

# 특집

## 북극항로 운항선박용 항해안전 지원시스템 프로토타입 개발

이한진, 강국진, 정성엽(KRISO)  
정성현, 최현, 최성우(동강엔텍)

### 1. 서론

북극항로 운항을 위해 해양 환경 - 특히, 기상정보 - 을 고려한 항로 설계에 대한 개념 모델이 "북극항로 운항선박용 항해안전 지원시스템 개념설계" 논문에서 제시되었으며, 기본 개념은 Fig. 1과 같다. Araon과 육상 항해 지원 서버간의 데이터 통신을 위해 위성통신을 고려하며, KIOST의 기상정보서버를 통해 확보된 기상 정보를 바탕으로 최적의 북극항로를 계산하여 전달한다.

본 논문에서는 앞서 제시된 북극항로 운항선박용 지원시스템의 설계를 각 모듈별 프로토타입으로 구현하여, 해당 기능을 검증하였다. 향후 구현된 프로그램은 실선(Araon)에 탑재되어 인천을 출발하여 북극으로 항해 중에 그 기능을 검증할 계획이며, 실선 테스트에서 발생한 오차 및 파라미터를 반영하여 성능 최적화를 실행할 예정이다.



Fig. 1 Conceptual architecture of voyage planning system for Araon

### 2. 본론

북극항로 운항선박용 항해안전지원시스템은 3개의 모듈로 Fig. 2와 같이 연계되어 있다.

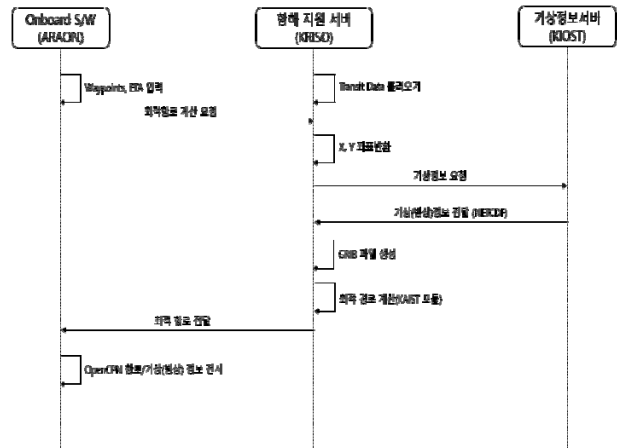


Fig. 2 System sequence diagram

#### 2.1 Onboard S/W (Araon 탑재 S/W)

Onboard S/W의 경우 앞선 논문에서 제시한 것과 같이 오픈 소스 ECDIS인 OpenCPN 을 수정하여 기능을 확장하였다. 주요 기능으로는 출발, 경유 및 도착 지점의 좌표 값 입력과 도착 예정 시간을 입력할 수 있다.

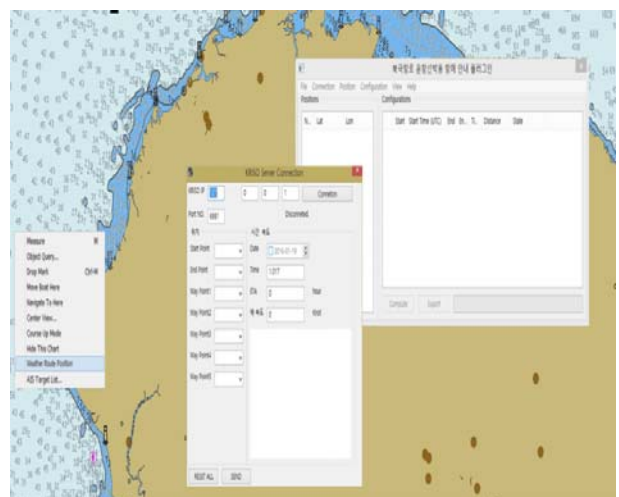


Fig. 3 Input GUI based on OpenCPN

입력 받은 값은 UDP 프로토콜 기반으로 위성통신을 통해서 항해 지원 서버(KRISO)로 전달된다. 지형 및 해역의

정보를 표시하기위한 북극지역의 해도로는 OpenCPN의 플러그인으로 제공하는 chart download plugin를 활용하여 해당 지역의 전자해도 정보를 다운로드하였다.

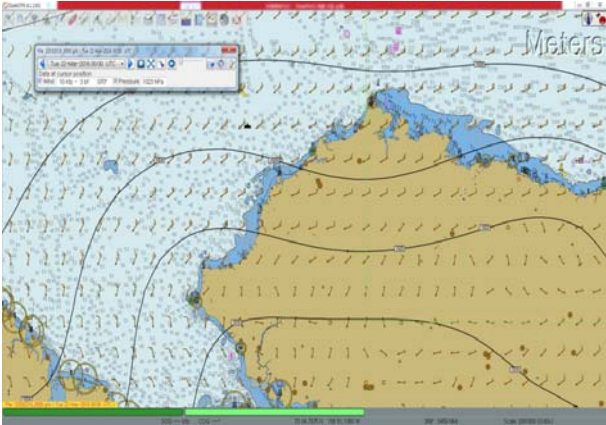


Fig. 4 ENC Chart near North Pole

Onboard S/W는 최적 항로 계산을 위해 항해 지원 서버에 정보를 전달한 후, 그 결과값으로 전달받게 된다. 결과값에는 최적 항로에 대한 경로 정보 값과 기상(빙상)정보가 포함된 GRIB 파일을 포함한다. 기상 정보는 이전 논문에서 제시된 것과 같이 Color Code 인코딩 되어 Onboard S/W인 OpenCPN에 표시된다. Fig 5는 테스트 값을 적용하여 OpencPN에 최적 항로 정보와 기상(빙상)정보를 Color Code기반으로 표시한 예시이다.

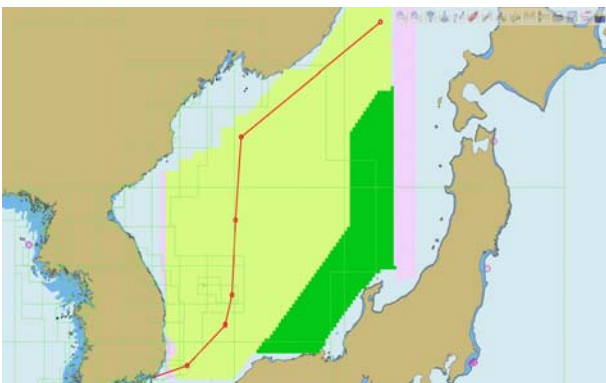


Fig. 5 Route and Ice information display on the OpenCPN

## 2.2 북극항로 운항선박용 항해지원 서버 (KRISO)

북극항로 운항선박용 항해지원 서버는 다음과 같이 동작한다.

### (1) Transit model load

Transit model은 북극항로 항해 선박의 특성을 사전에 분석하여 저장한다. 북극 항해 선박의 최적 경로와 최적 출력 값을 결정하기 위해 필요한 데이터를 추출하며, 해당 데이터와 빙상 정보, 도착 예정 시간 등을 고려하여 최적 항로를 계산한다(KAIST). Transit model 기반으로 최적 출력을 구하기 위한 공식은 아래와 같다.<sup>2</sup>

$$P_B = \left[ \left\{ (C_{FS} + \Delta C_F) \frac{S + S_{BK}}{S} + C_R \right\} \frac{1}{2} \rho S V_S^2 + \Delta_{MW} + \Delta_{MS} + \Delta_{DW} + \Delta_{DS} + \Delta_A + \Delta_D + \Delta_H + \Delta_I \right] \frac{V_S}{\eta_{DS} \eta_T}$$

where

$P_B$  =break power (W)

$C_{FS}$  =frictional resistance coefficient

$\Delta C_F$  =model-ship correlation

$C_R$  =residual resistance coefficient

$S$  =wetted surface area ( $m^2$ )

$S_{BK}$  =bilge keel area ( $m^2$ )

$V_S$  =ship speed (m/sec)

$\Delta_{MW}$  =motion wave resistance (N)

$\Delta_{MS}$  =motion swelle resistance (N)

$\Delta_{DW}$  =diffraction wave resistance (N)

$\Delta_{DS}$  =diffraction swelle resistance (N)

$\Delta_A$  =air resistance (N)

$\Delta_D$  =drifting resistance (N)

$\Delta_H$  =hull roughness resistance (N)

$\Delta_I$  =ice resistance (N)

$\eta_{DS}$  =quasi-propulsive coefficient

$\eta_T$  =transmission efficiency

Transit model 데이터를 저장하는 자료 구조는 Fig. 6과 같다.

```

float TEST_S,SRK,RMS,LNL;
//MODEL TEST RESULTS
float DELTA;
float VS[table(12),CM[table(12),TM[table(12),M[table(12),N[table(12),ETA[table(12),ETAS[table(12) ;
// WAVE AND SWELL DATA
float #WV[table(13),SW[table(13);
float DELMOT[table(16)[13][19]; // 0-180(19) DEG 0-30.48(13) M 2-17(16) KNOTS
float DELMOT[table(16)[13][19]; //MOTION SWELLE [DELMOTS] 0-180(19) DEG 0-30.48(13) M 2-17(16) KNOTS
float DELDF[table(16)[13][19]; //DIFFRACTION WAVE [DELDFW] 0-180(19) DEG 0-30.48(13) M 2-17(16) KNOTS
float DELDF[table(16)[13][19]; //DIFFRACTION SWELLE [DELDFSW] 0-180(19) DEG 0-30.48(13) M 2-17(16) KNOTS
float DELAIR[table(11)[19]; //AIR [DELAIR] 0-180(19) DEG 0-30(11) M/S
float DELDRIF[table(19)[19]; //DRIFTING [DELDRIF] 0-180(19) DEG 0-18(19) KNOTS
float DELHULL[table(19)[21]; //HULL ROUGH [DELHULL] 0-0.002(21) M 0-18(19) KNOTS

double getLL(double a,float b);
double get2ELL(double a, double p, float A,float B,float C,float D);
double get3ELL(double a, double p, double s, float A,float B,float C,float D,float E,float F,float G, float H);
public:
void readConfig(string filename);
void dumpConfig();

int getVS(float shipSpeed,float interpolation);
int getWaveHeight(float waveHeight);
int getSwellHeight(float swellHeight);

double getCR(float shipSpeed);
double getTW(float shipSpeed);
double getETA(float shipSpeed);
double getETA(float shipSpeed);
double getDELMOT(float shipSpeed, float shipAngle, float waveHeight);
double getDELMOTS(float shipSpeed, float shipAngle, float swellHeight);
double getDELDF(float shipSpeed, float shipAngle, float waveHeight);
double getDELDFSW(float shipSpeed, float shipAngle, float swellHeight);
double getDELAIR(float shipSpeed, float airAngle);
double getDELDRIF(float shipSpeed,float driftAngle);
double getDELHULL(float shipSpeed,float hullRoughness);

};
double getPower(float shipSpeed,float shipAngle, float waveHeight, float swellHeight, float airSpeed, float airAngle, float
    
```

Fig. 6 Data structure for Transit model

(2) X, Y 좌표 변환

Onboard S/W로부터 입력 받은 위도/경도 좌표 값을 아래의 공식으로 X, Y 좌표 값으로 변환한다.

$$X = FE + \rho \sin(\lambda - \lambda_0)$$

$$Y = FN - \rho \cos(\lambda - \lambda_0)$$

$$\rho = \frac{2ak_0t}{\sqrt{(1+e)^{(1+e)}(1-e)^{(1-e)}}$$

$$t = \tan\left(\frac{\pi}{4} - \frac{\varphi}{2}\right) \left(\frac{1+e \sin \varphi}{1-e \sin \varphi}\right)^{e/2}$$

where

- X = Horizontal axis
- Y = Vertical axis
- $\varphi$  = Latitude
- $\lambda$  = Longitude
- $\lambda_0$  = Longitude of origin
- $k_0$  = Scale factor at natural origin (North Pole)
- FE = False Easting
- FN = False Northing

(3) 기상(빙상)정보 요청 및 획득

기상(빙상)정보를 획득하기 위해 항해지원 서버는 기상정보 서버(KIOST)에 해당 지역의 기상 정보를 요청한다. 요청 시 위·경도 좌표와 도착시간을 고려한 시간정보를 기상정보 서버에 전송하게 된다. KIOST의 기상정보 서버는 해당 요청 정보를 바탕으로 기상(빙상) 정보를 NETCDF형태로 전송한다.

(4) GRIB 파일 생성

기상정보 서버로부터 전송받은 파일을 Onboard S/W(Araon)에 전송하기 위해 해당 파일을 분석하여 GRIB 파일로 변환한다. GRIB 파일에는 기상정보뿐만 아니라 빙상정보를 포함하여, Onboard S/W에 해당 정보를 바탕으로 Color code로 변환하여 빙상에 대한 정보를 Fig. 7과 같이 표현할 수 있다.

GRIB1	GRIB2	GRIB1 Data
SECTION 0 Indicator section	SECTION 0 Indicator Section	0 Latitudo/Longitude Grid also called Equidistant Cylindrical or Plate Carree projection
SECTION 1 Product definition section	SECTION 1 Identification Section	1 Mercator Projection
SECTION 2 Grid description section	SECTION 2 Local Use Section	2 Geostrophic Projection
SECTION 3 Bit-map section	SECTION 3 Grid Definition Section	3 Lambert Conformal, secant or tangent, conical or bipolar projection
SECTION 4 Binary data section	SECTION 4 Product Definition Section	4 Gaussian Latitude/Longitude Grid
SECTION 5 7777 (end section)	SECTION 5 Data Representation Section	5 Polar Stereographic projection
	SECTION 6 Bit-map Section	6 Universal Transverse Mercator (UTM) projection
	SECTION 7 Binary Data Section	7 Simple polyconic projection
	SECTION 8 End Section	8 Albers equal area, secant or tangent, conic or bipolar projection
		9 Mercator projection
		10 Rotated latitude/longitude grid
		11-12 Reserved
		13 Oblique Lambert conformal, secant or tangent, conical or bipolar projection
		14 Rotated Gaussian latitude/longitude grid
		15-19 Reserved
		20 Stretched latitude/longitude grid
		21-23 Reserved
		24 Stretched Gaussian latitude/longitude grid
		25-29 Reserved
		30 Stretched and rotated latitude/longitude grid
		31-33 Reserved
		34 Stretched and rotated Gaussian latitude/longitude grid
		35-40 Reserved
		40 Spherical harmonic coefficients
		41-43 Reserved
		43 Rotated spherical harmonic coefficients
		44-49 Reserved
		49 Stretched spherical harmonics
		71-79 Reserved
		80-81 Stretched and rotated spherical harmonic coefficients
		86 Space view perspective or orthographic
		91-19 Reserved
		192-194 Reserved for local use

Fig. 7 GRIB file format

(5) 최적 경로 계산 (KAIST)

최적 경로의 계산은 KAIST에서 개발된 최적 경로 계산 모듈을 사용하며, 그 인터페이스는 Fig. 8과 같다.

```
output KAIST(gamemap_point +init_info, AADS.Client + client); // 카이스트 모듈 진입 함수
// gamemap_point는 시작점과 도착점 그리고 waypoint 좌표
// AADS.Client + clients는 AADS.Client객체 포인터를 카이스트 모듈에 전달
// AADS.Client 클래스의 함수를 사용
// AADS.Client 클래스에는 KIOST로부터 기상데이터를 가져올수 있는 함수
```

Fig. 8 Interface between navigational support server and optimal routing calculation module

최적 경로 계산의 경우 주어진 파라미터를 활용하여 가장 최적의 해가 나올 때까지 반복계산하며, 최종 계산된 경로 값과 출력 값을 전달한다.

2.3 기상정보 서버(KIOST)

기상정보 서버는 KIOST에서 운영하고 있으며, 주기적인 시간 단위로 기상(빙상)정보를 수집하여 NETCDF형태로 저장한다. Fig. 9는 NETCDF로 저장된 기상(빙상)정보의 저장구조이다.

이름	수정된 날짜	유형	크기
AL2	2015-12-01 오후...	파일 폴더	
GFS	2015-12-01 오후...	파일 폴더	
TPZ	2015-12-01 오후...	파일 폴더	
WW3	2015-12-01 오후...	파일 폴더	
TPZ-ARC-ICF-2015-09-20-12-P000.nc	2015-09-21 오후...	ALZip GZ File	2,849KB
TPZ-ARC-ICF-2015-09-20-12-P024.nc	2015-09-21 오후...	ALZip GZ File	2,987KB
TPZ-ARC-ICF-2015-09-20-12-P048.nc	2015-09-21 오후...	ALZip GZ File	3,013KB
TPZ-ARC-ICF-2015-09-20-12-P072.nc	2015-09-21 오후...	ALZip GZ File	3,019KB
AL2-ARC-ICF-2015-09-20-12-P000.nc	2015-09-23 오전...	NC 파일	16,401KB

Fig. 9 Structure of saved NETCDF files

3. 결론

본 논문에서는 이전 개념 설계 논문을 바탕으로 하여 실제 구현하여 테스트하였다. 기존 저장된 데이터를 바탕으로 지정된 환경에서 테스트를 수행하였으며 추후, 실선에 탑재하여 기능 및 성능에 대한 검증을 수행할 예정이다.

## 참 고 문 헌

- 1) 이한진, 북극항로 운항선박용 항해안전 시스템 개념 설계, 2015. 11, 대한조선학회 추계학술대회
- 2) Han Jin Lee, Eun-Chan Kim, Kuk-Jin Kang, 2016, Prototype Development of Safe Voyage Planning System for Vessel operation in Arctic Sea Route, The 31<sup>st</sup> International Symposium on Okhotsk Sea & Sea Ice



이 한 진

- 1969년생
- 1997년 서울대학교 조선해양공학과 박사
- 현 재 : 선박해양플랜트연구소 책임연구원
- 관심분야 : 선박조종, e-Navigation
- 연 락 처 : 042-866-3607
- E - mail : hjlee@kriso.re.kr



정 성 현

- 1975년생
- 2006년 충남대학교 컴퓨터 시스템 멀티미디어 박사 수료
- 현 재 : 동강엠텍(주) 기업부설연구소장
- 관심분야 : 항해지원, 시뮬레이터
- 연 락 처 : \*\*\*-\*\*\*\*-\*\*\*\*
- E - mail : logixer@dkmtech.com



강 국 진

- 1954년생
- 1991년 인하대학교 조선공학과 박사
- 현 재 : 선박해양플랜트연구소 책임연구원
- 관심분야 : 선박공학, 방해공학
- 연 락 처 : 042-866-3435
- E - mail : kjkang@kriso.re.kr



최 현

- 1981년생
- 2007년 한밭대학교 컴퓨터공학과 학사
- 현 재 : 동강엠텍(주) 전임연구원
- 관심분야 : 항해지원 프로그램 개발
- 연 락 처 : \*\*\*-\*\*\*\*-\*\*\*\*
- E-mail : hyun3388@dkmtech.com



정 성 엽

- 1981년생
- 2016년 한국해양대학교 해양공학과 박사
- 현 재 : 선박해양플랜트연구소 선임연구원
- 관심분야 : 빙해선박, 빙저항
- 연 락 처 : 042-866-3432
- E - mail : syeop@kriso.re.kr



최 성 우

- 1987년생
- 2015년 충남대학교 컴퓨터공학과 석사
- 현 재 : 동강엠텍(주) 연구원
- 관심분야 : 항해지원 프로그램 개발
- 연 락 처 : \*\*\*-\*\*\*\*-\*\*\*\*
- E - mail : choisw@dkmtech.com