
 論 文

大韓造船學會誌
 第25卷 第3號 1988年 9月
 Journal of the Society of
 Naval Architects of Korea
 Vol. 25, No. 3, September, 1988

고성능 순차적 선형화 방법을 이용한 선박 최적 초기설계 기법

—최적화 설계 전용 언어의 개발 및 응용—

이 규 열*

Optimum Preliminary Ship Design Technique by Using Sophisticated Sequential Linear Approximation Method

—Development and Application of User Oriented Design Optimization Language—

by

K. Y. Lee*

Abstract

This paper presents a sophisticated Sequential Linear Approximation Method(SLAM) to solve nonlinear optimization problem and the performance of this method is compared with those of the Penalty Function Method (SUMT), Tangent Search Method (TSM) and Flexible Tolerance Method (FTM).

To improve the convenience and flexibility in using the proposed SLAM, an user oriented design optimization language is developed and the application examples are shown for the optimization of propeller principal dimensions and the optimization of bulk carrier principal particulars.

1. 서 언

모든 선박설계는 주어진 제한조건을 만족하면서 최적의 결과를 얻으려는 것이 목적이다. 이러한 견지에서 설계를 다루는 수학적 기법을 최적화 기법이라 할 수 있겠다. 여기서 최적의 결과는 최대의 이익 또는 최소 건조비 또는 최소 중량등을 의미하며 제한조건이란 선주의 요구사항, 항만조건, 안정성 및 각종 규정등을 의미한다.

대부분의 설계문제는 목적함수나 제한조건식이 설계 변수에 대해서 비선형이며 이를 해결하기 위한 기법도 다양하다. 본 논문에서는 순차적 선형화기법(Sequential Linear Approximation Method: SLAM)을 선형화 과정에서 효율적으로 처리하였으며 이 기법과 다른

두가지 비선형 최적화 기법 즉, Penalty Function Method와 Tangent Search Method와 성능을 비교 검토하였으며 또한 Flexible Tolerance Method와도 계산결과를 비교하였다.

이용자가 SLAM 프로그램을 간단하고 편리하게 사용할 수 있도록 일종의 최적설계 전용언어를 개발하여 SLAM프로그램 사용의 편리성, 유연성을 향상시켰다. 또한 이 설계 언어를 이용하여 프로펠러 최적 주요치수 결정 및 개념설계 단계에서의 선박의 최적 주요치수 결정을 위한 설계 Synthesis Model을 구성하였으며 계산 예를 보였다.

2. 비선형 최적화 기법

최적화 문제는 다음과 같이 표시할 수 있다.

본 논문은 대한조선학회 1988년도 춘계연구발표회에서 발표된 논문임.

접수일자: 1988년 5월 16일, 재접수일자: 1988년 7월 7일

* 정회원, 한국기계연구소 대덕선박분소

$$\begin{aligned} &\text{minimize } f(X) \\ &\text{subject to } g_i(X) \geq 0, \quad i=1, 2, \dots, m \\ &\quad \quad \quad h_i(X) = 0, \quad i=m+1, \dots, p \\ &\quad \quad \quad X = (X_1, X_2, \dots, X_n) \end{aligned}$$

그런데 공학문제에서는 목적함수 $f(X)$ 와 제한조건식 g_i, h_i 가 설계변수 X 에 대해서 비선형인 경우가 대부분이며, 이를 해결하기 위한 방법은 대표적으로 다음과 같은 기법이 있다.

2.1. Penalty Function Method

제한조건식이 있는 문제를 부등식 제한조건식들이 목적함수에 미치는 영향을 고려한 penalty항을 목적함수에 덧붙임으로써 제한조건이 없는 문제로 바꾼후, 직접 탐사법을 이용하여 최적점을 찾는 방법이다.

본 논문에서 최적화 기법의 성능 비교시 사용된 방법은 Nelder and Mead의 직접탐사법[1]과 External Penalty Function Technique[2]이었다.

2.2. Tangent Search Method[3]

제한조건식 및 목적함수의 gradient vector를 이용하여 최적점을 찾는 방법으로써, Fig. 1에는 2차원 문제에 대한 예가 표시되어 있다. Feasible region에서는 Hooke and Jeeves의 직접탐사법을 사용하여 최적점을 찾아가다가 새로운점 P_i 가 제한조건식 $g_i=0$ 을 위반하였을때 다시 그전의 점 P_{i-1} 로 되돌아와서 Grad g_i 방향과 수직인 tangent방향('tangent move')으로 나아가서 최적점을 구한다. 이렇게해서 구한 점이 또다시 같은 제한조건식 $g_i=0$ 또는 다른 어떤 제한조건식 $g_{i-1}=0$ 을 위반하였을 때에는 그점에서 다시 'tangent move'를 하게되고 만일 위반하지 않았을 때에는 Hooke and Jeeves의 방법으로 계속 탐사한다.

2.3. Sequential Linear Approximation Method

비선형 목적함수나 제한조건식을 어떤 임의의 점 $X^{(k)}$

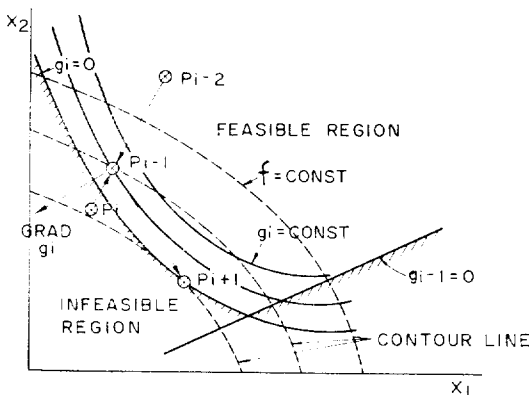


Fig. 1 Tangent search method

(출발점)에서 Taylor series로 전개하여 1차항만 고려하던 목적함수나 제한조건식은 다음과 같이 선형화된다.

$$\begin{aligned} &\text{minimize } f(X) - f(X^{(k)}) \\ &= \sum_{j=1}^n \frac{\partial f(X^{(k)})}{\partial X_j} \cdot (X_j - X_j^{(k)}) \end{aligned}$$

subject to

$$\sum_{j=1}^n \frac{\partial h_i(X^{(k)})}{\partial X_j} \cdot (X_j - X_j^{(k)}) = -h_i(X^{(k)}),$$

$$i=1, \dots, m$$

$$\sum_{j=1}^n \frac{\partial g_i(X^{(k)})}{\partial X_j} \cdot (X_j - X_j^{(k)}) \geq -g_i(X^{(k)}),$$

$$i=m+1, \dots, p$$

여기에서 X_j 의 변화량을 제한하는 식

$$|X_j^{(k+1)} - X_j^{(k)}| \leq \delta_j^{(k)}, \quad j=1, \dots, n$$

을 추가하던 위의 문제는 선형계획법[4]을 이용하여 $X_j^{(k+1)}$ 을 구할 수 있다.

이렇게 해서 얻어진 해 $X_j^{(k+1)}$ 에 대해 목적함수와 제한조건식을 다시 선형화하고 해를 구하면 $X_j^{(k+2)}$ 를 구할 수 있고 이와같은 선형화 작업을 반복하여 얻는 해 $X_j^{(k+3)}, X_j^{(k+4)}, \dots$ 들은 실제 최적점 X_j 에 접근하게 되리라 기대된다.

순차적 선형화 방법의 대표적 예로써 Griffith and Stewart[5] 방법을 들 수 있는데 이 방법의 단점은 목적함수나 제한조건식의 비선형성이 심할 경우에는 최적해에 수렴하지 않거나 일관성 없는해를 찾기때문에 여태까지는 Penalty Function Method나 Tangent Search Method에 비해서 성능이 좋지 않다고 여겨져 왔다.

2.4. 새로운 Sequential Linear Approximation Method

본 고에서는 전통적인 순차적 선형화 방법과는 달리, 선형화 방법과 직접탐사법을 병행하여 사용하면서 최적의 기준에 따라 최적해를 찾으려 하였다. 이때 사용된 최적의 기준은 제한조건식을 위반할 때에는 원래의 목적함수에 위반된 제한조건식을 덧붙인 새로운 함수였다.

2.4.1. 새로운 최적기준의 도입

$$\begin{aligned} F(X) = & f(X) + \left(\sum |g_i| \frac{df_{\max}}{dg_{\max i}} \right. \\ & \left. + \sum |h_i| \frac{df_{\max}}{dh_{\max i}} \right) * (4 + 1000 * L) \end{aligned}$$

for all violated inequality and equality constraints

여기서,

$$df_{\max} = \max_j |\partial f / \partial X_j|$$

$$dg_{\max} = \max_j |\partial g / \partial X_j|$$

$$dh_{\max} = \max_j |\partial h / \partial X_j|$$

L 은 0 또는 1의 값을 가지는데 $L=1$ 인 경우 제한조건식의 영향을 목적함수보다 더 많이 고려하고 $L=0$ 인 경우 제한조건식과 목적함수를 같은 order로 고려한다.

2.4.2. 최적 탐사 방법

본 고에서 사용된 최적화 기법 algorithm은 다음 과정으로 구성된다.

1) 임의의 점 $X^{(1)}$ 과 설계변수의 최대변화량 $\delta^{(1)}$ 을 토대로 목적함수와 제한조건식을 선형화한 후, 선형계획법에 의해 해 $X^{(2)}$ 를 구한다. 이때 단일 선형화 제한조건식을 동시에 모두 만족하는 해를 구할 수 없을 경우에는 제한조건식을 가장 적게 위반하는 Hyperplane의 corner 점을 일단 선형화해 $X^{(2)}$ 로 선정하고 새로운 최적 기준에서의 L 을 1로 둔다.

2) 만일 $F(X^{(2)}) \leq F(X^{(1)})$ 인 경우, 즉 탐사가 성공한 경우 : GOTO 3)

만일 $F(X^{(2)}) > F(X^{(1)})$ 인 경우, 새로운 $X^{(2)}$ 와 $\delta^{(2)}$ 을 정의한다 :

$$X^{(2)} = 0.75 * X^{(1)} + 0.25 * X^{(2)}, \delta^{(2)} = 0.25 * \delta^{(1)}$$

계산 종료 판정기준에 따라 $\delta^{(2)} \leq \epsilon$ (주어진 아주 작은 값)인 경우, 계산 종료하고 그때의 해 $X^{(2)}$ 가 최적해가 되며 그렇지 않는 경우 : GOTO 2)

3) $X^{(2)}$, $\delta^{(2)}$ 를 토대로 선형화 해 $X^{(3)}$ 를 구한다.

만일 $|X^{(3)} - X^{(1)}| \leq \alpha$ (어떤 작은 값)인 경우 :

i) 새로운 $X^{(3)}$ 과 $\delta^{(3)}$ 을 정의한다.

$$X^{(3)} = 0.75 * X^{(2)} + 0.25 * X^{(3)}, \delta^{(3)} = 0.25 * \delta^{(2)}$$

계산종료 판정기준에 따라 $X^{(3)}$ 가 최적해가 아닌 경우 $X^{(2)} \rightarrow X^{(1)}$, $X^{(3)} \rightarrow X^{(2)}$, $\delta^{(2)} \rightarrow \delta^{(1)}$, $\delta^{(3)} \rightarrow \delta^{(2)}$ 로 치환 후 GOTO 2)

ii) 만일 $|X^{(3)} - X^{(1)}| > \alpha$ 인 경우 :

Hooke and Jeeves에서와 같은 "Global Move"를 수행한다 :

$$\text{Global Move : } X = 2 * X^{(3)} - X^{(1)}$$

a) $F(X) \leq F(X^{(2)})$ 인 경우, 즉 탐사가 성공한 경우 :

$X \rightarrow X^{(1)}$ 치환후 GOTO 1)

b) $F(X) > F(X^{(2)})$ 인 경우

$F(X^{(3)}) \leq F(X^{(2)})$ 인 경우 : $X^{(3)} \rightarrow X^{(1)}$ 치환후 GOTO 1)

$F(X^{(3)}) > F(X^{(2)})$ 인 경우 : GOTO 3-i)

3. 새로운 Sequential Linear Approximation Method의 성능

3.1. 세 방법의 성능 비교

제안된 Sequential Linear Approximation Method (SLAM)의 성능을 검토하기 위해 5개의 수학적 문제에 대해 초기값을 변화시켜 가면서 최적값을 구하였고 이를 다른 두가지 방법 즉, Penalty Function Method (SUMT)와 Tangent Search Method(TSM)에 따라 계산한 결과와 비교 검토하였다. Table 1~5에서는 이에 대한 계산결과가 나타나 있다. 여기서 CPU 시간은 PRIME 750 기종에서의 계산시간이다.

Penalty Function Method는 계산시간은 TSM이나 SLAM에 비해서 적게 소요되었으나 정확한 해에 도달하지 못한 경우가 많았으며 또한 초기값이 변화에 따라 일관성 있는 해를 구하지 못하는 경우가 있었다.

Tangent Search Method는 세 방법중 계산시간을 제일 많이 소요하고 있으나 정확성이나 일관성은 SUMT에 비해 월등히 우수하였다.

Sequential Linear Approximation Method는 정확성이나 일관성에서 제일 우수하였고 계산시간도 SUMT와 같은 수준이었다.

Test problem 1 :

$$f(X) = -\sqrt{25 - (X_1 - 5)**2} - (X_2 - 5)**2$$

subject to $32 - 4X_1 - X_2**2 \geq 0$

Test problem 2 :

$$f(X) = (X_2 - X_1**2)**2 + (1 - X_1)**2$$

subject to $X_1**2 + X_2**2 \leq 1$

$$0.2 \leq X_1 \leq 2, 0.2 \leq X_2 \leq 2$$

Table 1 Results for test problem 1

	Starting Point	SUMT	TSM	SLAM	Starting Point	SUMT	TSM	SLAM
X_1	7.0	4.367	4.374	4.376	6.0	4.403	4.372	4.373
X_2	1.0	3.812	3.808	3.807	1.0	3.793	3.809	3.809
f		-4.815	-4.815	-4.815		-4.815	-4.815	-4.815
CPU(sec)		1	8	3		2	8	3

True solution : $X_1=4.374$, $X_2=3.808$, $f=-4.815$

Table 2 Results for test problem 2

	Starting point	SUMT	TSM	SLAM	Starting point	SUMT	TSM	SLAM
X_1	0.5	0.843	0.808	0.808	-5.0	0.807	0.808	0.808
X_2	0.5	0.531	0.589	0.589	5.0	0.590	0.589	0.589
f		0.0571	0.0409	0.0409		0.0409	0.0409	0.0409
CPU(sec)		1	13	2		1	25	2

True Solution : $X_1=0.808, X_2=0.589, f=0.0409$

Table 3 Results for test problem 3

	Starting point	SUMT	TSM	SLAM	Starting point	SUMT	TSM	SLAM
X_1	0.75	0.866	1.0	0.968	0.75	0.886	1.0	0.974
X_2	0.75	0.211	0.0	0.0	0.75	0.238	0.0	0.0
X_3	-0.375	-1.0	-0.947	-0.980	-0.75	-1.0	-0.947	-0.974
X_4	1.3125	1.1	1.05	1.05	1.3125	1.08	1.05	1.05
f		-4.095	-4.792	-4.795		-4.1	-4.790	-4.795
CPU(sec)		8	9	4		6	12	4

True Solution : $X_1=0.974, X_2=0.0, X_3=-0.974, X_4=1.05, f=-4.795$

Table 4 Results for test problem 4

	Starting point	SUMT	TSM	SLAM	Starting point	SUMT	TSM	SLAM
X_1	0.001	Diverge	0.669	0.670	0.001	0.629	0.670	0.670
X_2	0.001		0.0	0.0	0.001	0.0	0.0	0.0
X_3	0.001		-0.671	-0.670	-0.50	-0.708	-0.670	-0.670
X_4	1.051		1.05	1.05	1.051	1.05	1.05	1.05
f			-2.795	-2.795		-2.787	-2.795	-2.795
CPU(sec)			18	4		3	16	4

True Solution : $X_1=0.670, X_2=0.0, X_3=-0.670, X_4=1.05, f=-2.795$

Test problem 3 :

$$f(X) = -\{(X_4-1)**\sin X_2 + (X_1-X_3)**2\}$$

subject to

$$0 \leq X_1 \leq 1$$

$$0 \leq X_2 \leq 2$$

$$-1 \leq X_3 \leq 1$$

$$1.05 \leq X_4 \leq 2$$

$$X_1**2 + X_2**2 + X_3**2 + X_4**2 \leq 3$$

$$X_2 + X_3 + X_4 + X_2*X_3 + X_1*X_4 + X_3*X_4 \leq 2$$

$$X_2**2 - X_2*X_3 + X_3**2 + X_1**2*X_4 - X_1*X_3**2 \leq 6$$

Test problem 4 :

$$f(X) = -\{(X_4-1)**\sin X_2 + (X_1-X_3)**2\}$$

subject to

$$0 \leq X_1 \leq 1$$

$$0 \leq X_2 \leq 2$$

$$-1 \leq X_3 \leq 1$$

$$1.05 \leq X_4 \leq 2$$

$$X_1**2 + X_2**2 + X_3**2 + X_4**2 \leq 2$$

Test problem 5 :

$$f(X) = -X_1*X_2*X_3$$

subject to

$$0 \leq X_1 \leq 20$$

$$0 \leq X_2 \leq 11$$

$$0 \leq X_3 \leq 42$$

$$X_1 + 2(X_2 + X_3) \leq 72$$

3.2. Flexible Tolerance Method와의 성능 비교

비선형 최적화 문제에 또한 많이 이용되는 Flexible Tolerance Method (FTM)와 성능을 비교하기 위해 최근에 발표된 논문[6]에서 다루었던 두가지 수학적 문제에 대해서 계산하였고 그 결과를 Table 6,7에 나타내었다.

정해가 발표되지 않아서 그 정확성을 비교할 수는

Table 5 Results for test problem 5

	Starting point	SUMT	TSM	SLAM	Starting point	SUMT	TSM	SLAM
X_1	18	20	20	20	2	20	20	20
X_2	10	10.617	11	11	2	11	11	11
X_3	16	15.383	14.93	15	2	14.67	14.83	15
f		-3,266	-3,285	-3,300		-3,228	-3,263	-3,300
CPU(sec)		1	4	3		3	6	3

True Solution : $X_1=20, X_2=11, X_3=15, f=-3,300$

Table 6 Results for test case 1

	Starting point	FTM*	SLAM	Starting point	FTM*	SLAM
X_1	5.0	0.998	1.0	-5.0	1.001	1.0
X_2	5.0	4.739	4.743	-5.0	4.693	4.742
X_3	5.0	3.826	3.822	-5.0	3.885	3.824
X_4	5.0	1.380	1.380	-5.0	1.369	1.379
f		17.013	17.014		17.019	17.014
CPU(sec)		5.89	4			4

Table 7 Results for test case 2

	Starting point	FTM*	SLAM	Starting point	FTM*	SLAM
X_1	5.54	5.101	4.611	10.0	5.279	4.616
X_2	4.40	4.282	5.038	10.0	4.668	5.028
X_3	12.02	10.229	8.352	10.0	10.425	8.366
X_4	11.82	11.470	16.990	10.0	12.129	16.967
X_5	0.702	0.805	0.724	1.0	0.756	0.724
X_6	0.852	0.886	0.867	1.0	0.879	0.868
f		133.70	130.89		135.08	130.89
CPU(sec)		7.21	2		16.5	4

* : 참고문헌 [6]으로 부터 인용.

Test Case 1, 2의 수학적 문제는 참고문헌[6]의 No. 1, 2에 해당.

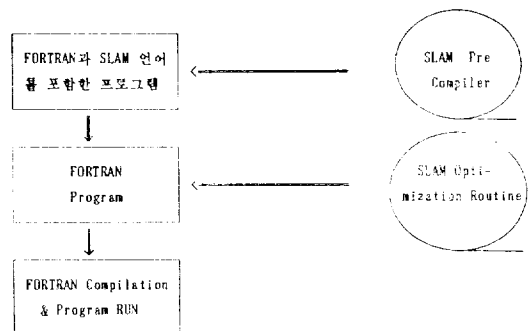
없었으나 Sequential Linear Approximation Method는 FTM에 비하여 일관성이 있었고 계산시간도 초기값이 변하더라도 크게 변하지 않았다.

이상에서 본 바와 같이 SLAM은 SUMT, TSM에 비해서 계산시간, 정확성, 일관성에서 우수하고 FTM에 비해서 일관성이 있는 것으로 판단되었다.

4. 최적화 설계 전용 언어

SLAM 기법을 이용자가 간단하고 편리하게 사용할 수 있도록 SLAM기법 전용설계언어를 FORTRAN 언어로 번역하는 일종의 Pre Compiler를 [7]과 같은 방법으로 개발하였다.

4.1. 시스템 구성



4.2. SLAM 언어 응용 예

한 예로써 “3.1”에서 다루었던 수학적 비선형 문제 중에서 Test Problem 1을 SLAM 기법 전용언어를 이용하면 다음과같이 프로그램을 간단화 할 수 있다.

아래 예에서 보는 바와 같이 FORTRAN 언어와

SLAM 언어를 같이 사용할 수 있으며 SLAM Compiler가 FORTRAN으로 번역하는 부분은 2번째 line의 “BEGIN”과 12번째 line의 “SOLVE” 사이의 모든 statement이다. 4번째 line의 “UNKNOWNNS” 다음에는 4개의 parameter를 가진 설계변수를 정의한다. 첫

```

PROGRAM HILL2
BEGIN
REDUCTION 0.01
UNKNOWNNS
X1, 7.0, 0.4, 0.0, 10.0
X2, 1.0, 0.4, 0.0, 10.0
F, -2.236, 0.4, -10.0, 20.0
SYSTEM
32.-4*X1-X2**2 .GE. 0.
F=-SQRT(25-(X1-5)**2-(X2-5)**2)
MINIMIZE F
SOLVE
WRITE(6,10) X1,X2,F
10  FORMAT(5(2X,E13.6))
STOP
END
TEXTEND

RESULTS
-----
                X1                X2                F
0.437596E+01    0.380738E+01    -0.481542E+01

PROGRAM HILL2
EXTERNAL F1
DIMENSION OPTX1 ( 3),OPTD1 ( 3),OPMI1 ( 3),OPMA1 ( 3),OPSP1 ( 54)
C
OPTX1 ( 1)=7.0
OPTD1 ( 1)=0.4
OPMI1 ( 1)=0.
OPMA1 ( 1)=10.
OPTX1 ( 2)=1.0
OPTD1 ( 2)=0.4
OPMI1 ( 2)=0.
OPMA1 ( 2)=10.
OPTX1 ( 3)=-2.236
OPTD1 ( 3)=0.4
OPMI1 ( 3)=-10.
OPMA1 ( 3)=20.
OPTVER=0.01
CALL OPT(F1, 1, 1,OPTX1, 3,OPTD1,OPMI1,OPMA1,OPTVER,OPTOPT,OPSP1)
X1 =OPTX1 ( 1)
X2 =OPTX1 ( 2)
F =OPTX1 ( 3)
IF (OPTOPT.EQ. 3.E30) THEN
WRITE (*,*) '+++MIN. ONE ROW VECTOR IS ZERO +++'
WRITE (*,*) OPTX1
STOP
ELSE IF (OPTOPT.EQ. 2.E30) THEN
WRITE (*,*) '+++AT LEAST ONE XMIN .GE. XMAX+++'
WRITE (*,*) OPTX1
STOP
ELSE IF (OPTOPT.EQ. 1.E30) THEN
WRITE (*,*) '+++NO SOLUTION FOUND +++'
WRITE (*,*) OPTX1
STOP
END IF
10  WRITE (6,10) X1, X2, F
FORMAT(5(2X,E13.6))
STOP
END
SUBROUTINE F1 (Y,X)
DIMENSION X( 3),Y( 3)
Y(2)=-1.*(0.)-(32.-4*X( 1)-X( 2)**2)
Y(3)=-SQRT(25-(X( 1)-5)**2-(X( 2)-5)**2)-(X( 3))
Y(1)=(X( 3))
RETURN
END

```

Fig. 2 Program list compiled by SLAM-pre compiler

첫번째 parameter는 그 설계변수의 초기값, 두번째는 초기 step size, 세번째와 네번째는 최소값과 최대값을 나타낸다. 위의 예에서 설계변수 X_1 의 경우는 초기값은 7.0, 초기 step size는 0.4, 최소값 0.0, 최대값 10.0을 갖는다.

그러므로 설계변수 자체에 대한 제한조건은 세번째 및 네번째 parameter로써 간단히 정의된다. 8번째 line의 "SYSTEM"에서 부터는 목적함수 및 제한조건식을 정의하는데 그들의 순서는 상관없고 대부분의 FOR-

TRAN 언어를 사용할 수 있으나 FUNCTION Call만 허용된다(SUBROUTINE은 허용되지 않음). 11번째 line의 "MINIMIZE F"는 목적 함수 F를 최소화하며 12번째 line "SOLVE"로써 SLAM 시스템이 끝난다는 것을 표시하고 있다. 그 이후의 line은 정상적인 FOR-TRAN Statement이다.

이상에서 설명한 바와 같이 SLAM 언어를 사용하게 되면 설계변수에 대한 dimension statement도 필요없고, 설계변수에 대한 제한조건식도 간단히 표시할 수

```

VMSDL=VKTSDL*0.51444
REV=RPM/60
BEGIN
REDUCTION REDUC
REAL RLWL, B, T, CB, CM, XLCB, PMCR, SALIN, TEMP, ANZP, ANZB, RMAT, FKAV, REV, DIAMOT,
1 RTCOR, TDCOR, WFCOR, SEAMAR, ETAT, ENGMAR, CWP, TRIM, ABT, HB, SAPP, EIPK2, RLR,
2 DIAMST, DIAMDL, DIAMMI, DIAMMA, VKTSST, VKTSDL, PODST, PODDL,
3 AEAOST, AEAODL, TRSTST, TRSTD, WFST, WFDL, TDST, TDDL.
UNKNOWN
DIAM, DIAMST, DIAMDL, DIAMMI, DIAMMA
VMS, VMSST, VMSDL, 3.0, 20.0
POD, PODST, PODDL, 0.35, 1.4
AEA0, AEAOST, AEAODL, 0.35, 1.0
TRST, TRSTST, TRSTD, 5.0, 5000.0
WF, WFST, WFDL, 0.0, 1.0
TD, TDST, TDDL, 0.0, 1.0
SYSTEM
DIAM, LE, DIAMOT*T
TD=HOLTTD(RLWL, B, T, CB, CM, DIAM, IFIX(ANZP+0.5))*(1.+TDCOR/100.)
TRST=HOLTRS(RLWL, B, T, TRIM, CB, CWP, CM, ABT, HB, SAPP, EIPK2, RLR, XLCB,
1 VMS, SALIN, TEMP, STOT, CVA)*(1.+RTCOR/100.)/(1.-TD)
WF=HOLTWF(RLWL, B, T, CB, CM, DIAM, STOT, TRIM, CVA, IFIX(ANZP+0.5))
1 *(1.+WFCOR/100)
AEAO=(1.3+0.3*ANZB)*TRST/ANZP/((9.81*(0.95*T-0.5*DIAM)+100.
1 -DAMPFD(TEMP))*DIAM*DIAM)+0.05*(3.+RMAT-ANZP)/FKAV
TRST=DICHTE(SALIN)*REV*REV*DIAM**4*THRC(AEAO, POD,
1 VMS*(1.-WF)/(REV*DIAM), IFIX(ANZB+0.5))*ANZP
PMCR=6.283*DICHTE(SALIN)*REV**3*DIAM**5*TORCC(AEAO, POD,
1 VMS*(1.-WF)/(REV*DIAM), IFIX(ANZB+0.5))*ANZP
2 /HOLTR(AEAO, CB, CM, XLCB, POD, IFIX(ANZP+0.5))*(1.+SEAMAR/100.)
3 /ETAT/ENGMAR
4 /0.7355
C MINIMIZE PMCR
MAXIMIZE VMS
SOLVE
VKTS=VMS/0.51444

*****
CALCULATION RESULTS OF RESISTANCE,
WAKE FRACTION AND THRUST DEDUCTION
*****
SPEED OF SHIP(KNOTS) 14.400
EFFECTIVE HORSE POWER(PS) 4438.264
RESISTANCE(KN) 440.666
WAKE FRACTION 0.373
THRUST DEDUCTION FACTOR 0.258
RELATIVE ROTATIVE EFFICIENCY 1.018

*****
CALCULATION RESULTS OF
PROPELLER DIAMETER AND SPEED OF SHIP
*****
SPEED OF SHIP(KNOTS) 14.400
EFFECTIVE HORSE POWER(PS) 4438.264
PROPELLER RPM 100.000
PROPELLER EFFICIENCY 0.579
MAXIMUM CONTINUOUS RATING(PS) 8150.000
OPTIMUM PROPELLER DIAMETER(M) 6.045
PITCH/DIAMETER 0.731
EXPANDED BLADE AREA RATIO 0.456
    
```

Fig. 3 Listing of the SLAM-program for optimization of propeller principal dimensions and ship's speed of 31.5K bulk carrier for given engine power and propeller RPM

있으며, 목적함수와 제한조건식을 기술할때 그 순서에 무관하므로 편리하게 이용자가 프로그램을 작성할 수 있다.

Fig. 2에는 위의 문제를 SLAM-Pre Compiler로써 Compiling한 예가 나타나 있다.

5. 프로펠러 최적 주요치수 결정

초기설계 단계에서 선박의 주요치수, 주기관 마력 및 회전수가 주어졌을 때 최고의 효율을 갖는 프로펠러 최적 주요치수를 SLAM 기법을 이용하여 다음과 같이 결정하였다.

5.1. 문제의 정식화

given : engine power (=PMCR), rpm (=n)
 results : Optimum propeller diameter (=Dp),
 Pitch(=P), blade area ratio(=AE/AO)
 and maximum speed of ship (=V)

design model :

속력에 따른 저항(=RT), 반류계수(=w), 추력감소계수(=t) 및 프로펠러 효율비는 Holtrop and Mennen 방법으로 구하였고, 프로펠러 주요치수 결정에는 Wageningen B-series 프로펠러 시험결과를 이용하였다. 또한 최소면적비 AE/AO는 Keller의 cavitation 발생제한 기준을 이용하였다.

상기문제를 수식화하면 다음과 같다.

설계변수 : Dp, P, AE/AO, V

주어진 변수 : PMCR, n

제한조건식 :

$$RT/(1-t) = \rho * n^3 * 2 * Dp^4 * K * T(J, AE/AO, P/$$

$$Dp)$$

여기서,

$$J = V * (1-w) / (n * Dp)$$

$$PMCR = \rho * 2 * \pi * n^3 * 3 * Dp^5 * KQ(J, AE/AO, P/Dp)$$

$$AE/AO = (1.3 + 0.3 * Z) * (RT / (1-t)) / ((P_0 - P_v) * Dp^2) + K$$

여기서,

Z : No. of propeller blade

P₀ : Static pressure at shaft center line

P_v : vapour pressure

K : 상수

목적함수 :

maximize V

즉, 미지수(설계변수)가 4개이고 등식이 3개이므로 1개의 자유변수를 갖는 비선형 최적화 문제로 귀착된다.

5.2. 프로펠러 최적 주요치수 결정 프로그램 및 계산결과

Fig. 3에는 SLAM 기법을 이용한 프로그램이 나타나 있으며 이를 이용하여 DWT 31,500ton급 bulk carrier의 최적 프로펠러 주요치수를 계산한 예를 함께 수록하였다. 참고로 이 경우의 실제선 프로펠러의 주요치수 및 속력은 Dp=5.7m, P/Dp=0.84, AE/AO=0.62, V=14.2kts이었다.

계산결과에 따르면 실제선 자료보다 선속과 프로펠러 직경이 커진것을 볼 수 있는데 이는 Keller의 cavitation식을 이용하여 계산된 전개면적비가 실제선자료보다 작아졌기 때문이다.

```

WF=HOLTWF(RLWL, B, T, CB, CM, DIAM, STOT, TRIM, CVA, IFIX, (ANZP+0.5))
1 * (1 + WFCOR/100)
C AEAO=(((1.3+0.3*ANZB)*TRST/ANZP)/((9.81*(0.95*T-0.5*DIAM)+100.
C 1 -DAMPFD(TEMP))*DIAM*DIAM)+0.05*(3.+RMAT-ANZP))/FKAV
AEAO=0.62
    
```

 CALCULATION RESULTS OF
 PROPELLER DIAMETER AND SPEED OF SHIP

SPEED OF SHIP(KNOTS)	14.182
EFFECTIVE HORSE POWER(PS)	4223.365
PROPELLER RPM	100.000
PROPELLER EFFICIENCY	0.550
MAXIMUM CONTINOUS RATING(PS)	8150.000
OPTIMUM PROPELLER DIAMETER(M)	5.883
PITCH/DIAMETER	0.769
EXPANDED BLADE AREA RATIO	0.620

이번에는 전개면적비 $AE/AO=0.62$ 를 Input으로 준 경우의 최적 프로펠러 치수를 구하였다.

이때 Fig. 3의 SLAM-Program 중 "A" 부분의 AE/AO ...만 단순히 $AE/AO=0.62$ 로 변형시키면 된다.

아래에는 수정된 program statement와 이에 따른 계산결과가 나타나 있다. 이에 따르면 $AE/AO=0.62$ 를 고정한 경우, $V=14.182$ kts, $Dp=5.89$ m로 거의 실제선 자료와 일치하는 것을 알 수 있다.

6. 선박의 최적 주요치수 결정

개념설계 단계에서 요구 재화중량, 화물창 용적 및 속력이 주어졌을때 건조비를 최소로 하는 bulk carrier 최적 주요치수를 SLAM 기법을 이용하여 다음과 같이 결정하였다.

6.1. 문제의 정식화

given : Deadweight(=Dw), Hold Capacity(=Vol), Speed of Ship(=V), Draft(=T), Propeller rpm(=n)

Results :

L, B, D, CB, Light ship weight (=WLS), Engine Power(=PMCR), RT, Dp, P, AE/AO, Cost

제한조건식 :

$$L*B*T*Cb*1.03=Dw+WLS(L, B, D, T, Cb, PMCR)$$

$$Vol=Vol(L, B, D, T, Cb)$$

$$D \geq T + \text{Freeboard}(L, B, D, T, Cb, \text{Super Structure})$$

$$GM \geq 0.04*B$$

$$RT=RT(L, B, D, T, Cb, CM, LCB, V)$$

$$RT/(1-t) = \rho * n^{**2} * Dp^{**4} * KT(J, AE/AO, P / Dp)$$

$$PMCR = \rho * 2\pi * n^{**3} * Dp^{**5} * KQ(J, AE/AO, P / Dp)$$

$$AE/AO = (1.3 + 0.3*Z) * (RT / (1-t)) / ((P_0 - P_v) * Dp^{**2}) + K$$

$$\text{Cost} = \text{Cost}(L, B, D, T, Cb, PMCR)$$

목적함수 :

minimize cost

상기 문제는 미지수, 즉 "Results" 항에 기술되어 있는 설계변수가 11개이고 등식이 7개이므로 4개의 자유변수와 2개의 부등식을 갖는 비선형 최적화 문제로 귀착된다.

6.2. Bulk Carrier 최적 주요치수 결정 프로그램 및 계산결과

Fig. 4에는 SLAM기법 전용 설계언어를 이용한 최적 주요치수 결정 프로그램이 나타나 있다. "SYSTEM"과 "SOLVE" 사이의 주요등식 및 부등식은 다음과 같다.

$$DW+WLS=HYSTA(RL, B, T, CB, XLCB, RLWL, CM, Cwp)$$

은 배수량과 주요치수와의 관계를 나타내며

$$WLS=WEIGHT(RL, B, D, T, CB, PMCR, WS, WO, WM)$$

은 경하중량, 선각중량, 선체의장부 중량 및 기관부 중량을 계산하며, $VOL=HOLDBK(RL, B, D, T, CB)$ 는 요구하는 화물창 용적을 만족하는 등식이다. 그 다음 line의 D. GE. T+FREBD(RL, ...)은 $D \geq T + \text{Freeboard}$ 를 나타내는 부등식이며 "FREBD"는 ILLC 1966에 따라 건현을 계산한다. 그 다음의 statement는 "5.2"에서 언급된 저항, 추진계수, 프로펠러 주요치수, 기관마력의 관계를 나타내는 식들이다. 그다음에는 초기 복원성능을 표시하는 GM이 계산되고 GM의 최대치와 최소치가 선박의 주요치수의 함수로써 표시되었고, 또한 주요치수에 대한 제한조건식들이 표시되어 있다.

마지막으로 $COST=COSTEN(PMCR, WS, WD, RL, B, D)$ 은 건조비를 추정하는 식이다. Fig. 4에는 DWT 50,000ton급 bulk carrier의 주요치수를 SLAM-Program을 이용하여 결정한 예가 나타나 있다. 이 프로그램에서 사용된 경하중량 추정, 건조비 추정 방법은 개략적인 것이기 때문에 절대적인 값에는 상당한 오차가 있을 것으로 사료되나 동일한 방법을 사용하였을 경우 상대적인 비교목적에는 유용하리라 생각된다.

7. 결 론

비선형 최적화 기법인 Sequential Linear Approximation Method(SLAM)의 성능을 선형화 과정에서의 오차를 최소화하여 향상시켰으며 이를 Penalty Function Method, Tangent Search Method 및 Flexible Tolerance Method와 비교 검토한 결과, 정확성, 일관성면에서 그 우수성을 확인할 수 있었고, 계산시간면에서는 제일빠른 Penalty Function Method 보다는 시간이 좀더 소요되지만 같은 수준이라고 볼 수 있다.

이용자가 간단하고 편리하게 사용할 수 있는 SLAM 기법 전용 설계언어를 이해하는 일종의 Pre Compiler를 개발하여 프로그램의 편리성, 유연성 및 생산성을

높혔다.

선박의 최적 주요치수 결정문제, 프로펠러 최적 주요치수 결정문제에 대해서 SLAM 기법을 이용한 design synthesis model을 구성하였고, bulk carrier에 대해서 그 계산결과를 보였다. 프로펠러 최적 주요치수의 계산결과는 실적선 자료와 거의 유사한 것으로 나타났으며, 선박 최적 주요치수 결정시에는 증량추경, 전조비 추정방법이 개략적인 것이어서 계산결과와 절대적인 값은 오차가 많을 것으로 사료되나 상대적인 비교 분석시에는 유용할 것으로 기대된다.

앞으로는 이러한 선박의 기본성능 추정 즉, 증량추경, 용적추경, 저항추진 성능추정 및 전조비 추정을 정확히 할 수 있는 방법을 개발하여 SLAM 기법을 이용한 design synthesis model에 종합화하던 초기설계 단계에서 보다 정도 높은 설계를 신속, 정확히 수행할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] Nelder, J.A. and Mead, R., "A Simplex Method for Function Minimization", *Computer Journal*, Vol. 7, No. 4, pp.308-313, Jan. 1965.
- [2] Zangwill, W.I., "Nonlinear Programming Via Penalty Functions", *Management Science*, Vol. 13, No. 5, pp.344-358, Jan. 1967.
- [3] Hilleary, R.R., "The Tangent Search Method for Constrained Minimization", U.S. Naval Postgraduate School Technology Report/Research Paper No. 59, Mar. 1966.
- [4] Dantzig, G.B., "Linear Programming and Extension", Princeton University Press, Princeton, N.J., 1963.
- [5] Griffith, R.E. and Stewart, R.A., "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems", *Management Science*, Vol. 7, No. 4, 1966.
- [6] 임상전, 김기화, 양영순, "FTM을 이용한 파형 격벽의 최적구조설계", 대한조선학회지, 24권 4호, 1987.12.
- [7] Soeding, H., "Design Software, Optimization in Ship Design", WEGEMT 1978.
- [1] Nelder, J.A. and Mead, R., "A Simplex Method for Function Minimization", *Computer Journal*,