Journal of the Korean Society of Marine Environment & Safety
Research Paper
Vol. 29, No. 6, pp. 681-687, October 31, 2023, ISSN 1229-3431(Print) / ISSN 2287-3341(Online) https://doi.org/10.7837/kosomes.2023.29.6.681

인공지능 기법에 의한 최적 운항자세 선정에 관한 연구

박동우*†
* 동명대학교 해양모빌리티학과 교수

Study on the Selection of Optimal Operation Position Using AI Techniques

Dong-Woo Park*†
* Professor, Department of Marine Mobility, Tongmyong University, Busan 48520, Korea

요 약 : 최적 운항자세 선정 기술이란 주어진 운항 배수량과 운항 선속에서 최소의 저항을 가지는 즉, 최적의 연료 소비 효율을 가지는 초기 선수흘수와 선미흘수를 제시하는 것이다. 본 논문의 주 목적은 대상선박의 유효동력 데이터를 기반으로 주어진 운항조건에서 최대의 에너지효율을 가지는 최적의 운항자세를 선정하는 프로그램 개발하는 것이다. 본 프로그램은 인공지능 기법에 의한 파이썬 기반 GUI(Graphical User Interface)로 작성되어 선주가 쉽게 사용할 수 있도록 하였다. 그 과정에 있어 대상 선박 소개, 전산유체역학(CFD)을 통한 유효동력 데이터 수집, 심층학습을 사용한 유효동력 모델 학습 방법 그리고 심층신경망(DNN) 모델을 응용한 최적 운항자세 제시 프로그램을 구체적으로 설명하였다. 선박은 운항 별로 화물을 싣고 내리게 되고, 이에 화물 적재량이 변화되고 배수량이 변경된다. 선주는 배수량 별 예상 선속에 따라 최소저항을 가지는 즉, 최대의 에너지효율을 가지는 최적의 운항자세를 알고자 한다. 개발된 GUI는 해당선박의 태블릿 PC와 앱에 설치하여 최적 운항자세 선정에 활용 가능하다.

핵심용어 : 최적 운항자세, 인공지능, 파이션 기반 GUI, 심층학습, 심층신경망

Abstract : *The selection technique for optimal operation position selection technique is used to present the initial bow and stern draft with minimum resistance, for achievingthat is, the optimal fuel consumption efficiency at a given operating displacement and speed. The main purpose of this studypaper is to develop a program to select the optimal operating position with maximum energy efficiency under given operating conditions based on the effective power data of the target ship. This program was written as a Python-based GUI (Graphic User Interface) usingbased on artificial intelligence techniques sucho that ship owners could easily use the GUIit. In the process, tThe introduction of the target ship, the collection of effective power data through computational fluid dynamics (CFD), the learning method of the effective power model using deep learning, and the program for presenting the optimal operation position using the deep neural network (DNN) model were specifically explained. Ships are loaded and unloaded for each operation, which changes the cargo load and changes the displacement. The shipowners wants to know the optimal operating position with minimum resistance, that is, maximum energy efficiency, according to the given speed of each displacement. The developed GUI can be installed on the ship's tablet PC and application and used to determineselect the optimal operating position.*

Key Words : *Optimal operation position, Artificial intelligence, Python-based GUI, Deep learning, Deep neural network (DNN)*

1. 서 론

최적 운항자세 선정 기술이란 주어진 운항 배수량과 운항 선속(운항 배수량과 운항 선속을 본 논문에서는 운항조건이라고 정의함)에서 최소의 저항을 가지는 즉, 최적의 연료 소비 효율을 가지는 초기 선수흘수와 선미흘수를 제시하는 것이다. 본 논문의 저자는 2017년 연구과제인 연안해운 온실가스 감축기술 경제성 평가를 위한 기획연구(해양수산부, 선박안전기술공단)에서 동일한 배수량에서 초기 선수와 선미 흘수 조건에 따라 선박의 저항을 나타내는 정량적 값인 유효마력 관점에서 약 2.5 % 차이가 나는 것을 보여주었다. 유효마력의 차이는 자세에 따라 선박이 유체로부터 받는 압력이 달라짐으로 인해 파형의 변화가 있고, 이로 인해 조파저항에서 차이를 나타내는 것이다.

선박의 배수량 별 최적 운항자세의 필요성과 효과를 보면 다음과 같다. 배수량 70 ~ 80 %에서는 구상선수 성능 미 발휘에 따른 상당한 연료 요구되고, 최적 운항자세 선정에 의해

† dwpark@tu.ac.kr, 051-629-1654

기대 이상의 연료 절감 효과가 가능하다. 배수량 80～90 %에서는 수면 위로 노출된 구상선수로 인해 과도한 저항 증가가 발생되기 때문에 최적 운항자세 선정에 의해 대폭적인 연료 절감 효과가 가능하다. 그리고, 배수량 90～100 %에서는 운항속도에 따라 최적화된 자세선정이 필요하다.

최적 운항자세와 관련된 연구결과는 다음과 같다.

Lee et al.(2021)은 11,000 TEU 컨테이너선의 소요마력 저감을 위한 트림변화에 관한 연구 결과를 소개하였다. 저항성능의 경우, 전반적으로 선수트림이 증가할수록 유효동력이 작아지는 경향을 설명하였다. 또한, 트림변화에 따른 선수벌브 상단의 돌출 정도가 선수파형 분포의 변화를 가져오고, 결과적으로 유효동력에 영향을 준다는 결과를 언급하였다. Han et al.(2015)은 트림변화에 따른 컨테이너선의 저항성분별 특성 연구를 소개하였다. 대상선박은 6,800 TEU 컨테이너선으로 저속과 고속의 동적운동 특성 차이로 트림, 침하 변화량을 검토하기 위하여 자유상태(Free)와 고정상태(Fixed)에서 수치계산을 수행하였고, 전 저항의 차이를 면밀하게 확인하였다. 또한, 고속 조건에서 선미트림이 선수트림과 EVEN 상태 저항이 더 크게 발생하는 것을 확인하였다. Seo et al.(2015)은 CFD를 이용한 컨테이너 선형의 트림별 저항성능 해석 결과에서 총 저항에서 저항성분을 분석을 하였다. Part et al.(2013)은 선박의 트림 자세가 저항 성능에 미치는 영향에 대하여 조사한 결과를 발표하였다. 본 논문에서는 배수량-트림자세 별 저항성능을 분석하기 위해서 수치해석을 수행하였고, 해석과정에서 선체를 미소영역으로 나누어 미소영역 별로 마찰저항과 압력저항을 분석하였다. 최적 운항자세와 관련된 연구결과에 따르면 2050년 온실가스 목표량 달성에 있어 운항적 조치는 탄소배출량 감축에 있어 중요한 역할을 하는 것으로 파악된다.

본 논문의 주 목적은 선박의 유효동력 데이터를 기반으로 주어진 운항조건에서 최대의 에너지효율을 가지는 최적의 운항자세를 선정하는 프로그램을 개발하는 것이다. 선박은 운항 별로 화물을 싣고 내리게 되고, 이에 화물 적재량이 변화되고 배수량이 변경된다. 선주는 배수량 별 예상 선속에 따라 최소저항을 가지는 즉, 최대의 에너지효율을 가지는 최적의 운항자세를 알고자 한다. 본 프로그램은 인공지능 기법에 의한 파이썬 기반 GUI(Graphical User Interface)로 작성되어 선주가 쉽게 사용할 수 있도록 하였다. 그 과정에 있어 대상 선박 소개, 전산유체역학(CFD)을 통한 유효동력 데이터 수집, 심층학습을 사용한 유효동력 모델 학습 방법 그리고 심층신경망(DNN) 모델을 응용한 최적 운항자세 제시 프로그램을 구체적으로 설명하였다.

2. 대상 선박 및 데이터 생성

2.1 대상 선박 및 데이터

대상 선박은 Fig. 1의 엠에스페리의 뉴스타호로 전장 163.57 m, 폭 25.60 m, 깊이 13.21 m 그리고 설계흘수는 4.5 m 이다.

Fig. 1. Target ship.

데이터는 Table 1에 보여준 대로 운항 가능한 4가지 흘수인 3.5 m, 4.0 m, 4.5 m 그리고 5.0 m에서 트림조건은 흘수 별로 EVEN을 포함하여 7가지에서 그리고 선속은 15노트에서 22노트까지 1노트 간격으로 8가지로 조합된 총 224가지를 계산하여 수집하였다. 즉, Table 1의 56가지 흘수에 대해서, 각 흘수 별로 8가지 선속에 대한 계산을 수행하였다. Table 1에서 트림(Trim)은 선수흘수와 선미흘수의 차이로, 선수흘수-선미흘수로 정의하였고, 음(-)의 부호를 가지 트림은 선미트림(Trim by Stern)을 나타내고 있다.

Table 1. Calculation condition

Condition	3.5m draft		4.0m draft		4.5m draft		5.0m draft	
	F.P.	A.P.	F.P.	A.P.	F.P.	A.P.	F.P.	A.P.
EVEN	3.50	3.50	4.00	4.00	4.50	4.50	5.00	5.00
-0.5m Trim by Stern	3.25	3.75	3.75	4.25	4.25	4.75	4.75	5.25
+0.5m Trim by Bow	3.75	3.25	4.25	3.75	4.75	4.25	5.25	4.75
-1.0m Trim by Stern	3.00	4.00	3.50	4.50	4.00	5.00	4.50	5.50
+1.0m Trim by Bow	4.00	3.00	4.50	3.50	5.00	4.00	5.50	4.50
-1.5m Trim by Stern	2.75	4.25	3.25	4.75	3.75	5.25	4.25	5.75
+1.5m Trim by Bow	4.25	2.75	4.75	3.25	5.25	3.75	5.75	4.25

2.2 데이터 생성을 위한 수치해석 방법[1)]

심층학습에 사용된 기본 데이터는 2.1절에서 언급한 224가지이다. 이 224가지 데이터 생성 방법은 CFD 해석을 통해 총 저항을 계산하고 유효동력을 추정하였다. 본 연구에서는 모형 크기에서의 총 저항(R_{TM})을 계산하기 위해 3차원 비정상, 비압축성 난류 유동을 가정하였다. 따라서 이에 대응하는 지배방정식인 연속방정식과 RANS(Reynolds-averaged Navier-Stokes) 방정식은 식(1) 및 식(2)와 같다.

$$\frac{\partial}{\partial x_i}(\rho \bar{u}_i) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho \bar{u}_i) + \rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \rho\overline{u'_i u'_j}\right] + \rho \bar{g}_i \tag{2}$$

여기서 $\bar{}$는 시간 평균된 값을 의미한다. $\bar{u}_i$와 $\bar{u}_j$ $(i,j=1,2,3)$는 x_i와 $x_j (i,j=1,2,3)$ 방향에 대한 평균 유속이며, ρ는 유체의 밀도, $\bar{p}$는 평균 압력, μ는 유체의 점성 계수, $\bar{g}_i$는 중력가속도이다. 또한, 식(2)의 레이놀즈 응력(Reynolds stress) 항인 $\rho\overline{u'_i u'_j}$ 은 Reynolds Stress 난류 모델을 적용하였다. 자유수면을 고려하기 위해 VOF(volume of fluid)를 사용하여 다상 유동을 수치 해석하였다. 유동장 계산은 동적 트림을 고려하기 위해 DFBI(dynamic fluid body interaction) 기법을 적용하였다. DFBI 기법은 선체의 자세 변화에 따라 계산 영역 전체가 이동 및 회전하는 방법으로 선체의 싱키지(sinkage)는 계산 영역 전체의 상 방향 수직 이동으로 나타나며, 트림(trim)은 계산 영역 전체의 폭 방향 회전으로 나타난다. 계산 시간 간격은 △t=0.02s를 적용하여 총 60초까지 계산을 수행하였다. 매 시간 간격에서 5회의 내부 반복 계산을 수행하였으며, 초기 조건에 따른 자세 오차를 배제하기 위해 계산 1초 후부터 선체에 작용하는 힘과 모멘트를 통해 선체의 동적 자세 변화를 고려하였다. 선체의 동적 자세를 고려함에 따라 저항값이 주기적인 변화를 보이는데, 45초 이후부터 60초까지의 평균값을 계산하여 저항 값으로 사용하였다. 평균값을 계산하는 시간에 따른 오차는 약 0.3 % 미만이다. 난류 모델은 Reynolds stress model을 적용하였고 입구 경계 조건은 고정 속도 조건 및 자유 수면 높이 고정 조건을 사용하였다. 대상 선박의 자세 변화는 선체 표면의 유체 압력 변화로 인한 종경사(trim) 및 침하량(sinkage)을 고려하였으며, 이를 제외한 나머지 운동은 구속한 채 해석하였다.

점성 유동해석 결과로부터 선체의 수직방향 응력과 수평방향 응력으로부터 선체에 작용하는 모형 크기에서의 전 저항(R_{TM})을 식(3)으로 부터 계산하였다.

$$C_{TM} = \frac{1}{S}\iint(-C_P n_x + C_{fx})dS \tag{3}$$

여기서 S는 선체의 접수면적, C_P는 선체에 수직으로 작용하고 있는 수직응력의 압력계수, n_X는 선박의 축 방향 단위벡터, C_{FX}는 선체에 수평으로 작용하는 수평응력 계수이고, C_{TM}은 선체에 작용하는 전 저항 계수이다. CFD로부터 계산된 식(3)의 C_{TM}을 이용하여 ITTC-1957 해석법에 따라 유효동력을 추정하였다. 실선 확장 과정에 있어 2차원 해석법을 사용하였다. 모형선-실선 상관계수는 유사 선종과 유사 선박 크기인 5척의 수조모형시험와 CFD 결과와의 차이를 활용하였다. CFD 결과는 모형선 크기와 실선의 계산 결과를 사용하였다.

2.3 심층학습을 통한 유효동력 모델 학습

CFD를 통한 대상 선박의 유효동력 데이터는 224가지의 운항조건에 대해서 계산했기 때문에 연속적인 값을 가지는 실제 운항조건에 적용하기 위해서는 내삽 등의 방법을 활용해야만 한다. 본 눈문에서는 이런 문제를 해결하기 위해 Fig. 2와 같이 심층신경망으로 유효동력 모델을 구성하였다.

모델의 입력은 속도, 배수량, 자세변화 조건으로 길이가 3인 실수 벡터이며, 출력은 유효동력으로 실수 스칼라다. 모든 입출력의 각 요소 x_i는 식(4)와 같이 표준정규분포 형태로 정규화하며, $\mathbb{E}_B[x_i]$와 $\mathrm{Var}_B[x_i]$는 학습 데이터의 배치에 대한 평균과 분산을 의미한다.

$$\mathrm{Standardize}(x_i) = \frac{x_i - \mathbb{E}_B[x_i]}{\sqrt{\mathrm{Var}_B[x_i]}} \tag{4}$$

정규화된 입력은 먼저 식(5)처럼 아핀(affine) 변환을 통해 C 길이의 특징 벡터 표현으로 변환하며, 신경망의 첫 번째 Affine_{in} 층의 매개변수 $W_{in} \in \mathbb{R}^{3\times C}$는 선형 변환 행렬을, $b_{in} \in \mathbb{R}^{C}$는 bias 벡터를 의미한다.

$$\mathrm{Affine}_{in}(x) = x W_{in} + b_{in} \tag{5}$$

1) 수치해석 방법은 본 저자가 선박의 저항성능 해석에 사용하는 표준화된 방법으로 Park et al.(2013)에도 유사하게 기술되어 있음. 본 논문에서는 대상선박이 다름에 따라 수치계산 영역 등의 구체적인 값에는 차이가 있음.

특징 벡터는 L개의 잔차(residual) 블록 층을 통해 처리되며, 각 잔차 블록은 레이어 정규화(layer normalization, LN) 층과 다층 퍼셉트론(multi layer perceptron, MLP) 층으로 구성된다. LN 층은 식(6)과 같이, 입력 벡터를 채널에 대한 평균 $\mathbb{E}_C[x]$과 분산 $\mathrm{Var}_C[x]$을 사용해 정규화한 다음에 매개변수 $\gamma \in \mathbb{R}^C$와 $\beta \in \mathbb{R}^C$로 크기와 위치를 조절하여 심층학습 과정을 안정화한다(Ba et al., 2016). MLP 층은 두 개의 Affine 층과 그 사이에 하나의 비선형 층으로 구성되며, 본 연구에서는 비선형 층의 함수로 렐루 함수(rectified linear unit, ReLU)를 사용하였다. MLP 층은 식(7)과 같이 입력 특징 벡터의 채널 차원을 더 높은 D차원으로 확장 변환하는 첫 번째 Affine 층의 $W_1 \in \mathbb{R}^{C\times D}$과 $b_1 \in \mathbb{R}^D$, 비선형 층을 통과한 다음 다시 원래 채널 차원인 C로 축소 변환하는 두 번째 Affine 층의 $W_2 \in \mathbb{R}^{D\times C}$와 $b_2 \in \mathbb{R}^C$을 매개변수로 가진다.

$$\mathrm{LN}(x) = \gamma \frac{x - \mathbb{E}_C[x]}{\sqrt{\mathrm{Var}_C[x] + \epsilon}} + \beta \tag{6}$$

$$\mathrm{MLP}(x) = \max(0, x W_1 + b_1) W_2 + b_2 \tag{7}$$

각 잔차 블록은 현재 블록의 입력 벡터 x_l가 식(8)처럼 LN 층과 MLP 층을 차례로 거친 다음 원래 x_l와 더하는 잔차 연결 방식으로 갱신된 특징 벡터 x_{l+1}을 구한다. 잔차 연결은 여러 층을 깊게 쌓는 방식의 심층신경망 학습에서 그래디언트(gradient)가 소실되는 문제를 완화할 수 있다.

$$x_{l+1} \leftarrow x_l + \mathrm{MLP}(\mathrm{LN}(x_l)) \tag{8}$$

모든 잔차 블록으로 처리된 특징 벡터는 식(9)와 같이 LN 층을 거친 다음 마지막 Affine 층으로 예측값 $\hat{y}$로 변환되며, 마지막 $\mathrm{Affine}_{\mathrm{out}}$ 층은 $W_{out} \in \mathbb{R}^{C\times 1}$와 $b_{out} \in \mathbb{R}^1$를 매개변수로 사용한다.

$$\hat{y} \leftarrow \mathrm{Affine}_{\mathrm{out}}(\mathrm{LN}(x)) \tag{9}$$

심층신경망 모델의 예측값 $\hat{y}$와 목표값 y 사이의 거리를 심층학습의 손실 함수로 사용했으며, 거리를 계산하기 위해 식(10)과 같이 Smooth L1 함수를 사용하였다. Smooth L1 함수는 절대값 오차가 1.0보다 작다면 제곱값 오차를, 아니면 절대값 오차를 사용하는 방식으로, 제곱값 오차 방식과 비교해 이상값에 덜 민감하여 그래디언트가 폭발하는 문제를 줄일 수 있다.

$$\mathrm{Loss}(\hat{y}, y) = \begin{cases} 0.5(\hat{y} - y)^2, & \text{if } |\hat{y} - y| < 1.0 \\ |\hat{y} - y| - 0.5, & \text{otherwise} \end{cases} \tag{10}$$

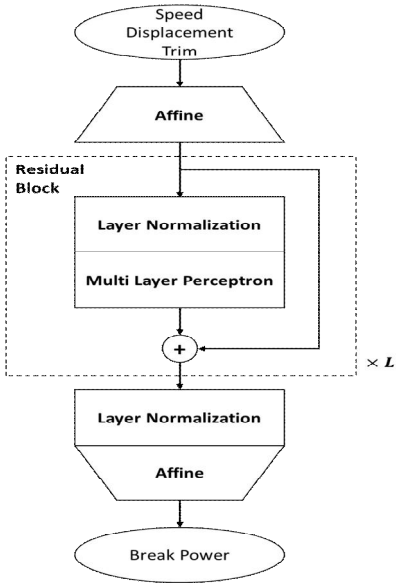

Fig. 2. DNN architecture.

심층신경망 학습에 사용한 초매개변수는 Table 2와 같으며 모델의 매개변수를 최적화시키는 방법은 모멘텀 기반의 경사하강법인 Adaptive Moment Estimation(Adam)을 사용하였다. 매개변수 갱신을 위한 학습률은 0.001로 고정하였고, 학습과정에서 과도한 발산을 막기 위해 그래디언트의 최대치가 그래디언트 노름(norm)의 1.0배가 되도록 설정하였다. 총 1,000번의 에포크(epoch) 동안 학습을 진행하였고 배치(batch)는 데이터 전체인 224개를 사용하였다. 첫 번째 아핀 층의 출력부터 길이가 64인 특징 벡터를 사용하였고, 총 4개의 잔차 블록을 사용하였다. 각 다층 퍼셉트론(MLP)의 은닉층에서는 특징 벡터의 길이를 256로 확장 변환하였으며 활성 함수는 렐루 함수(ReLU)를 사용하였다.

Table 2. Hyperparameters of DNN learning

Optimizer	Adam
Learning rate	0.001
Gradient clipping by norm	1.0
Number of epochs	1,000
Batch size	224
Number of feature channels: C	64
Number of residual blocks: L	4
Number of hidden channels: D	256
Activation function	ReLU

3. 결과 및 토론

3.1 유효동력 DNN 모델을 이용한 최적의 운항자세 제안

심층신경망은 학습 데이터에 적합한 임의의 함수를 근사할 수 있다. 이렇게 DNN 모델로 근사화한 함수는 학습에 사용하지 않은 일반적인 입력에 대한 추론도 가능하다.

DNN을 사용해 Fig. 3의 총 224개의 유효동력 데이터를 학습한 모델은 Fig. 4처럼 임의로 설정한 더 밀집된 유효동력을 내삽으로 예측할 수 있다.

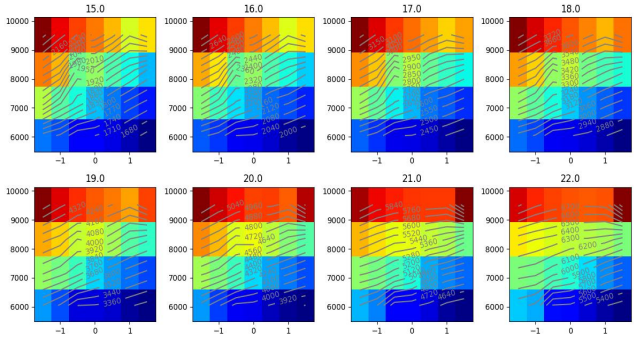

Fig. 3. Graph of calculated effective power using CFD (x: trim, y: displacement).

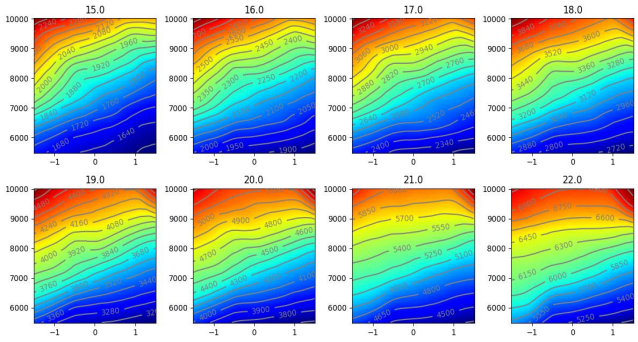

Fig. 4. Graph of predicted effective power using DNN (x: trim, y: displacement).

Table 3의 코드는 Fig. 4를 시각화하기 위해 NumPy(Harris et al., 2020)와 Matplotlib(Hunter 2007) 라이브러리를 사용한 파이썬 코드다. 배수량과 트림조건을 각각 101개로 늘려서 더 밀집되도록 만들고 DNN을 통해 내삽된 제동동력을 추론한 다음 수치별 색상과 등치선 그래프로 결과를 보여주었다.

Table 4의 코드는 Fig. 5의 주어진 속도와 배수량 조건에서 최적 운항자세를 찾기 위해 SciPy(Virtanen et al. 2020) 라이브러리를 사용한 파이썬 코드다. 해당 코드는 주어진 속도(v_query)와 주어진 배수량(d_query) 조건에서 -1.5와 +1.5 사이의 자세변화 범위 중 유효동력을 최소화하는 운항자세를 찾는 코드다. 에너지 효율이 가장 좋지 않은 운항자세의 경우, Table 4 코드에서 함수 결과의 부호를 바꾸면 유효동력이 가장 큰 운항자세 조건을 찾을 수 있다. Fig. 5는 주어진 속도 21.5 노트, 주어진 배수량 9360 m^3인 조건에서 최소저항을 가지는 최적의 운항자세 결과를 보여준다.

Table 3. Python code for visualization of interpolated data

Python code for visualization of interpolated data

```
import numpy as np
import matplotlib.pyplot as plt

N = 101
vs = np.linspace(15, 22, 8)          # velocity
ds = np.linspace(5500, 10000, N)  # displacement
ts = np.linspace(-1.5, 1.5, N)      # trim
vdt = np.meshgrid(vs, ds, ts, indexing='ij')
vdt = np.stack(vdt, axis=-1).reshape(-1, 3)
p = DNN(vdt).reshape(8, N, N)        # power

for i in range(len(vs)):
    plt.title(vs[i])
    plt.pcolormesh(ts, ds, p[i], cmap='jet')
    c = plt.contour(ts, ds, p[i], colors='gray')
    plt.clabel(c)
    plt.show()
```

Table 4. Python code for trim optimization to minimize effective power

Python code for trim optimization to minimize effective power

```
import numpy as np
from scipy.optimize import minimize_scalar
from functools import partial

def function(v, t, d):
    return DNN(np.array([[v, d, t]])).squeeze()

# v_query & d_query are user-defined
f = partial(function, v=v_query, d=d_query)
result = minimize_scalar(f, bounds=[-1.5, 1.5])
best_trim = result['x']
best_power = result['fun']
```

3.2 응용 프로그램

개발된 프로그램을 바탕으로 Fig. 6과 같이 사용자가 운항조건인 속도와 배수량을 입력하면, 최적의 운항자세를 찾아내는 응용 프로그램을 구현하였다. 운항조건인 속도와 배수량은 여러 조건이 입력될 수 있도록 하였다. 결과물인 최적의 운항자세 그래프에는 가장 연료를 적게 소비하는 자세

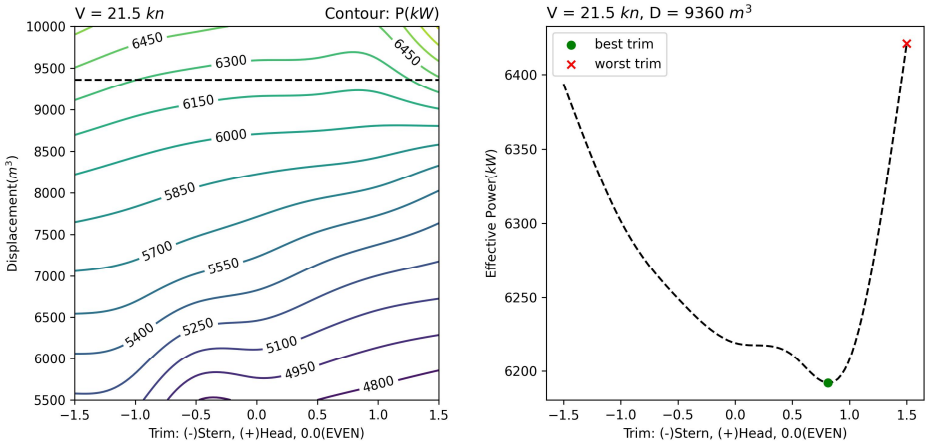

Fig. 5. Example contour graph (left) and cross section graph (right).

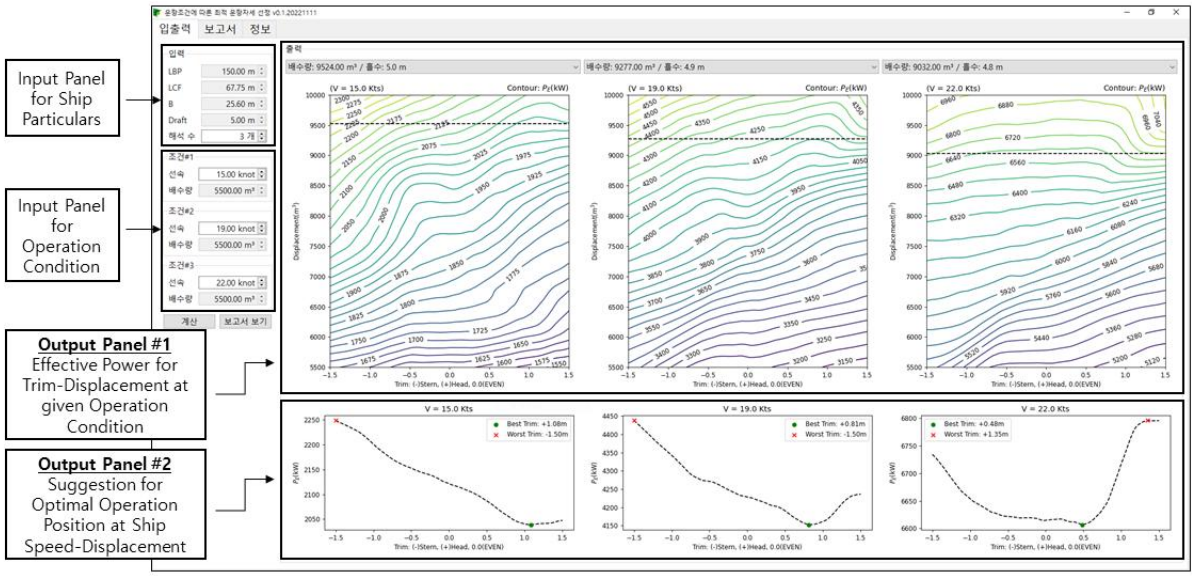

Fig. 6. Optimal trim proposal program.

(Fig. 6의 Best Trim)와 가장 연료를 많이 소비하는 자세(Fig. 6의 Worst Trim)를 제시하도록 하였다. 개발된 GUI는 해당선박의 태블릿 PC와 앱에 설치하여 최적 운항자세 선정에 활용 가능할 것으로 판단된다.

4. 결 론

본 논문은 선박의 유효동력 데이터를 기반으로 주어진 운항조건에서 최대의 에너지효율을 가지는 최적의 운항자세를 선정하는 프로그램 개발과 응용 프로그램을 소개하였다. 본 프로그램은 인공지능 기법에 의한 파이썬 기반 GUI (Graphical User Interface)으로 작성되어 선주가 쉽게 사용할 수 있도록 하였다 결론은 다음과 같다.

(1) 대상선박은 4가지 흘수, 흘수 별로 트림조건 7가지 그리고 선속은 8가지로 총 224가지 데이터를 사용하였다.

(2) 이를 바탕으로 심층학습을 통한 유효동력 모델을 학습하였다. DNN으로 연속적인 운항조건에 대한 제동동력 모델을 학습하였다. 모델의 입력은 속도, 배수량, 자세변화 조건으로 길이가 3인 실수 벡터이며, 출력은 유효동력으로 실수 스칼라로 하였다.

(3) 주어진 운항 속도와 배수량이 정해지면 DNN 모델을 사용해 트림 자세에 따른 유효동력 그래프를 구할 수 있으며, 이를 통해 운항조건에서 가장 효율적인 운항자세를 제안할 수 있었다.

(4) 또한 사용자가 쉽게 사용할 수 있도록 Fig. 6의 GUI 응

용 프로그램을 개발하였으며, 이는 여러 가지 운항조건을 동시에 비교할 수 있도록 인터페이스를 구성하였다.

후 기

이 논문은 2021년도 동명대학교 교내학술연구비 지원에 의하여 연구되었음(2021A035, 202102170001).

References

[1] Ba, J. L., J. R. Kiros, and G. E. Hinton(2016), Layer Normalization. arXiv preprint arXiv:1607.06450.

[2] Han, K. M., H. S. Park, and D. W. Seo(2015), Study on Resistance Component of Container Ship According to Trim Conditions, Journal of Ocean Engineering and Technology, 29(6), pp. 411-417.

[3] Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant(2020), Array programming with NumPy. Nature, 585, pp. 357-362.

[4] Hunter, J. D.(2007), Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, Vol. 9, No. 3, pp. 90-95.

[5] Lee, J. H., J. H. Chun, M. S. Kim, B. J. Park, Y. Yu, Y. Y. Lee, H. Ahn, Y. Kim, and I. Lim(2021), A Study on Trim Variation to Reduce the Required Power of 11,000 TEU Container Ship in Operation Condition, Journal of the Society of Naval Architects of Korea, 58(3), pp. 198-205.

[6] Park, D. W., S. B. Lee, S. S. Chung, H. W. Seo, and J. W. Kwon(2013), Effects of Trim on Resistance Performance of a Ship, Journal of the Society of Naval Architects of Korea, 50(2), pp. 88-94.

[7] Seo, D. W., H. S. Park, and K. M. Han(2015), Analysis of Resistance Performance for Various Trim Conditions on Container Ship using CFD, Journal of Ocean Engineering and Technology, 29(3), pp. 224-230.

[8] Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. van der Plas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors(2020), SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, pp. 261-272.

Received : 2023. 10. 02.
Revised : 2023. 10. 23.
Accepted : 2023. 10. 27.