

선내탑재 의사결정지원 시스템을 위한 발라스트 최적화 알고리즘에 관한 연구

신성철*

*목포해양대학교 해양시스템공학부 교수

A study on a ballast optimization algorithm for onboard decision support system

Sung-Chul Shin*

*Division of Ocean System Engineering, Mokpo Maritime University, Mokpo 530-729, Korea

요 약 : 선박 침수 사고의 경우, 선박의 운용 책임자가 취할 수 있는 대응방안이 한정되어 있어 정확하고 신속한 의사결정을 위해서는 기존의 안전관련 시스템을 활용한 효율적인 의사결정 지원 시스템이 필요하다. 수밀 및 준수밀 문, 격벽 밸브, 배수 펌프 등과 같이 침수 사고 시 작동하는 대부분의 시스템들은 침수가 선박 전체로 전파되는 것을 막도록 충분한 구획분할 정도를 확보하는데 목적이 있다. 침수 시나리오가 파국적이지 않다고 가정하더라도 발라스트 탱크의 사용은 침수 전파 방지와 선박 안정성을 향상하기 위한 매우 효과적인 방안이 될 수 있다. 본 논문에서는 침수 손상 시 최적의 대응방안을 위해 채워져야 하는 발라스트 탱크들을 선정하고, 각 발라스트 탱크의 수위를 결정하는 최적화 알고리즘을 기술한다.

핵심용어 : 선박안전, 의사결정지원 시스템, 손상 복원성, 인공지능, 최적화

ABSTRACT : Because there are only a limited number of means of action that are available for the master to pursue in the event of flooding, onboard decision support system has been required. The majority of systems activated during a flooding emergency (such as watertight and semi-watertight doors, bulkhead valves, dewatering pumps etc.) almost exclusively aim to restore a sufficiently high level of subdivision to prevent flooding from spreading through the ship. Even though assuming the flooding scenario is not catastrophic, the use of ballast tanks can be an additional and very effective tool to ensure both prevention of flooding spreading and also improve ship stability. This paper describes an optimization algorithm devised to choose the set of ballast tanks that should be filled in order to achieve an optimal response to a flooding accident.

KEY WORDS : ship safety, onboard decision support system, damage stability, artificial intelligence, optimization

1. 서 론

2000년 5월에 IMO 사무총장은 대형 여객선의 안전에 대한 전반적인 재검토를 요청했다. 모든 여객선은 해상에서 인명 안전에 대한 국제 협약(SOLAS)을 지켜야하므로 최근 건조된 대형 여객선들이 현재의 안전 기준에 부합하는 것은 분명하다. 그러나 응급상황시 대응조치를 비롯한 각종 안전 규약들이 지금의 대형 여객선들이 건조되기 이전에 제정되었기 때문에 SOLAS나 국제만재흡수선 조약 등을 재고할 필요가 있다(IMO, 2003).

대형 여객선의 안전 관리에 대한 새로운 필요성에 따라 많은 선박 운항자들이 조작 절차를 재평가해보고 선박의 안전 시스템의 효율성을 면밀히 분석하기 시작했다. 이러한 활동의 일환으로 2002년에 COMAND라는 프로젝트가 진행되었다. COMAND는 침수 사고시 의사결정을 보조할 수 있는 통합 의사결정 지원 시스템(Decision support system, DSS) 개발을 목적으로 하고 있다.

DSS는 기본적으로 다음과 같은 세 가지 기능을 제공해야 한다:

- ① 모니터링 기능: 관련 데이터를 수집하고 그것을 간단하면서 이해하기 쉬운 방법으로 시각화함으로써 선박 내 각

*정회원, scshin@mmu.ac.kr 061)240-7264

중 시스템의 현 상태를 지속적으로 관리하는 기능. "현재 상태가 어떠한가?"라는 질문에 대한 답이 되는 기능.

② 예측/진단 기능: 현 상태가 어떻게 진전될 것인지를 신속하게 평가하는 기능. 모니터링 기능에 의한 데이터에 기반을 둔다. "현 상태가 어떤 결과를 초래할 것인가?"에 대한 근본적인 해답으로 가능한 한 정확해야 한다.

③ 조언 기능: 현재 응급 상황을 완화시키기 위해서 최선의 대응책을 제안하는 기능. "어떤 조치를 취해야 하며 우선순위를 어떻게 두어야 하는가?"라는 질문에 대한 답이다.

COMAND 프로젝트는 DSS의 기능구조를 상세화하고 이를 구현하는데 목적이 있다. 본 논문에서는 이 가운데 손상 복원성을 위한 조언 기능 한 가지만을 다룬다.

현재까지 그러한 기능을 제공하는 응용프로그램이 없다. 반면에 그러한 응용프로그램을 구축하는데 기반이 되는 많은 도구와 방법론들이 개발되었다. COMAND에서 수행되고 본 논문에서 기술하는 연구 결과는 최종적으로 DSS가 달성해야 하는 목표에 필수적인 이론적 기반이 될 것이다.

2. DSS 기본 전략

일반적으로 침수 사고에 대응하기 위한 조치는 몇 가지로 한정되어 있다. 크로스플러딩(cross-flooding)과 카운터플러딩(counter-flooding), 배수와 수밀 혹은 준수밀 문과 밸브의 차단 등이 그 예이다. 발라스트(ballast) 탱크와 힐링(heeling) 탱크의 조작에 초점을 맞추어 유사한 침수 시나리오에 대해서는 그 당시의 대응조치들이 동일하게 적용될 수 있다는 점을 일반화할 수 있다. 여기에는 보다 확장된 개념으로서 연료나 화물 등을 복원성을 고려하여 버리지 않고 그대로 둔다든지 하는 것도 포함될 수 있다. 예를 들어 Fig. 1과 같이 세 구획 침수의 경우, T1, T3 탱크를 채우고 T4 탱크를 비우는 것이 손상 당시의 복원성 향상을 위해 최선의 대응방안이었다면 이와 유사한 침수 사고의 경우 같은 대응방안을 사용할 수 있을 것이다.

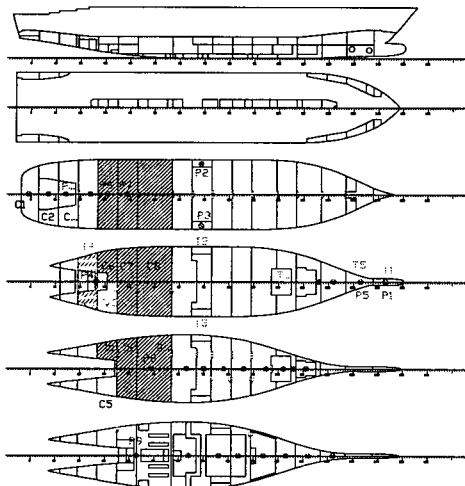


Fig. 1 Example of a three compartment flooding case

이 같은 문제의 특성은 수집된 사례 혹은 데이터들을 이용하는 인공지능 기법인 사례기반추론(Case-based reasoning, CBR)을 COMAND 조언 기능에 대한 기본적인 전략으로 채택하게 된 이유가 된다. 최선의 카운터플러딩 방안을 제공하기 위해 COMAND 시스템은 사례베이스를 검색하고, 가장 가까운 사례를 선정해서 최근접 사례의 해를 추출한 후, 현재의 사고 사례와 차이점을 반영하여 최근접 사례의 해를 수정한다.

CBR은 비슷한 상황에서 유용한 지식을 사용하여 새로운 문제를 해결하기 위한 방법으로 사용된다. CBR의 전반적인 과정은 Fig. 2와 같이 네 단계(Aamodt, and Plaza, 1994 and Kolodner, 1993)로 표현할 수 있다:

- ① 검색(RETRIEVE): 최근접 사례를 검색
- ② 재사용(REUSE): 최근접 사례의 문제를 풀기에 유용했던 정보와 지식을 추출
- ③ 수정(REVISE): 추출된 초기해의 교정
- ④ 저장(RETAIN): 미래 문제 해결에 사용되도록 현재 사례를 사례베이스에 추가

유사한 하나의 혹은 여러 개의 미리 해결된 사례를 검색하고 이로부터 적절한 초기해를 얻은 후 초기해의 개선을 통해 초기해를 수정하는 방법으로 새로운 문제를 해결할 수 있다. 새롭게 해결된 문제는 다시 기존의 사례베이스에 추가된다.

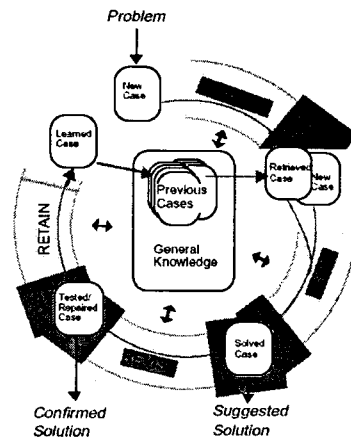


Fig. 2 The CBR cycle

CBR은 규칙기반 추론 방법과 달리 정형화 되지 않은 문제에 적합하다. 전문가 시스템을 비롯한 규칙기반 추론법은 논리 추론과 문제 영역의 정밀한 표현을 필요로 한다. 규칙기반 추론법에 필요한 명시적인 영역 지식 혹은 규칙 도출이 경우에 따라서는 어려울 수도 있다. 이러한 문제에 대해 CBR은 적절한 해를 제공할 수 있다.

상기의 이유로 본 연구에서 선박 침수사고시 카운터플러딩의 경우 CBR이 적합한 것으로 판단되었다. 선박 침수 사고 통계를 통해 각 구획의 침수 확률 및 그 조합의 확률을 분석해 본 결과 제한된 수의 사고 사례를 대상으로 사례베이스를 구축하는 것이 가능하였다. 사례베이스 내에서는 각 구획의 ID와 탱크 상태 변수를 통해 효과적으로 사례를 표현할 수 있었다.

카운터플러딩 문제에서 CBR 사용과 관련된 핵심 사안은 다음

의 세 가지이다:

- ① 사례베이스내에서 특정 사례의 적절한 표현법과 검색법: 인덱싱(indexing) 문제
- ② 유사한 상황에서 관련된 해를 재사용하기 위해 선택된 임의의 사례에 일반적으로 적용될 수 있는 적절한 수정 가능
- ③ 최종 선택된 해의 만족도를 평가할 수 있는 적절한 방법

3. 자료구조와 인덱싱

인덱싱 문제는 각 사례를 표현하기 위한 적절한 변수의 선택으로 귀결된다. 침수된 선박의 상태는 매우 간단하고 한정된 수의 변수들로 표현이 가능하다:

■ $C_i(t)$: 시간 t 일 때 각 구획 i 에서의 침수된 물의 양을 나타낸다. 이러한 형태의 정보는 $[C_i]$ 로 보다 간단한 형태로 나타낸다. 이것은 시간 t 일 때 침수된 구획의 전체 목록을 표현한다.

■ Draught, KG, Trim: 침수사고 이전의 적재상태를 나타낸다.

상기의 사고상태 변수들이 표현하는 침수 사고는 발라스트를 사용하여 복원성을 높일 수 있고, 그 과정은 제한적이다. 그렇지 않으면 침수 범위와 초기 적재상태를 갖는 특정 사례와 연관된 해가 상이할 것이다. 이러한 이유로 탱크와 구획의 조합이 중복되지 않는 것을 가정하면 다음의 상태 변수를 사례 정의에 추가하는 것이 가능하다:

■ $T_j(t)$: 시간 t 에 채워진 탱크 j 의 체적을 나타내고 $[T_j]$ 로 간단하게 사용한다. 이것은 초기 상태에서 손상을 받았거나 사용 중에 있어서 채워졌다고 간주할 수 있는 탱크 목록을 나타낸다.

수밀 문의 작동과 침수 전파에 대한 그 효과를 무시하면 임의의 침수 시나리오를 표현하기에 상기의 변수들로 충분하다. 이러한 관점에서 최근접 이웃 탐색에 기반한 간단한 사례 매칭(matching) 알고리즘으로 가장 적합한 사례를 검색하는 것이 가능하다. 현 상황의 새로운 사례의 각 필드와 사례베이스에 있는 사례들의 필드를 비교하는 과정에 적용되는 사례 매칭 규칙은 다음과 같다:

- 두 필드(예를 들면 $[C_i]$)가 정확히 일치하면 그 필드의 전체 가중치를 준다.
- 두 숫자(Draught, KG, Trim과 같은)가 유사하면 전체 범위에 비례해서 어느 정도 서로 가까운지를 먼저 계산한 후 그 숫자 필드의 가중치를 곱한다.
- 각 필드의 총 매치 가중치 합계는 두 사례의 근접도를 나타낸다.
- 가장 높은 매치 가중치를 갖는 사례가 사례베이스로부터 선택된다.

각 변수에 대한 적절한 필드 가중치는 사례 검색이 유사한 손

상 형태에 좀 더 민감하도록 선정한다. 이것은 일반적으로 카운터플러딩의 해가 초기 적재상태보다는 손상 형태에 더 민감하기 때문이다. 이때 고정된 가중치를 사용하기 보다는 구획 크기와 실제 침수 구획 체적의 상대적인 중요성을 반영하도록 정규화된 값을 사용하였다.

사례베이스에서 각 손상 사례에 관련된 해는 두 가지 변수로 표현된다:

- $[T_j]$: 최적의 평형상태가 되도록 채워져야 하는 발라스트 탱크들의 아이디 목록을 나타낸다. 이 목록에 들어 있지 않는 모든 탱크는 비워져야 한다.
- S_j : 각 탱크의 동작상태를 나타낸다. 변수의 값으로 이진값을 생각할 수 있다. 예를 들면, 조작 가능한 상태는 TRUE로, 고장난 상태는 FALSE로 표현한다. 그렇지만 0부터 1사이의 연속적인 실수값이 각 발라스트 탱크의 동작상태를 평가함에 있어 불확실성을 표현하는데 보다 합리적이다.

```

case instance case1 is
  compartments = [C1, C3];
  tanks = [T6];
  draught = 0.3;
  KG = 0.5;
  trim = 0.0;

solution is
  if S6 = TRUE
  then
  [];
  else
  if S4 = TRUE
  then
  [T6, T4];
  else
  if S2 = TRUE
  and
  if S3 = TRUE
  then
  [T6, T2, T3];
  else
  no_solution;
  end;
end;

case instance case2 is
  compartments = [C1, C3];
  tanks = [];
  draught = 0.2;
  KG = 0.5;
  trim = 0;

solution is
  if S4 = TRUE
  then
  [T4];
  else
  if S2 = TRUE
  and
  if S3 = TRUE
  then
  [T2, T3];
  else
  no_solution;
  end;
end;
  
```

Fig. 3 Example of a typical case

손상 사례의 해는 Fig. 3에서 볼 수 있듯이 각 옵션에 속한 탱크의 동작 상태를 반영하여 우선순위가 매겨진 탱크 조합의 목록 $[T_j]$ 로 주어진다. 가장 좋은 옵션은 목록의 가장 윗부분에 온다. S_j 의 값에 대한 조건들은 이미 채워져 있어서 비워야 하는 탱크와 비워져 있어서 채워야 하는 탱크 모두를 고려한다. 일반적으로 올바른 조건을 선택하는 과정은 자동화하기가 용이하다. 이것은 다음의 몇 가지 기본적인 규칙을 따르기 때문이다:

- 해가 인덱스 목록에 포함된 탱크 중 하나가 비워져 있

는 것을 요구한다면 S_j 의 대응값에 대한 조건은 TRUE이어야 한다.

■ 해가 인덱스 목록에 포함된 탱크 중 하나가 채워져 있는 것을 요구한다면 그 탱크에 대해 어떤 조건도 필요로 하지 않는다.

■ 해가 인덱스 목록에 포함되지 않은 탱크 중에 하나가 채워져 있는 것을 요구한다면 S_j 의 대응값에 대한 조건은 TRUE이어야 한다.

■ 해가 인덱스 목록에 포함되지 않은 탱크 중에 하나가 채워져 있는 것을 요구한다면 그 탱크에 대해 어떤 조건도 필요로 하지 않는다.

가장 유사한 사례의 해가 구해지면 실제 침수 문제와 선택된 사례와의 차이를 고려하여 해를 수정하게 된다. 나중에 이러한 수정 과정을 상술하겠지만 현재 문제와 선택된 사례간에 차이가 적다면 수정단계를 생략할 수 있다.

수정 이후에 해의 절차를 최적으로, 그리고 자유표면효과에 의한 불안정성없이 수행할 수 있는지 해를 검증하게 된다. 초기 발라스트 탱크 배치로부터 수정된 해에 의해 기술된 배치로 가기 위해 필요한 절차를 수립하기 위해 사용된 규칙기반의 충돌 해결 알고리즘에 따라 검증 결과가 결정된다. 검증 알고리즘은 변화의 추이를 반영하면서 카운터플러딩 절차를 찾기 위한 것이다. Fig. 4는 이 같은 개념을 설명하고 있다.

Present Status(from monitoring at time t)
$T_4(t)=50.8 \wedge S_4(t)=TRUE$
$T_5(t)=0.0 \wedge S_5(t)=FALSE$
$T_j(t)=0.0 \wedge S_j(t)=TRUE \forall j \wedge j \neq 4 \wedge j \neq 5$
Desired Status (from CBR)
$T_2=100.0$
$T_3=500.0$
$T_5=250.0$
$T_j=0.0 \forall j \wedge j \neq 4 \wedge j \neq 5$
Actions Required (after verification and conflict resolution)
empty tank 4 then
fill tank 5 then
fill tank 2 then
fill tank 3

Fig. 4 A snapshot of system variable values

채워야 하는 탱크 목록과 비워야 하는 탱크 목록은 모든 탱크 용량의 현재 상태와 원하는 상태의 비교를 통해 즉각 도출된다. 한편, 문장 생성은 "empty tank"나 "fill tank"와 같은 표준 정적 문자열의 비교에 의해 쉽게 다루어질 수 있다. 모니터링 시스템으로부터 데이터가 들어왔을 때 이러한 두 가지 절차는 동적으로 수행될 수 있다.

그러나 필요한 조치의 적절한 순서는 각 순열의 선박 복원성에 대한 영향을 조사(검증)함으로써 찾을 수 있다. 이러한 분

석은 반복 수행될 필요는 없다. 검증 동안에 처음 채우거나 비워야 하는 탱크는 GZ 곡선 상에서 갖는 이득을 기반으로 선택된다. 분석 결과는 $T_j(t)$ 의 실제값에 따라 달라지므로 경험식식에 의해 수행되어서는 안 되고, 사례베이스의 해에 통합되어야 한다.

해를 구성하는 S_j 와 $[T_j]$ 는 손상의 결과로 혹은 탱크 운용을 위한 시스템이 고장을 일으킨 경우와 같이 채워진 탱크를 비울 수 없을 때 발생하는 문제를 이미 고려하고 있다. 비어있는 탱크를 채울 수 없는 것도 마찬가지이다.

4. 해의 수정

최근접 사례의 검색을 통해 선택된 해는 고려중인 현재의 침수 문제에 완전히 적합하지 않을 수 있다. 선택된 최근접 사례와 현재 사례의 차이가 클수록 또한 사례베이스의 구성이 적절치 못할수록 해의 부적합성은 심화될 것이다. 일반적으로 사례기반추론에서 수정 기능은 수정으로 인해 제한됨이 없이 선택된 해에 포함된 정보를 최대한 활용할 수 있어야 하고, 사례의 차이점을 반영하여 적절한 변화를 가해야 한다.

특정 인덱스와 앞서 기술한 것과 같은 해의 형태가 정해지면, 카운터플러딩 문제에 적절한 수정 기능을 찾는 것은 손상 선박의 최종 평형 상태를 최적화하기 위해 채워져야 하는 발라스트 탱크를 선택하기 위한 적절한 기준과 방법론을 정의하는 것이 된다. 대부분의 선박이 선내에 설치된 탱크의 수가 작기 때문에 가장 적절한 방법론은 언덕 오르기 탐색(hill-climbing search)이다. 탐색을 돕기 위해 적절한 기준 함수 평가시 정적 복원력 계산을 채용했다.

이러한 관점에서 최근접 사례의 해를 구성하는 탱크 조합은 언덕 오르기 탐색을 최적화하기 위한 초기해로서 사용 가능하다. 이것은 수정기능이 선택된 최근접 사례에 포함된 지식을 사용하는 것을 의미한다. 또한 작동상태변수(S_j)는 고장난 탱크를 제외시킴으로써 탐색 공간을 줄여준다.

상기의 해를 위한 조건 구조는 다양한 대안들의 상호 배타성을 내포하고 있다. 다시 말해서, 만약 해 A가 어떤 조건 S_j 를 갖고 있고 두 번째 해 B보다 상위에 있다면, 해 B에 포함된 조건은 해 A의 것보다 덜 제한적이어야 한다. 그렇지 않다면 해 B는 선택되지 않을 것이다. 최소 탱크 체적의 요구조건과 더불어 이러한 고려사항은 초기 해탐색 공간을 상당히 줄여준다.

수정기능으로서 가장 중요한 사항은 각각의 해에 대한 양호성의 기준이 될 적절한 기준 함수를 찾는 것이다. 손상된 선박의 생존성이 하나의 기준으로 사용될 수 있다. 관련된 IMO 규칙을 보면, 손상 선박의 생존성은 잉여 GZ 곡선의 주요 특성인 복원력 범위, 최대 복원력(GZ_{max}), 초기 평형 각도의 함수인 복원성 지수 s_0 로 나타내고 있다. 현재의 규칙에서는 복원성 지수 s_n 는 트림(trim)이나 잉여 건현은 명시적으로 고려하

지 않는다. 더구나 s 는 0에서 1사이의 값을 갖고, 경사(heel)에 대한 계단함수(stepwise function)이다. 그러므로 실제 생존성이 같지 않음에도 불구하고 s 값이 동일하게 1로 최대값을 갖게 하는 발라스트 배치가 여러 개 있을 수 있다. Fig. 5는 임의의 손상 시나리오이고 Fig. 6과 Fig. 7은 각각 s 값을 1로 하는 서로 다른 발라스트 대응안이다. 이러한 이유로 s 값만을 최대화하기 보다는 생존성에 영향을 미치는 다른 요인을 조사하였다.

Floating position
Case: IN11/P5-6.1.0-2
Stage: PROGRESSIVE Phase: EQ

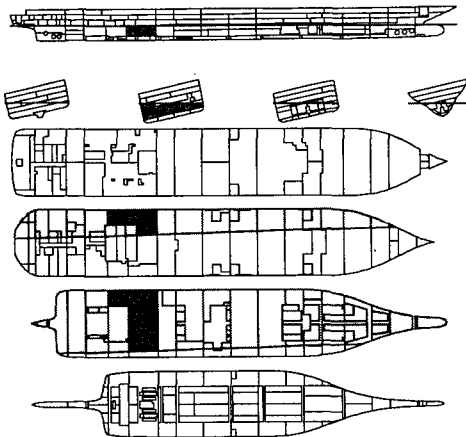


Fig. 5 Example of a damage case: $s = 0.257$; trim = 1.213 m; heel = 11.4 deg; draught = 7.653 m

Floating position
Case: 1ST-3/P5-6.1.0-2
Stage: PROGRESSIVE

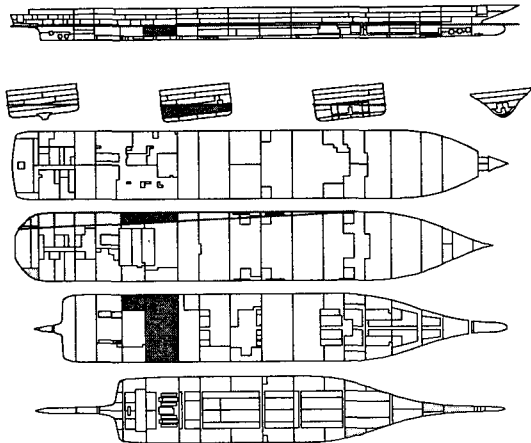


Fig. 6 Possible ballast configuration solution: $s = 1.000$; trim = 1.246 m; heel = 5.9 deg; draught = 7.276 m

Floating position
Case: 1ST-4/P5-6.1.0-2
Stage: PROGRESSIVE

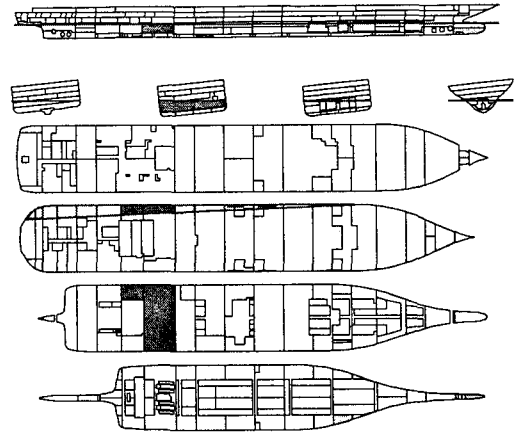


Fig. 7 Alternative ballast configuration solution: $s = 1.000$; trim = 1.061 m; heel = 5.9 deg; draught = 7.276 m

일반적으로 GZ곡선상의 많은 파라미터들이 더 낮은 KG값으로 이득을 얻는지는 논쟁의 여지가 있다. KG값이 낮아질수록 복원력 범위와 GZmax, GZ 곡선의 면적은 모두 증가하지만, 반면에 초기 평형 각도는 감소하는 경향이 있다. 그러나 발라스트를 사용하는 주요 효과 중에 하나가 선박의 무게중심 CG의 위치를 변경시키는 것이므로 KG는 최소화해야 하는 파라미터임에 틀림이 없다. 또한 충분한 복원력이 있는 상태에서 점진적 침수를 막기 위해 평형상태의 트림, 흘수, 경사각을 최소화함으로써 잉여 건현을 최대화하는 것이 좋다. 이 같은 견지에서 최소 잉여 흘수, 트림, 경사각 및 KG는 추가적인 평가 기준으로 고려할 수 있다.

주어진 발라스트 배열을 달성하는데 필요한 모든 발라스트 탱크를 채우거나 비우는데 드는 시간이 일정한 역할을 한다는 것에 주의해야 한다. 복원성과 선박 자세의 측면에서 동일한 두 개의 발라스트 배열에서 좀더 짧은 시간에 달성할 수 있는 배열이 선호된다고 가정할 수 있다. 주어진 발라스트 배열을 달성하기 위해 필요한 시간은 최종 배열뿐만 아니라 순간순간의 예기치 못한 일에 따라서도 달라진다. 이러한 이유로 최적화 파라미터로서 시간 변수의 처리는 검증과정에서 직접 계산하는 것이 가장 적합하다.

CBR 시스템에서 발라스트 시간을 평가할 때 사용될 수 있는 한 가지 유용한 가정은 펌프 용량이 동일하다고 생각하여 더 많은 수의 탱크와 더 큰 체적의 탱크가 포함된 대안일수록 시간이 오래 걸린다고 간주하는 것이다. 물론 더 큰 체적과 더 많은 수의 탱크를 채우는 것은 흘수에 부정적인 영향을 줄 수 있다. 이것은 평형상태에서 건현에 영향을 끼친다. 이러한 이유로 총 탱크 용적에 대한 흘수의 의존성을 반영하는 최소화 함수는 발라스팅 시간의 견지에서도 이득이 있다.

어떤 사례가 초기에 많은 용적의 탱크가 채워져 있었다면 위 가정은 성립하지 않는다. 이 때문에 인덱스와 해에 포함된 모

든 j 값에 대해 비우거나 채워야할 탱크의 총 용적의 형태 ($Tanks = \sum \max(T_j)$)로 평가기준 파라미터를 추가할 수 있다. 탱크를 채우는 속도는 탱크마다 크게 다르기 때문에 부가적인 정규화 승수를 곱한 $Tanks = \sum a_j \times \max(T_j)$ 의 형태로 변경할 수 있다.

해면상태(sea state)가 높은 상황에서 잔여 굽힘 모멘트를 고려해야 함에도 불구하고, 자유롭게 떠있는 사례에 대해 구조 강도는 높은 해면상태일 때만 문제가 되는 점을 감안하여 굽힘 모멘트는 무시할 수 있다.

상술한 일반적인 사항들을 기반으로 이상적인 카운터플러딩 발라스트 배치안을 탐색하는 문제를 간단한 다목적 최적화 문제로 정식화할 수 있다. 초기 탐색공간은 s 값을 최대로 하는 사례만을 포함하도록 제한되고, 목적함수는 흘수, 트림, 경사각, KG 및 탱크 용적의 향으로 이루어진다. 각 항목들은 두 가지 첨자로 구분되는 항을 포함하는데 첨자 f 로 표시된 것은 손상후 카운터플러딩 조치를 취하지 않고 최종평형 상태를 이루었을 때의 값을 의미하고, 첨자 b 로 표시된 것은 카운터플러딩 조치를 취한 후 평형상태에 이른 값을 의미한다.

각 평가함수 값은 흘수, 트림, 경사각, KG 범위와 카운터플러딩에 가용한 탱크의 총 용적의 값으로 정규화함으로써 계산된다. 이때 $Tanks_{max} = \sum a_j \times \max(T_j)$ 이다. 최종 평형상태에 도달할 수 없을 경우에는 해당 파라미터의 가정된 최대 값을 할당하도록 되어 있다.

이상의 원리를 최적화 과정으로 정식화하면 Fig. 8과 같다.

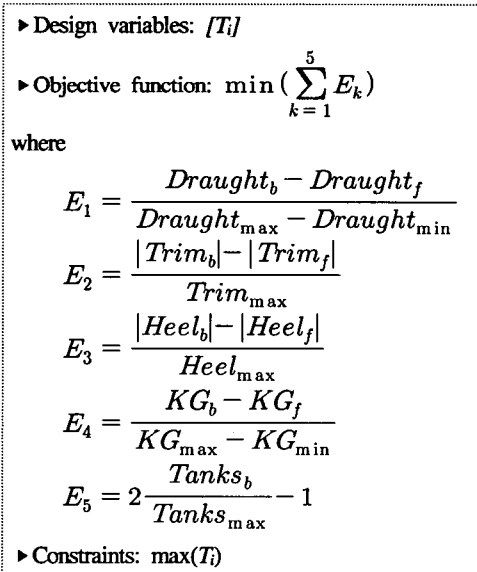


Fig. 8 Ballast tank optimization

모든 E_k 는 -1에서 1사이의 값을 취한다. 다섯 개의 각 평가항목들은 같은 가중치 1을 갖는다. 즉, 동일한 중요성을 갖는다. 각 항목이 손상 선박의 최종 평형상태에 미치는 영향을 조사한다면 목적함수는 아래와 같이 바뀔 수 있다.

► Objective function: $\min \left(\sum_{k=1}^5 \omega_k E_k \right)$

여기서 높은 가중치 값은 해당 파라미터의 중요성을 증가시키게 될 것이다.

5. 결 론

본 연구를 통한 결론은 다음과 같다:

1. 침수사고 시 대응방안 수립을 위한 의사결정 시스템은 사례기반 추론 기법을 이용할 수 있다.
2. 침수 상태를 표현하는 시스템 변수들은 흘수, 트림, 경사각, KG 등으로 간단하게 표현되며 이것은 앞으로 사례베이스를 구축하는 출발점이 될 것이다.
3. 카운터플러딩을 위해 필요한 탱크와 그 탱크의 용량을 결정하는 문제는 최적화 문제로 정식화 할 수 있다.

본 논문에서는 침수 사고를 다루기 위한 선내탐재 의사결정 지원 도구를 만드는 것이 목적인 연구과제 COMAND에서 수행된 연구 중 일부를 설명하였다.

본 논문에서 제시된 대부분의 개념들은 보다 더 깊은 연구를 필요로 하고, 혁신적인 프로그램 코드로서 구체화되어야 한다. 연구 과제를 수행하면서 많은 어려움과 질문들이 새롭게 던져졌고, 이로 인해 새로이 해결해야할 난점들이 많은 것이 사실이다. 그러나 본 논문을 통해 정리된 사항들은 COMAND를 수행해 나가는 중요한 토대가 될 것이며, 모든 논의의 기본틀을 형성할 것이다.

참 고 문 헌

- [1] IMO SLF 46/INF3(2003).
- [2] Kolodner, J. (1993), "Case-Base Reasoning", Morgan Kaufmann Publishers, Inc.
- [3] Aamodt, A. and Plaza, E. (1994), "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", AI Communications, IOS Press.
- [4] Barletta, E.R. (1994), "A Hybrid Indexing And Retrieval Strategy For Advisory CBRSystems Built with ReMind", Proceedings of the European Workshop on Case-Based Reasoning.
- [5] Aamodt A. and Veloso M. (1995), "Case-Based Reasoning Research and Development", Proceedings ICCBR-95, Lecture Notes in Artificial Intelligence, 1010. Springer Verlag.
- [6] Voss A. (1996), "How to Solve Complex Problems with Cases", Engineering Application of Artificial Intelligence, Vol. 9, No. 4. Elsevier Science Ltd.