

II. 문제설정 및 Sword 알고리즘의 개요

그림 1에서와 같이 N 개의 표적이 있을 때, 등간격으로 배열된 M 개의 센서로부터 얻은 센서출력을 이용하여 표적의 방위각 $(\theta_i(t), i=1, 2, \dots, N)$ 을 추정하고자 한다. 시간 t 에서 k 번째 센서의 출력은 협대역신호모델(narrowband signal model)을 사용하여 다음 식과 같이 표현될 수 있다.

$$r_k(t) = \sum_{i=1}^N s_i(t)e^{j\omega_k t} + n_k(t) \quad (1)$$

여기서 첫번째 센서에 도달한 $s_i(t) = A_i(t)e^{j\omega t}$ 는 ω 를 전송주파수(carrier frequency)로 가지는 i 번째 표적의 복소신호이다. n_k 는 평균이 0이고 분산이 σ^2 인 k 번째 센서의 백색잡음(white noise)이다. 센서잡음들은 서로 상관관계가 없으며, 표적신호와도 상관관계가 없다고 가정한다. $\tau_{ik}(t)$ 는 i 번째 표적신호에 의한 첫번째와 k 번째 센서사이의 시간지연이다. 표적이 센서로부터 멀리 떨어져 있을 경우, 센서에 도달되는 표적신호는 평면파로 간주될 수 있으며, 시간지연과 방위각 사이에는 다음 관계식이 존재한다.

$$\tau_{ik}(t) = (d/c)(k-1)\sin\theta_i(t) \quad (2)$$

c 는 음파의 속도이며, d 는 두 센서의 간격이다.

센서출력을 다음 식과 같이 벡터로 표현할 수 있다.

$$r(t) = A(t)s(t) + n(t) \quad (3)$$

여기서 출력벡터 $r(t)$ 와 신호벡터 $s(t)$ 와 잡음벡터 $n(t)$ 는 다음과 같다.¹⁾

$$\begin{aligned} r(t) &= [r_1(t), r_2(t), \dots, r_M(t)]^T \\ s(t) &= [s_1(t), s_2(t), \dots, s_N(t)]^T \\ n(t) &= [n_1(t), n_2(t), \dots, n_M(t)]^T \end{aligned} \quad (4)$$

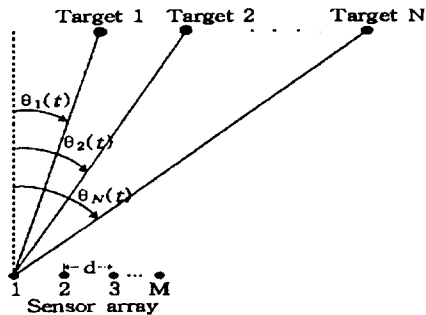


그림 1. 표적과 배열 센서의 기하학적 구조
Fig. 1. Array geometry for tracking.

행렬 $A(t)$ 는 방위행렬(direction matrix)이며, 다음 식과 같이 된다.

$$A(t) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \gamma_1(t) & \gamma_2(t) & \dots & \gamma_N(t) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1^{M-1}(t) & \gamma_2^{M-1}(t) & \dots & \gamma_N^{M-1}(t) \end{bmatrix} \quad (5)$$

여기서 $\gamma_i(t)$ 는 다음 식과 같다.

$$\gamma_i(t) = e^{j(d/c)\sin\theta_i(t)}, \quad i = 1, 2, \dots, N \quad (6)$$

식 (3)과 가정들로부터 센서출력 공분산행렬 $R(t)$ 는 다음 식과 같이 표현된다.

$$\begin{aligned} R(t) &= E\{r(t)r^H(t)\} \\ &= A(t)SA^H(t) + \sigma^2 I \end{aligned} \quad (7)$$

여기서 $S = E\{s(t)s^H(t)\}$ 는 신호공분산행렬이다.

Sword 등은 샘플링 시간 $k+1$ 에서의 방위행렬 $A(k+1)$ 를 다음 식과 같이 정의하였다.

$$A(k+1) = A(k) + \Delta(k) \quad (8)$$

$\Delta(k)$ 는 샘플링 주기동안의 방위행렬 변화량이며, 표적의 방위각 변화량이 반영되어 있다. 센서출력 공분산행렬의 변화량은 방위행렬 변화량 $\Delta(k)$ 에 대한 Riccati 형태의 행렬방정식인 다음 식과 같이 표현된다.

$$\begin{aligned} R(k+1) - R(k) &= M(k)\Delta^H(k) + \Delta(k)M^H(k) + \Delta(k)S\Delta^H(k) \end{aligned} \quad (9)$$

여기서 $M(k) = A(k)S$ 이다.

식 (9)를 $\Delta(k)$ 에 대하여 풀면 샘플링 주기동안의 방위각 변화량을 구할 수 있다. 그러나 식 (9)는 비선형 방정식이므로 해를 직접 구하기가 어렵다. 식 (9)를 현재의 방위각에서 선형근사화하여 방위각 변화량에 대한 선형 방정식을 얻고, 이를 최소자승법으로 풀면 방위각 변화량을 구할 수 있다. 이것을 현재의 방위각에 더하여 증으로써 표적의 방위각 제적을 추적하게 된다.

샘플링 주기동안의 방위각 변화량을 선형근사화하는 것은 근사화 오차가 커서 추정 오차를 크게 하는 중요한 원인이 된다. 본 논문에서는 현재까지 누적된 정보를 이용하여 방위각을 예측하고, 이 방위각 예측치를 이용하여 선형근사화함으로써 근사화 오차를 줄여 추정 오차를 줄였다. 이 때 방위각을 예측하기 위하여 간단한 구조의 칼만 필터를 사용하였다[1]. 표적이 근접하거나 교차할 때 선형근사화 식의 해가 비정상적으로 커지는 경향이 있으므로 새로운 가격함수를 정의하고, 이를 최소화하는 해를 이용함으로써 교차하는 표적의 추적 성능을 향상시켰다.

1) 본 논문에서 T 와 H 는 전치(transpose)와 복소공역전치(complex conjugate transpose)를 각각 나타내며, $E(\cdot)$ 는 확률 평균을 나타낸다.

III. 수정된 다중표적 추적 알고리즘

i 번째 표적의 방위각에 대한 상태방정식과 측정방정식은 다음 식과 같이 모델링된다.

$$\begin{aligned} x_i(k+1) &= Fx_i(k) + w_i(k) \\ z_i(k) &= Hx_i(k) + v_i(k) \end{aligned} \quad (10)$$

$x_i(k) = [\theta_i(k) \dot{\theta}_i(k)]^T$ 는 방위각과 방위각 변화율로 구성되는 2차 상태벡터이다. $z_i(k)$ 는 측정방위각이다. $w_i(k)$ 와 $v_i(k)$ 는 프로세스잡음과 측정잡음이며, 평균이 0이고 분산이 각각 $Q_i(k)$ 와 $\sigma_{z_i}^2(k)$ 인 정규분포를 갖는다. F 와 H 는 상태천이행렬과 측정행렬이며, 정속도운동모델에 대하여 다음 식과 같이 표현된다.

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}; H = [1 \ 0] \quad (11)$$

여기서 T 는 샘플링 주기이다.

초기에 센서출력 공분산행렬에 MUSIC 알고리즘을 적용하여 표적의 개수 및 초기 방위각, 신호공분산행렬, 센서잡음크기를 추정하고[7], T 초 후에 MUSIC 알고리즘을 한번 더 적용한다. 방위각 추정치 ($\hat{\theta}_i(-1)$, $i=1,2,\dots,N$)와 ($\hat{\theta}_i(0)$, $i=1,2,\dots,N$)를 이용하여 각 표적의 상태벡터와 상태벡터의 공분산행렬을 다음 식과 같이 초기화한다[1].

$$\begin{aligned} \hat{\theta}_i(0|0) &= \hat{\theta}_i(0) \\ \dot{\hat{\theta}}_i(0|0) &= \frac{\hat{\theta}_i(0) - \hat{\theta}_i(-1)}{T} \\ P_i(0|0) &= \begin{bmatrix} \sigma_{z_i}^2 & \sigma_{z_i}^2/T \\ \sigma_{z_i}^2/T & 2\sigma_{z_i}^2/T^2 \end{bmatrix}, \quad i=1,2,\dots,N. \end{aligned} \quad (12)$$

시간 k 에서 i 번째 표적의 상태벡터추정치 $\hat{x}_i(k|k)$ 와 이것의 공분산행렬 $P_i(k|k)$ 이 주어져 있을 때, 제안한 알고리즘의 한 주기는 다음과 같이 3 단계로 요약할 수 있다.

단계 1. 예측

상태벡터 추정치 $\hat{x}_i(k|k)$ 와 이것의 공분산행렬 $P_i(k|k)$ 를 이용하여 상태벡터 예측치 $\hat{x}_i(k+1|k)$ 와 이것의 공분산행렬 $P_i(k+1|k)$ 를 다음 식과 같이 구한다.

$$\begin{aligned} \hat{x}_i(k+1|k) &= F \hat{x}_i(k|k) \\ P_i(k+1|k) &= F P_i(k|k) F^T + Q_i(k), \quad i=1,2,\dots,N \end{aligned} \quad (13)$$

$\hat{x}_i(k+1|k)$ 의 첫번째 항이 방위각 예측치 $\hat{\theta}_i(k+1|k)$ 가 되며, 이를 식 (5)에 대입하면 방위행렬 예측치 $\hat{A}(k+1|k)$ 를 얻을 수 있다. 이것을 식 (7)에 대입하여 센서출력 공분산행렬의 예측치 $\hat{R}(k+1|k)$ 를 다음 식을 이용하여 예측할 수 있다.

$$\hat{R}(k+1|k) = \hat{A}(k+1|k) S \hat{A}^H(k+1|k) + \sigma^2 I \quad (14)$$

단계 2. 이노베이션 계산

T 초 후에 표적들은 새로운 방위각 위치로 이동한다. 이 때의 방위행렬을 다음 식과 같이 정의한다.

$$A(k+1) = \hat{A}(k+1|k) + \delta A(k+1) \quad (15)$$

여기서 $\delta A(k+1)$ 는 방위행렬의 예측오차이다. 새로운 센서출력 공분산행렬 $R(k+1)$ 은 식 (7)로부터 다음과 같이 추정할 수 있다.

$$R(k+1) = A(k+1) S A^H(k+1) + \sigma^2 I \quad (16)$$

식 (15)를 식 (16)에 대입하고, 이것에서 식 (14)을 빼면, $\delta R(k+1) = R(k+1) - \hat{R}(k+1|k)$ 을 다음 식과 같이 표현할 수 있다.

$$\begin{aligned} \delta R(k+1) &= M(k+1) \delta A^H(k+1) + \\ &\delta A(k+1) M^H(k+1) + \delta A(k+1) S \delta A^H(k+1) \end{aligned} \quad (17)$$

여기서 $M(k+1) = \hat{A}(k+1|k) S$, $M^H(k+1) = S \hat{A}^H(k+1|k)$ 이다.

시간 $k+1$ 에서 i 번째 표적에 대한 칼만 필터의 방위각 이노베이션 $\delta \theta_i(k+1)$ 를 다음 식과 같이 정의한다.

$$\delta \theta_i(k+1) = z_i(k+1) - \hat{\theta}_i(k+1|k) \quad (18)$$

$\delta \theta_i(k+1)$ 는 확률변수(random variable)이며, 평균이 0이고 분산은 다음 식으로 구한다.

$$V_{\delta \theta_i} = H P_i(k+1|k) H^T + \sigma_{z_i}^2. \quad (19)$$

방위각 이노베이션은 식 (17)을 선형근사화하여 최소자승법으로 구할 수 있다. 식 (17)은 Sword 알고리즘의 식 (9)와 형태가 같으며, 단지 $A(k)$ 대신에 $\hat{A}(k+1|k)$ 로 대체되었을 뿐이다. 그러므로 Sword 알고리즘과 동일한 선형근사화 과정을 거친 후 다음 식과 같은 선형근사화된 식을 얻는다.

$$y = B \delta \theta \quad (20)$$

식 (20)에서

$$\begin{aligned} y &= (\delta R \text{의 첫번째 열을 없앤 첫번째 행}) \\ &= [\delta R_{12}, \delta R_{13}, \dots, \delta R_{1M}]^T \end{aligned}$$

$$\delta\theta = [\delta\theta_1, \delta\theta_2, \dots, \delta\theta_N]^T$$

이며, 행렬 B 는 다음 식으로 정의한다.

$$B = j \frac{d}{c} \omega \cdot \begin{bmatrix} \cos(\hat{\theta}_1) \hat{\gamma}_1 u_1 & \dots & \cos(\hat{\theta}_N) \hat{\gamma}_N u_N \\ 2\cos(\hat{\theta}_1) \hat{\gamma}_1^2 u_1 & \dots & 2\cos(\hat{\theta}_N) \hat{\gamma}_N^2 u_N \\ \vdots & & \vdots \\ (M-1)\cos(\hat{\theta}_1) \hat{\gamma}_1^{M-1} u_1 & \dots & (M-1)\cos(\hat{\theta}_N) \hat{\gamma}_N^{M-1} u_N \end{bmatrix} \quad (21)$$

여기서

$$\hat{\theta}_i = \hat{\theta}_i(k+1 | k)$$

$$\hat{\gamma}_i = \hat{\gamma}_i(k+1 | k)$$

$$u_i = M u_i(k+1), \quad i=1,2,\dots,N$$

이다.

표적이 교차할 때 추적 성능을 향상시키기 위하여 방위각 이노베이션의 크기를 제한하는 항을 가지는 새로운 가격함수를 다 음 식과 같이 정의하였다.

$$J = [y - B\delta\theta]^H [y - B\delta\theta] + \delta\theta^T L \delta\theta \quad (22)$$

대각행렬 $L = \text{diag}[l_1, l_2, \dots, l_N]$ 은 방위각 이노베이션의 크기를 제한하는 가중행렬(weighting matrix)이다. 가격함수를 최소화시키는 수정된 방위각 이노베이션 벡터는 다음 식과 같다.

$$\delta\theta = [B^H B + L]^{-1} B^H y \quad (23)$$

단계 3. 쇄신

수정된 방위각 이노베이션 ($\delta\theta_i(k+1), i=1,2,\dots,N$)을 이용하여 i 번째 표적의 상태벡터와 이것의 공분산행렬을 식 (24)와 같이 쇄신한다.

$$\hat{x}_i(k+1 | k+1) = \hat{x}_i(k+1 | k) + G_i(k+1)\delta\theta_i \quad (24)$$

$$P_i(k+1 | k+1) = [I - G_i(k+1)H]P_i(k+1 | k)$$

여기서 $G_i(k+1)$ 은 칼만이득행렬이며 다음 식을 이용하여 계산한다.

$$G_i(k+1) = P_i(k+1 | k)H^T [HP_i(k+1 | k)H^T + \sigma_{\theta_i}^2]^{-1} \quad (25)$$

쇄신된 각 표적의 상태벡터에서 첫번째 항이 시간 $k+1$ 에서 각 표적의 방위각 추정치가 된다.

IV. 시뮬레이션 및 결과

제안된 알고리즘의 성능을 서론에서 언급한 다른 알고리즘들과 비교 평가하기 위하여 두 표적이 교차하는 상황하에서 시뮬레이션을 하였다. 그림 2와 같이 두 표적은 선배열센서에 평행하

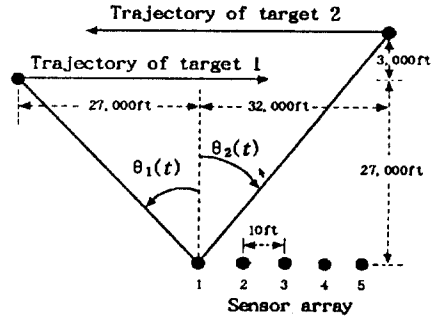


그림 2. 시뮬레이션을 위한 표적과 배열센서의 기하학적 구조. Fig. 2. Two crossing target simulation geometry.

게 서로 반대방향으로 움직인다. 표적의 속도는 각각 1500 ft/s이며, 방위각 궤적들은 약 20초에 서로 교차한다. 선배열센서는 5개의 센서가 10 ft의 등간격으로 배열되어 있다. 각 센서잡음크기 σ^2 은 1이라 가정하였다. 시간 k 에서 센서 출력공분산행렬을 다음 식과 같이 추정한다.

$$\hat{R}(k) = \frac{1}{N_s} \sum_{j=0}^{N_s} r((k-1)T + \frac{jT}{N_s}) r^H((k-1)T + \frac{jT}{N_s}) \quad (26)$$

N_s 는 샘플링 주기 $[(k-1)T, kT]$ 동안의 단편(snapshot)의 개수이며 200으로 정하였고, 샘플링 주기 T 는 1초로 정하였다.

프로세스잡음의 공분산행렬 Q_i 와 측정잡음의 분산 $\sigma_{\theta_i}^2$ 를 다음 식과 같이 정하였다.

$$Q_i = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} a, \quad a = 1^\circ \quad (27)$$

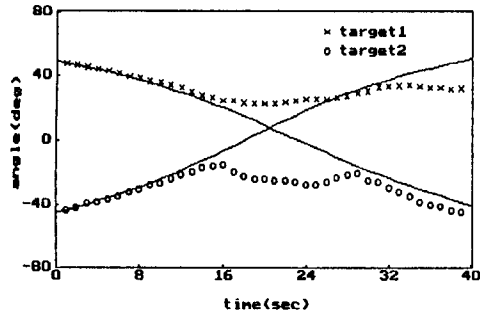
$$\sigma_{\theta_i}^2 = 9^\circ, \quad i=1,2$$

식 (23)의 방위각 이노베이션의 크기를 제한하는 가중행렬을 다음 식과 같이 정하였다.

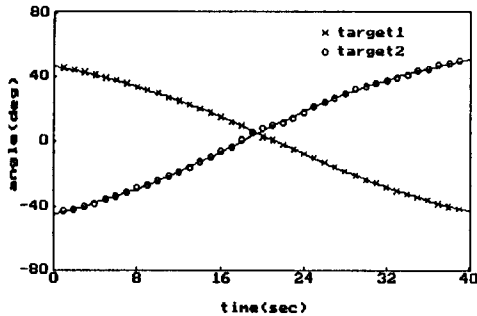
$$l_i = \frac{1}{20} \times (\text{행렬 } B^H B \text{의 } i\text{번째 대각 항}) \quad (28)$$

표적들의 신호대 잡음비 $\text{SNR} = 10 \log(S_{ii}/\sigma^2)$ 이 각각 14 dB와 12 dB인 경우, 대표적인 방위각 추적 결과는 그림 3과 같다. 그림 3(a)는 Sword 알고리즘을 이용한 것이고 그림 3(b)는 제안된 알고리즘을 이용한 것이다. 실선은 참 궤적(real track)이며, 점선은 추적한 궤적이다. Sword 알고리즘에서는 추정 오차가 시간에 따라 점점 커져 추적이 실패하였으나, 제안한 알고리즘에서는 추정 오차가 매우 작을뿐만 아니라 궤적이 교차할 때에도 성공적으로 추적하였다.

다른 알고리즘들과 성능을 비교하기 위하여 3 가지 다른 상황에서 100번의 몬테카를로 시뮬레이션을 하였다. 첫번째 상황은 두 표적의 신호크기가 $S_{11} = 25(14 \text{ dB})$ 과 $S_{22} = 16(12 \text{ dB})$ 로 상



(a) Sword 알고리즘
(a) Sword's algorithm



(b) 제안된 알고리즘
(b) proposed algorithm

그림 3. 교차하는 방위각 궤적에 대한 대표적인 추적 예.
Fig. 3. A typical sample run for crossing tracks.
(SNR : 14 dB for target 1 and 12 dB for target 2).

당히 다른 경우이며, 이 때의 추적 성공률은 표 1과 같다. Sword 알고리즘과 Lo 알고리즘은 매우 낮은 성공률을 보였으나, 제안한 알고리즘은 96%의 높은 성공률을 보였다. Rao 알고리즘은 88%의 성공률을 보여, 성능이 제안된 알고리즘과 비슷하다. 이는 표적의 신호크기가 상당히 차이가 나기 때문이다. 두번째 상황은 두 표적의 신호크기가 $S_{11}=25(14 \text{ dB})$ 와 $S_{22}=20(13 \text{ dB})$ 로 서로 유사한 경우이며, 이 때의 추적 성공률은 표 2와 같다. 제안한 알고리즘의 성능은 여전히 좋지만 다른 알고리즘들의 성능은 나쁘다. 특히 Rao 알고리즘의 성능이 급격히 떨어짐을 볼 수 있다. 이것은 Rao 알고리즘의 성능이 표적의 신호크기의 차이에 의존적임을 나타낸다. 세번째 상황은 $S_{11}=S_{22}=1(0 \text{ dB})$ 인 경우이며 두 신호크기가 달라야 한다는 Rao 알고리즘의 가정에 위배되는 상황이며, 시뮬레이션 결과가 표 3과 같다. 제안된 알고리즘은 계속 만족할만한 성능을 보여주고 있다.

제안된 알고리즘을 두 표적이 기동(maneuver)하는 경우에 적용하였다. 그림 4는 그 대표적인 추적 결과이다. 두 궤적의 최소 각차(minimum angular distance)가 7° 인 경우, 시뮬레이션 결과는 91%의 성공률을 보였다.

표 1. 추적 성공률.

Table 1. Success rate.

(SNR : 14 dB for target 1 and 12 dB for target 2).

Target	Algorithm			
	Proposed	Sword's	Lo's	Rao's
1	96	20	25	88
2	96	20	26	88

표 2. 추적 성공률.

Table 2. Success rate.

(SNR : 14 dB for target 1 and 13 dB for target 2).

Target	Algorithm			
	Proposed	Sword's	Lo's	Rao's
1	92	23	30	11
2	92	23	29	11

표 3. 추적 성공률.

Table 3. Success rate.

(SNR : 0 dB for target 1 and 0 dB for target 2).

Target	Algorithm			
	Proposed	Sword's	Lo's	Rao's
1	88	20	20	*
2	88	19	20	*

* 두 방위각 궤적을 왔다 갔다하며 추적한다.

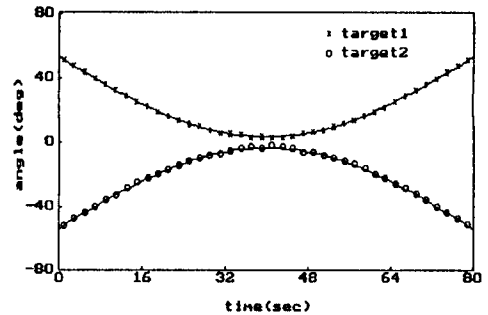


그림 4. 기동하는 방위각 궤적에 대한 대표적인 추적 예.
Fig. 4. A typical sample run for maneuvering tracks.
(SNR : 14 dB for target 1 and 12 dB for target 2).

V. 결론

본 논문에서 간단한 구조의 칼만 필터를 도입하여 Sword 알고리즘을 수정하였다. 현재의 방위각 추정치 대신에 방위각 예측치를 사용하여 추정 오차를 줄일 수 있었다. 또한 가격함수에 방위각 이노베이션의 크기를 제한하는 항을 추가하였다. 이 가격함수를 최소화하는 방위각 이노베이션을 이용하여 표적이 교차하거나 근접할 때 추적 성능을 매우 향상시켰다. 다양한 컴퓨터 시뮬레이션을 통하여 이를 확인하였다.

참 고 문 헌

- [1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.
- [2] S. S. Blackman, *Multiple-Target Tracking with Radar Application*, Artech House, 1986.
- [3] K. C. Chang and Y. Bar-Shalom, "Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers," *IEEE Transactions on Automatic Control*, vol. AC-29, pp. 585-594, July 1984.
- [4] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. OE-8, pp. 173-184, July 1983.
- [5] G. C. Carter, "Coherence and time delay estimation," *Proceedings of the IEEE*, vol. 75, pp. 236-255, Feb. 1987.
- [6] C. K. Sword, M. Samaan, and E. W. Kamen, "Multiple target angle tracking using sensor array outputs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-26, pp. 367-373, Mar. 1990.
- [7] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. AP-34, pp. 276-280, Mar. 1986.
- [8] K. W. Lo and C. K. Li, "An improved multiple target angle tracking algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-28, pp. 797-805, July 1992.
- [9] C. R. Rao, L. Zhang, and L. C. Zhao, "Multiple target angle tracking using sensor array outputs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-29, pp. 268-271, Jan. 1993.