차서와 같은 형태의 비상운전절차로써는 효과적인 사고관리를 담당하기에는 한계가 있을 것으로 판단된다. 따라서 비상운전시 보다 안전하게 발전소를 관리하기 위해서는 사고관리절차서 뿐만아니라 비상운전지원시스템과 같은 전문가 시스템이 필요하다 하겠다. 비상운전지원시스템이란 개념적으로 말하면 원자력발전소에서 발생 가능한 모든사고에 대해 사고경위 및 전개상황을 미리 파악하고 있어 어떤 중대사고가 발생하더라도 이에 적절한 운전방법을 제시할 뿐아니라 필요시 자동적으로 사고를 대처해나가는 시스템을 말한다. 이런 비상운전지원시스템은 PSA/IPE 결과물과 발전소 운전변수 및 기기상태변수를 취득하여 이를 바탕으로 최적의 운전을 지시하는 수퍼바이저 (Supervisor)로 모델링할 수 있다. 본 논문에서는 발전소 운전상태 및 운전원의 조치사항을 이산사건시스템(Discrete Event System) 모델링 방법을 사용하여 수퍼바이저를 구성하고 이를 바탕으로 비상운전지원시스템을 개념적으로 구축하고자한다.

2. 수퍼바이저 제어이론

- 플랜트 P를 다음과 같이 두 개의 언어 (language)로 모델 : $P = (L_p, M_p)$
  - 언어 : 각 사건(Event)을 하나의 알파벳으로 나타내며 사건의 추이는 복수개의 알파벳으로 표현할 수 있는 데 이를 스트링이라하고 이의 집합을 언어라 한다. 즉 플랜트는 이산사건추이의 집합 즉 언어로 표현된다. Σ = {스트링}
  - L_p : P의 모든 운전형태를 나타낼 수 있는 Prefixed-closed 언어
  - M_p : 운전의 최종단계를 나타내는 언어 (Marked language) , $M_p \subseteq L_p$
- 수퍼바이저 : $S = (L_s, M_s)$, 플랜트와 같이 언어로 표시

- 컴포지션(Composition) : 플랜트 P와 수퍼바이저 S가 동시에 사건을 수행시키는 것을 컴포지션이라 함

  $P \| S = (L_{p\|s}, M_{p\|s})$, $\Sigma = \Sigma_p \cup \Sigma_s$

  $L_{p\|s} = \{s \mid del(\Sigma-\Sigma_p)(s) \in L_p \cap del(\Sigma-\Sigma_s)(s) \in L_s\}$

  $= del(\Sigma-\Sigma_p)^{-1}(L_p) \cap del(\Sigma-\Sigma_s)^{-1}(L_p)$

  $M_{p\|s} = del(\Sigma-\Sigma_p)^{-1}(M_p) \cap del(\Sigma-\Sigma_s)^{-1}(M_p)$

  단, $\Sigma = \Sigma_p = \Sigma_s$, $P \| S = (L_p \cap L_s, M_p \cap M_s)$

- 컴플리트 (complete) 수퍼바이저 : 제어할 수 없는 사건 (uncontrollable event)에 대해서 추적하는 능력이 있는 수퍼바이저
  - 수퍼바이저 S가 플랜트 P에 컴플리트함 $\Leftrightarrow P \| S = P \| (L_s\Sigma_u^*, M_s)$
  - 플랜트 P가 수퍼바이저 S에 컴플리트함 $\Leftrightarrow P \| S = (L_p\Sigma_u^*, M_p) \| S$
- 넌블럭킹(Nonblocking) : 플랜트 $P = (L_p, M_p)$ 가 넌블럭킹하다 함은 L_p의 어떤 스트링도(string)도 M_p안에 있는 마크트 스트링(marked string)에 컴플리트하다는 의미
- 수퍼바이저 설계 : 플랜트 P 가 주어질 때 수퍼바이저 S를 구하는 문제
  - $L_{p\|s} \subseteq L_{spec}$
  - S 는 P에 대해 컴플리트하여야 함
  - P || S 는 넌블럭킹하여야 함, $L_{p\|s} = \overline{L_{p\|s}}$

  ($L_{p\|s}$ 안에 있는 언어는 플랜트내 마크트 언어와 수퍼바이저 마크트언어의 일부로 표현됨 예 : $L_{p\|s}$ ={abc}, M_p = {, ..eabcd..,}, 일 때 수퍼바이저의 Ms 는 {,..eabcg..,} 임)
- 제어성(Controllability) :

  K 는 L_p 에 대해서 다음 조건이 만족하면 제어가능(controllable)하다고 함.

  조건 : $\overline{K}\Sigma u \cap Lp \subseteq \overline{K}$
- 수퍼바이저 : Ssup ($K^{\uparrow}$)

  $supC(M_p \cap L_{spec})$ 이 공집합(nonempty)이 아닐 때 S_{sup}는 다음과 같이 구해진다.

  $S_{sup} = (\overline{supC(M_p \cap L_{spec})}, supC(M_p \cap L_{spec}))$

  단, supC(M) = ∪{K : K⊆M 그리고 L에 대해서 제어가능일 경우}
- The greatest fixpoint of operator Ω :

  $\Omega(K) = E \cap sup\{T : T \subseteq \Sigma^*, T=\overline{T} \text{ and } T\Sigma_u \cap L_p \subseteq \overline{K}\}$, $E = M_p \cap L_{spec}$
- K1, K2 가 넌컴플리팅(nonconflicting)이면 $L_{spec} = L_{spec1} \cap L_{spec2}$에 대한 global least restrictive 수퍼바이저는 모듈라 수퍼바이저 S_1, S_2의 컴퍼지션으로 구할 수 있다.

3. 수퍼바이저 구축

용어정의

먼저 수퍼바이저를 구축하기 위하여 다음과 같은 용어를 정의한다.

사건 (운전조작행위 및 시스템 및 기기상태)

- Controllable Event : 발전소의 안전계통 작동 명령 및 회복조치
- Uncontrollable Event : 안전계통중 일부 기기의 고장으로 계통기능상실
- Illegal State : 원자로 노심손상사고
- Observable Event : 안전계통 작동의 응답중 수퍼바이저가 감지 할 수 있는 사건
- Unobservable Event : 발전소 상태를 판단하지 못하는 사건

제한조건

- 안전제한조건 : 플랜트가 피해야 할 시컨스, 사고가 진행되다가 최종적으로 노심손상

사고에 이르는 시퀀스를 제거 (부적절한 상태를 제거)

· Liveness 제한조건 : 플랜트가 제약조건을 만족시키기 위해서 반드시 거쳐야할 시컨스

Implicit Livenass 제한조건 : 각 안전계통의 운전시 필요한 제한조건 (예:펌프를 기동할 때에는 펌프 전단의 밸브는 항상 열려져 있어야 한다)

Explicit Livenss 제한조건 : 사고 종류마다 결정되어지는 제한조건 (예:각종 초기 사건에 대해서 해당 안전계통은 적절히 동작되어야 한다)

수퍼바이저 구조

비상운전수퍼바이저는 플랜트로부터 각종 입력신호를 받아 플랜트의 상태를 진단하고 이에 적절한 계통의 운전방법을 발전소에 지시하므로써 발전소를 안전한 상태로 유지하는 기능을 가지고 있다. 이런 수퍼바이저의 구성방법은 먼저 원자력발전소에 일어날 수 있는 모든사고에 대하여 안전계통의 작동순서 및 작동상황(Observable Event라 함)을 판단하여 각 운전절차 및 계통마다 오토마타를 구성한다. 그리고 각각 구한 오토마타에 대해서 상호 컴포지션을 수행하고 그 결과에 대하여 불필요한 사건트레이스(Event Trace)를 모두 제거하면 우리가 원하는 수퍼바이저를 만들 수 있다. 그러나 이와 같이 하나의 수퍼바이저로 원자력발전소의 비상운전절자를 담당하는 것은 비효율적이며 시스템이 복잡해지면 모델링자체가 거의 불가능하게 된다. 그러나 원자력발전소의 운전은 상호 의존성이 강하여 하나의 수퍼바이저로 원자력발전소의 운전절차를 관장한다는 것은 분석상 매우 어려우므로 본 논문에서는 사건수목별로 수퍼바이저를 구성하는 분산수퍼바이저 (그림 1)를 제안하였다. 분산수퍼바이저는 3가지로 구성되어 있는데 첫 번째로 사고를 진단하는 사고진단 부분, 두 번째로는 각 사고에 적절히 대응하는 콘트롤러(PSA 측면에서 보면 사건수목에 해당)부분, 세 번째는 각 안전계통의 운전을 담당하는 부수퍼바이저 부분이다.

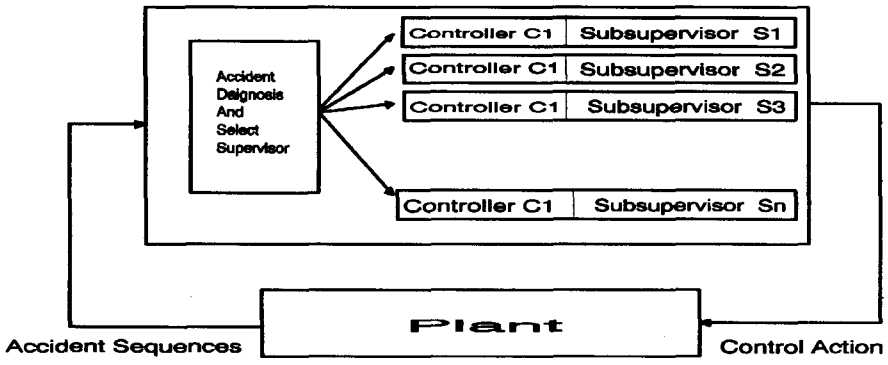


그림 1. 비상운전 분산수퍼바이저 구조

사고진단 : 발전소현상을 분석하여 발전소 사고종류를 판별하고 이에 맞는 수퍼바이저 선택

컨트롤러 C : 플랜트 P로 부터 사고의 상황을 전달받아 이에 적절한 사고대처 수습운전방안을 제시하는 기능을 가지고 있으며 안전계통의 작동순서를 결정하는 고수준(High level)의 수퍼바이저

부수퍼바이저 S_i : 콘트롤러 C가 안전계통의 작동 명령을 내리면 부수퍼바이저 S_i는 안전계통의 동작을 지시 (안전계통마다 독립적인 부수퍼바이저를 구축)

부수퍼바이저

먼저 비상운전수퍼바이저의 여러개의 부수퍼바이저 중 소형냉각재사고가 발생하였을 때 고압안전계통의 작동에 대한 부수퍼바이저를 설계하고자 하며 분석의 간략화를 위해서 고압안전주입 계통

은 그림 2와 같이 단순화 하였다. 고압안전주입 계통은 트레인 1의 펌프가 작동하다가 기기의 고장으로 인해 냉각수를 공급하지 못할 경우 트레인 2로 운전모드가 전환되어 펌프 2가 작동하여 냉각수를 공급하게 된다. 각 기기의 운전상태를 이산사건으로 모델링을 하기위해서 기본 운전모드를 고려하여야 하는 데 탱크, 펌프는 정상운전 상태를 고려하였으며 밸브는 정상상태 운전모드 이외에 고장 및 보수를 포함시켰다. 기기별 오토마타는 그림 3에 나타나 있으며 이에 대한 컴플리트 오토마타는 다음과 같이 3단계로 구한다.

1) 플랜트의 고수준(high level) 입장에서 기본적인 것만 모델링 : $E = (L_E, M_E)$
2) 저수준(low level)서 여러개의 부시스템(sub system)차원에서 각각 오토마타 구축 :
   M_i의 두단계에서 구한 프로쎄서를 결합 : $P = (L_p, M_p)$
   $L_p = del(\Sigma-\Sigma_E)^{-1}(M_E) \cap (M_1 \cup M_2 \cup ... \cup M_n)$, $L_p = M_p$
3) 위에서 구한 고수준과 저수준 오토마타를 컴포지션하여 컴플리트 오토마타를 구함

예) 밸브에 대한 L_p 구성방법

- 밸브의 기본적인 명령 : c_repair_valve
- 밸브의 기본적인 상태응답(response) : r_valve_closed, r_valve_opened, r_valve_failed
- 밸브의 기본언어 (fundamental language) :
  Σ_E = {c_repair_valve, r_valve_closed, r_valve_opened, r_valve_failed}
  M_E = {r_valve_opened*r_valve_closed+r_valve_failed*c_repair_valve}
- 외부에서 운전원이 밸브에 가할 수 있는 기본 프로쎄스, ϕ
  $\phi 1$: 밸브를 개방하라 할 때 이에 대한 밸브의 응답 시퀀스
  M1 = {c_open_valve*r_valve_open+c_open_valve*r_valve_closed}
  $\phi 2$: 밸브 닫음에 대한 시퀀스
  M2 = {c_close_valve*r_valve_closed}
  $\phi 3$: 밸브 보수에 대한 시퀀스
  M3 = {c_repair_valve}
- 컴플리트 오토마타 : L_p
  $L_p = del(\Sigma-\Sigma_E)^{-1}(M_E) \cap (M_1 \cup M_2 \cup ... \cup M_n)$, $L_p = M_p$
  = {(c_open_valve*(r_valve_opened*c_close_valve*r_valve_closed
  +r_valve_failed*c_repair_valve))*}

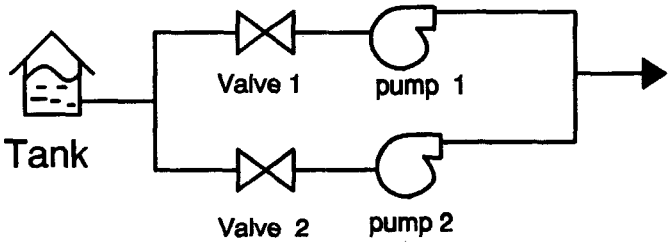
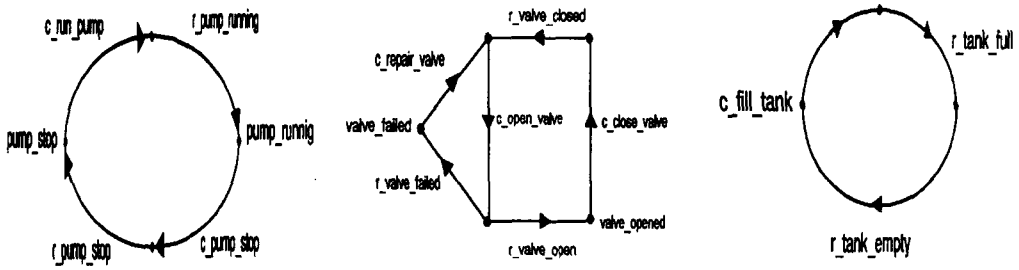


그림2. 간략화된 고압저온관안전주입 계통도

이와 같이 동일한 방법으로 펌프, 탱크에 대하여 컴플리트 오토마타를 구하고 계통의 컴플리트 오토마타를 구하면다음과 같다.

P_{valve} : Σ_{valve} = {c_open_valve, c_close_valve, c_repair_valve, r_valve_closed,
r_valve_opened, r_valve_failed}

P_{pump} : Σ_{pump} = {c_run_pump, c_stop_pump, r_pump_stop, r_pump_running}
P_{utank} : Σ_{tank} = {c_fill_tank, r_tank_empty, r_tank_full}
$P = L_{valve} \parallel L_{pump} \parallel L_{pump}$, $\Sigma = \Sigma_{valve} \cap \Sigma_{pump} \cap \Sigma_{tank}$
위에서 구한 계통의 컴플리트 오토마타에 제한조건에 따라 운전을 지시하는 수퍼바이저는 다음과 같이 구한다. 예를 들어 제한조건 (Implicit Liveness)이 "펌프를 기동시킬 때는 탱크가 full인 상태에서 밸브를 열고 펌프를 기동시킨다" 이라고 하자. 그러면 제한조건을 표현하는 오토마타를 그림 4와 같이 구하고 (이는 계통에 대해 컴플리트하고 넌블록킹함) 이를 계통의 컴플리트 오토마타와 컴포지션하여 수퍼바이저를 구한다.
후보(Candidate) 수퍼바이저 : S′ , $\Sigma' \subseteq \Sigma$
Σ' = {c_open_valve, c_run_pump, c_fill_tank, r_valve_closed, r_pump_running,
c_repair_valve, r_tank_full}
수퍼바이저 : S = S′ || P



펌프 오토마타 밸브 오토마타 탱크 오토마타

그림3. 기기 오토마타

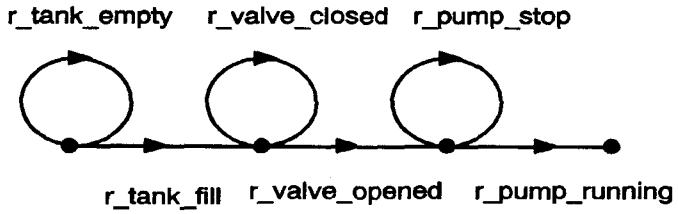


그림 4. 후보수퍼바이저 (S′)

그러나 고압안전주입계통은 두 개의 트레인으로 구성되어 있고 트레인마다 각각의 운전제한조건 (Implicit, Explict Liveness)이 있으므로 트레인마다 오토마타를 구하면 다음과 같다. 그림 5는 트레인 1의 운전방식과 기기가 고장일 경우 트레인 2로 운전전환을 보여주는 오토마타(T1)을 보여주고 있다. 그림 6은 트레인 2에 대한 오토마타(T2)로서 트레인 2의 운전상황과 트레인 2에 기기가 고장일 경우에 시스템이 dead lock 상태로 가는 것을 나타내고 있다. 따라서 이런 제한조건을 만족시키는 고압안전주입계통을 관리하는 수퍼바이저는 다음과 같이 구할 수가 있다.
수퍼바이저 : S = S || T1 || T2

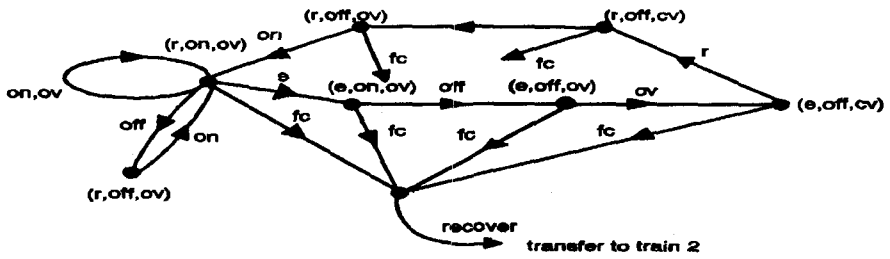


그림 5. 트레인 1에대한 오토마타

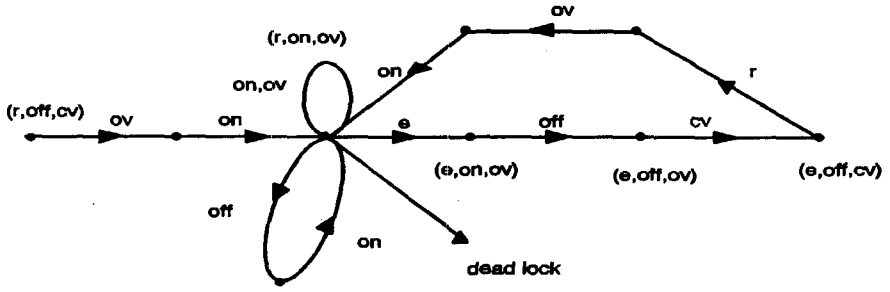


그림 6. 트레인 2에 대한 오토마타

이상과 같은 방법으로 발전소의 모든 안전계통에 대한 부수퍼바이저를 구축할 수 있으며 안전계통의 동작 순서에 대한 콘트롤러와 사고진단부분도 이와 같은 방법으로 구할 수 있다. 이렇게 구한 각각의 수퍼바이저를 그림1에서 나타난 분산 수퍼바이저의 구조에 적용하면 발전소 전체를 제어하는 비상운전지원시스템을 구축할 수 있다.

4. 결론

본 논문에서는 분산제어이론을 이용하여 원자력발전소의 비상운전지원시스템구축 방법을 제시하였다. 비상운전지원시스템은 하나의 수퍼바이저로 간주할 수 있으며 운전절차가 매우 복잡하기 때문에 단일의 수퍼바이저보다 여러개의 부수퍼바이저로 구성되어 있는 분산수퍼바이저 구조를 제안하였으며 간단한 고압안전주입계통의 비상운전절차에 대하여 이산사건모델링방법을 통하여 수퍼바이저를 구성하였다.

5. 참고문헌

1. Peter J. G. Ramadge, W. M. Wonham, IEEE, vol. 77, no. 1, pp. 88-98, Jannuary 1989
2. Discrete Event Dynamic Systems Analyzing Complexity and Performance in the Mordern World, Edited by Yu-Chi Ho, IEEE Press, 1991
3. S. Balemi, "Supervisory Control Of a Rapid Thermal Multiprocessor", IEEE Tran. on AC., Vol. 38, NO. 7, July 1993