

입자 군집 최적화를 이용한 함정 소자코일 교정인자 산출

Obtaining Degaussing coil input parameters using PSO

이정우*, 임선호, 도재원

Lee Jung-Woo, Lim Sun-Ho, Doh Jae-Won

LIG 넥스원(주) 연구개발본부 Maritime 1연구센터

jungwoo.lee@lignex1.com

1. 서론

함정 선체인 강자성체에 의해 발생하는 영구자기장과 유도자기장을 저감하여 자기 스틸스를 실현하는 방법으로 탈자와 소자가 사용된다. 탈자는 함정의 영구자기장을 제거하는 방법으로 연안에 별도의 탈자시설을 갖추고 Deperm-R, Flash-D 등의 규격화된 탈자 절차에 따라 탈자를 진행하는 수동적 스틸스 기법이다. 반면, 소자는 탈자처리 후 함정에 남아 있는 잔영구자기장과 지구자기장에 의한 유도자기장을 최소화시키기 위한 방법으로 함정 내부에 3축 방향으로 코일을 설치하고 코일에 전류를 인가하여 수중 함정 자기장 신호를 상쇄하는 능동 스틸스 기법이다[1]. 이 때, 소자코일에 인가되는 전류를 소자코일 교정인자라고 부르며, 소자코일 교정인자 산출 문제는 현재 함정 자기장을 상쇄할 수 있는 자기장을 발생시키기 위한 다수의 소자코일 교정인자를 결정하는 조합 최적화 문제로 표현될 수 있다. 한편, 입자 군집 최적화(Particle Swarm Optimization, PSO)는 Kennedy와 Elberhart에 의해 제안된, 조합 최적화 문제를 효율적으로 풀기 위한 메타 휴리스틱 알고리즘이다[2]. PSO는 GA와 유사하게 개체군집(population)에 기반한 최적화 방법이며, 기초 연구 분야 뿐만 아니라, 실제 엔지니어링 응용 프로그램에도 다수 적용되어 그 성능이 입증되었다[3-5]. 본 논문에서는 PSO를 이용하여 함정 소자코일 교정인자 산출 문제를 푸는 방법을 제안하고자 한다. 이를 위해, PSO에 대한 소개와 소자코일 교정인자 산출 문제에 대한 정의를 하고, 실험을 통해 제안된 방법의 타당성을 보일 것이다.

2. 입자 군집 최적화(Particle Swarm Optimization, PSO)

PSO는 물고기, 새떼와 같이 군집활동을 하는 집단의 행동양식을 모방한 자연계 기반 확률적 계산최적화 방법이다. Particle 은 최적화된 함수공간 상의 하나의 해를 나타낸다. 통상 함수 매개변수의 수에 따라 d 차원 벡터로 표시되며, 여기서는 i 번째 particle이 k 번째 탐색한 위치벡터를 다음과 같이 나타낸다.

$$X_i^k = [x_{i1}^k, x_{i2}^k, \dots, x_{id}^k].$$

각 particle은 각자가 경험한 최 해의 위치 pbest를 가지며 집단전체는 집단 전체적으로 경험한 최적해의 위치 gbest를 가진다. PSO에서 각 입자들은 현재의 위치 벡터와 pbest, gbest를 이용하여 다음과 같이 next iteration에서의 자신의 속도를 갱신한다.

$$v_{id}^{k+1} = w \times v_{id}^k + c_1 \times rand() \times (pbest_i - x_{id}^k) + c_2 \times rand() \times (gbest - x_{id}^k)$$

또한, 수식 5에 의해, 자신의 위치를 갱신한다.

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}$$

즉, PSO는 탐색을 반복하면서 얻어진 gbest와 pbest의 방향으로 각 particle의 현재 위치를 수정해 나가면서 전역 최적해를 찾기를 확률적으로 기대하는 알고리즘이다[3].

3. 소자코일 교정인자 산출 문제

한국 해군의 기존 함정자기 측정소에서는 각 소자코일 교정인자 구하기를 오로지 운용자의 경험과 측정 수치의 육안 확인에 의존해 왔고, 실제로 이는 각 함정 당 수십번의 기동 측정(약 2주)을 필요로 했다. 이 작업은 다음의 중복순열에 의한 크기의 해공간을 탐색하는 것과 같다. 소자코일의 개수를 N , 코일 $n(1 \leq n \leq N)$ 에 인가 가능한 전류치

의 집합을 A_m 그리고 $|A_1|=|A_2|=...=|A_N|=m$ 이라고 가정하면, 이 문제의 해집합의 개수는 m^N 이 된다. 실제 한국 해군이 보유한 소자장비는 $7 \leq N \leq 21$, $800 \leq m \leq 3000$ 정도가 되므로, 이를 Bruce-force 같은 무작위 탐색 방법을 이용하여 해를 찾고자 할 경우, 한 인스턴스당 컴퓨터 계산시간을 0.003초라 가정하더라도, 최소 $1.9950e+010$ 년이라는 총 계산 시간이 나온다. 따라서 본 논문에서는 이 문제를 다음과 같은 목적함수로 표현하고 이를 최소화하는 전류의 인스턴스를 찾는 최적화 문제로 모델링할 것이다. 목적함수는 다음과 같다.

$$f(x) = \max_s |B + Cx| \quad \text{식(1)}$$

x : N개의 코일 전류값에 해당하는 N차원 공간 상의 instance
 B : 상쇄하고자 하는 합성 자기장 센서 측정치
 $C = [C_1 C_2 \dots C_N]$
 C_n : 코일 n의 소자코일효과 벡터 ($1 \leq n \leq N$)
 s : 센서 번호 ($1 < s < S$)
 S : 총 센서 갯수

어떤 코일n에 1AT의 전류가 인가되었을 때 발생하는 자기장을 그 코일의 소자코일효과(C_x)라고 한다. 각 코일에 $x=[x_1, x_2, \dots, x_N]$ 의 전류를 인가하였을 때의 소자코일효과 C_x 와 합성의 남은 자기장(B)의 합은 S개의 센서 각각에서 감지 되는데, 이 S개의 센서 중 가장 큰 값을 보이는 센서를 찾아 이를 최소화시키는 것이 목적이다. 따라서 이 문제의 해는 다음과 같이 $f(x)$ 를 최소화(minimization)시키는 전류값 x 를 찾는 것이 된다.

$$\min_x f(x) \quad \text{식(2)}$$

4. 실험

제안된 방법의 모의 실험은 합성 길이방향 유도자기장(ILM)의 보상을 기반으로 하였다. 상쇄시킬 합성 길이방향 유도자기장과 합성 소자코일효과 데이터는 FEM(유한요소해석) 도구를 이용한 모델링을 통해 획득하였다.

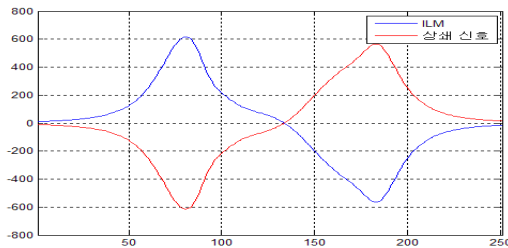


그림 1. 모델 합성 ILM 및 보상 신호

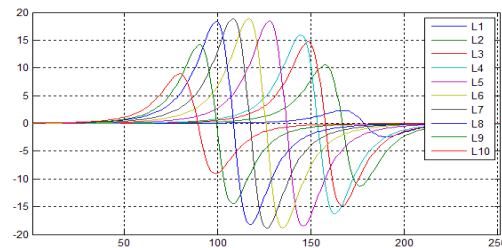


그림 2. 모델 합성 L코일효과

그림 1의 ILM(푸른선)을 보상하기 위해 그림 2의 L코일효과의 선형조합에 의한 보상신호(붉은선)를 생성해야 하며(식(1)을 최소화(minimize)하기 위한 방법 도식), 이는 각 코일에 인가할 적절한 교정인자를 산출하는 것이다. 이 문제를 풀기 위한 PSO의 파라미터는, particle수:20, 반복횟수:200, $C_1=C_2=2$ 이고, inertia weight는 초기 0.9로 시작하여 0.4까지 선형적으로 감소시켰다. 표1은 각 코일에 인가할 전류치이고 그림3은 그로 인해 상쇄된 합성 자기장이다.

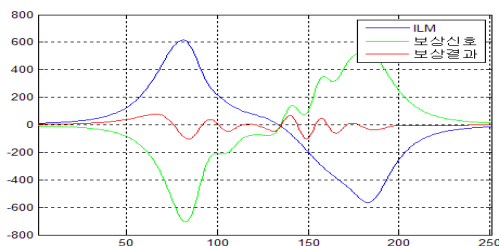


그림 3. 보상결과

L1	-81.28	L6	-13.01
L2	-28.93	L7	-12.21
L3	14.18	L8	-17.03
L4	-38.74	L9	-9.66
L5	-24.34	L10	-48.32

표 1. L코일 교정인자(A)

5. 결론

본 논문에서는 조합최적화 문제를 풀기 위한 대표적인 메타휴리스틱 알고리즘인 PSO를 이용하여 함정 소자코일 교정인자 산출 문제를 풀이하였다. 결과는 소자코일 전류 인가 전(peak : 615.75)에 비해 산출된 교정인자 인가 후 (peak : 103.43) 약 83%의 소자효과를 보았다.

참고문헌

- [1] Timothy Malcolm Baynes BSc. (2002), Analysis of the Demagnetisation Process and Possible Alternative Magnetic Treatment for Naval Vessel : Doctor of Philosophy's thesis, The University of New South Wales, School of Physics, Faculty of Science.
- [2] Kennedy J., Eberhart R.C., , Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, pp.1942-1948,1995.
- [3] Angeline P.J., Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, Evolutionary Programming, vol. VII, Berlin, Springer, pp. 601-610, 1998.
- [4] Clerc M., Kennedy J., The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation, Vol. 6, pp. 58-73.2002.
- [5] Trelea I.C., The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters, Vol. 85, pp. 317-325, 2003.