

TMS320C25 DSP를 이용한 실시간 TWS 시스템 구현

(On the Real Time Implementation of the TWS System Using the TMS320C25 DSP)

奇 錫 哲,** 李 商 郁*

(Seok Cheol Kee and Sang Uk Lee)

要 約

본 논문에서는 고속 신호처리 프로세서(digital signal processor)인 TMS320C25를 사용하여 칼만 필터링 기법을 이용한 실시간 TWS(track-while-scan) 시스템의 구현에 대하여 고찰하였다. 먼저 고정 소숫점 연산에 의해 칼만 필터를 구현할 때 생기는 FWL(finite word length)의 영향에 대하여 알아 보았다. 실시간 TWS 시스템은 TWS연산부, 스캔 컨버터(scan converter) 그리고 시스템 제어부로 구성하였고 시스템 버스는 multi-bus를 채택하였다. TWS 시스템은 최대 8개의 표적을 동시 추적하기 위하여 제작되었으며, 실험을 통하여 8개의 표적을 동시에 추적하는데 부동 소숫점 연산시 약 0.35sec, 고정 소숫점 연산시 약 0.28sec의 시간이 소요된다는 결과를 보임으로써 칼만 필터를 실시간으로 처리할 수 있는 충분한 가능성을 제시하였다.

Abstract

In this paper, a real-time implementation of the TWS(track-while-scan) system using the high-speed DSP(digital signal processor) TMS320C25 is described. First, attempts have been made to investigate the FWL(finite word length) effect, which is caused by employing a fixed point arithmetic, of implementing the Kalman filter. The real-time TWS system consists of TWS arithmetic unit, scan converter, and system controller. In addition, the TWS system is integrated in the Multi-Bus. In experiment, it is observed that by employing the floating point arithmetic the computation time of 0.35sec is required for tracking 8 targets simultaneously, while 0.28sec is required for the fixed point arithmetic. Since the TWS system is designed to track up to 8 targets simultaneously, we conclude that the system is enough to process Kalman filter in a real-time.

I. 서 론

TWS(track-while-scan)는 대공 또는 대함용 탐지 레이더에 제한된 표적의 추적 기능을 부가시키는 알고리즘으로써, 군에서는 기본적인 대공 및 대함 방어 체계에, 상용으로는 공항 관제탑에서 항공기의 항로

*正會員, **準會員, 서울大學校 制御計測工學科
(Dept. of Control and Instrumentation Eng., Seoul Nat'l Univ.)

接受日字: 1989年 4月 12日

유도등에 경제적인 시스템으로 널리 사용되고 있다.

1960년대 초 Kalman¹⁴⁾이 정적 확률 프로세스(stationary stochastic process)에 대한 Wiener 필터 개념을 비정적(nonstationary)인 경우에 대해서 확장시키면서 표적의 추적 알고리즘에 사용하게 되었다. 칼만 필터는 구조가 선형적이며 수렴성이 어느 정도 보장될 뿐만 아니라 종전의 주파수 영역분석 방법에서 벗어난 시간 영역에서의 처리 기법이므로 하드웨어에 의한 실시간 처리가 가능한 기법이다.

한편 표적에 대한 좌표계의 선택은 추적 알고리즘의 선택과 측정 시스템의 선택에 크게 좌우된다고 할 수 있다. 본 논문에서 가정한 측정 시스템은 극 좌표계를 사용하는 탐지용 레이이다. 이럴 경우 일반 선형 칼만 필터를 적용하기 위해서는 목표물의 상태 방정식을 선형화해야 할 필요가 있으나 좌표 변환 과정은 필요없게 된다. 이때 일반적으로 선형화로 인한 동적 오차는 직교 좌표계를 사용하였을 경우 보다 훨씬 크고, 계산량의 부담도 커지게 된다.¹⁵⁾ 따라서 본 논문에서 사용할 추적 기법은 직교 좌표계를 사용하게 되고, 레이더 측정 시스템의 측정 방정식이 극 좌표계로 표현되므로 좌표 변환 과정이 필요하게 된다.

한편 최근에 급속히 발달하기 시작한 반도체 기술에 의해 곱셈과 같은 연산을 빠르게 수행할 수 있는 DSP(digital signal processor)가 소개된 이후 많은 관심이 집중되었다. 칼만 필터와 같이 행렬의 곱셈을 반복하게 되는 경우에 DSP를 사용하면 실시간 구현에 매우 잇점이 있다. 본 논문에서는 DSP의 일종인 TMS320C25¹⁶⁾를 사용하여 추적 시스템의 연산을 담당하게 하였다. 그러나 TMS320C25는 고정 소숫점(fixed-point) 연산을 하는 프로세서이므로 FWL(finite word length)의 영향에 의해 추적 성능이 저하될 수 밖에 없다. 그러므로 본 논문에서는 FWL 영향에 의한 칼만 필터 출력 오차의 통계학적 특성을 분석하여 실제 하드웨어 구현시 예상되는 성능 저하에 대해 고찰 하였다.

본 논문에서는 먼저 DSP와 같이 연산을 빠르게 수행할 수 있는 시스템의 구현을 위하여 고정 소숫점 및 부동 소숫점(floating-point) 연산으로 구현하는 방법과 그 실험 결과를 제시하여 실시간 처리가 가능함을 보였다. 또한 TMS320C25를 사용하여 최대 8 표적까지 동시 추적할 수 있는 TWS 연산부와 추적 상황을 화면에 표시하기 위한 스캔컨버터, 그리고 시스템 버스인 multi-bus를 제어하기 위한 시스템 제어부로 구성된 실시간 TWS 시스템의 설계 및 구현, 그리고 실험결과에 대하여 논하였다.

II. 칼만 필터 구현을 위한 고정 소숫점 연산고찰

디지털 필터를 범용 계산기나 특별한 목적을 가진 하드웨어로 구현함에 있어서 계산기의 FWL의 영향은 매우 중요한 문제이다. 본 논문에서는 곱셈 후 반올림 오차가 디지털 칼만 필터의 출력에 미치는 영향을 분석하여 보았다. 일반적으로 디지털 필터에서 생길 수 있는 오차원은 필터계수의 양자화, 입력 데이터의 A/D 변환, 곱셈 후 반올림 등이 있다.¹⁷⁾ 그런데 여러가지 오차원 중에서 특히 곱셈 후 반올림 오차가 중요한 이유는 다른 오차원과는 달리 이 오차가 고정 소숫점 연산에서 가장 큰 영향을 미치기 때문이다. 따라서 곱셈 후 반올림에 의한 필터 출력 오차의 분산(σ_o^2)을 구하기 위하여 다음과 같은 가정을 하였다.

- 양자화는 2's complement 반올림을 사용하였다.
- 각 신호와 오차원은 서로 독립(independent)이다.
- 고정 소숫점 연산시 모든 데이터 및 계수의 길이는 b비트로 동일하다.
- 곱셈 후 반올림 오차는 영 평균이고, 분산이 $2^{-2b}/12$ 인 백색 가우시안 잡음 프로세스이다.

1) σ_o^2 의 유도

일반적인 표적의 상태 방정식과 측정 방정식은 다음과 같다.

$$X(k+1) = F X(k) + G W(k) \quad (1a)$$

$$Z(k) = H X(k) + V(k) \quad (1b)$$

여기서 $X(k)$ 는 표적의 동특성에 관련된 상태 변수이고 $W(k)$, $V(k)$ 는 영평균 백색 가우시안이다. 이와같은 시스템 모델이 주어진다면 이산형 칼만 필터는 다음과 같다.¹⁸⁾

$$\begin{aligned} \hat{X}(k+1) &= F \hat{X}(k) + K(k) [Z(k) - H F \hat{X}(k)] \\ &= [F - K(k) H F] \hat{X}(k) + K(k) Z(k) \\ &= \Phi(k) \hat{X}(k) + K(k) Z(k) \end{aligned} \quad (2)$$

여기서 $\hat{X}(k)$ 는 칼만 필터의 상태 변수로써, 실질적으로는 1step 예측된 표적의 동특성에 관한 변수이다. 그리고 이산형 칼만 필터의 이득행렬(gain matrix)과 공분산 행렬(covariance matrix)은 다음과 같이 주어지며, (')는 전치행렬(transpose matrix)이다.¹⁸⁾

$$K(k) = P(k) H' [H P(k) H' + R]^{-1} \quad (3a)$$

$$P(k) = F [I - K(k-1) H] P(k-1) F' + G Q G' \quad (3b)$$

$$\Phi(k) = [I - K(k) H] F \quad (3c)$$

이산형 칼만 필터의 이득 행렬은 어느 정도의 시간이 지난 이후 수렴을 하게 되므로 그 이후 안정한 상태에서는 시간 영역에서 불변하게 되어 상수 행렬로 생각할 수 있다. 실제 시뮬레이션 결과 6~8번의 반복 후에는 이득 행렬이 수렴함을 알 수 있었다. 본 논문에서는 칼만 필터의 이득 행렬은 부동 소숫점 연산을 하는 계산기를 이용하여 미리 계산하여 구하였다. 이는 식 (1b)의 $Z(k)$ 가 n 차원이라고 가정할 경우 $n \times n$ 역행렬을 구해야 하는데 이것을 고정 소숫점 연산에 의해 계산하기는 매우 어렵고 그 정확도 또한 의문시 되기 때문이다.⁹⁾ 그래서 식(2)는 시간에 대한 변수가 없는 상수 계수의 식으로 다음과 같이 변형이 가능하다.

$$\hat{X}(k+1) = \Phi \hat{X}(k) + K(k) Z(k) \quad (4)$$

여기서 FWL에 의해서 구현된 실제 필터의 입력 계수의 양자화와 연산후 반올림을 나타내면 다음과 같다.

$$\tilde{X}(k+1) = [(\Phi + \Delta\Phi) \tilde{X}(k)]_r + [(K + \Delta K) \tilde{Z}(k)]_r \quad (5a)$$

$$\tilde{X}(k+1) = \Phi \tilde{X}(k) + \alpha(k) + \tilde{K}(k) \tilde{Z}(k) + \beta(k) \quad (5b)$$

식(5)에서 (\sim)는 실제입력, 상태변수, 필터계수를 나타내며 [$]_r$ 은 곱셈 후 반올림을 나타내고 $\alpha(k)$, $\beta(k)$ 는 곱셈 후 반올림에 의한 오차를 나타내는 불규칙 프로세스이다. 또한 (Δ)는 FWL에 의한 필터 계수, 입력, 상태 변수의 오차를 의미한다. 그러므로 다음과 같은 관계를 쉽게 알 수 있다.

$$\tilde{X}(k) = \hat{X}(k) + \Delta X(k) \quad (6a)$$

$$\tilde{Z}(k) = Z(k) + \Delta Z(k) \quad (6b)$$

$$\tilde{K} = K + \Delta K \quad (6c)$$

$$\tilde{\Phi} = \Phi + \Delta\Phi \quad (6d)$$

식 (5b)에서 식 (4)를 빼면 다음과 같다.

$$\Delta X(k+1) = \tilde{\Phi} \Delta X(k) + \tilde{K} \Delta Z(k) + \Delta\Phi \hat{X}(k) + \Delta K Z(k) + \alpha(k) + \beta(k) \quad (7)$$

$\Delta X(k)$ 를 상태 오차 벡터로 정의하고 상태 오차 벡터의 해를 식(7)을 이용하여 다음과 같이 구할 수 있다.

$$\begin{aligned} \Delta X(k+1) &= (\tilde{\Phi})^{k+1} \Delta X(0) + \sum_{j=0}^k (\tilde{\Phi})^{k-j} [\tilde{K} \Delta Z(j) \\ &\quad + \Delta\Phi \hat{X}(j) + \Delta K Z(j) + \alpha(j) + \beta(j)] \end{aligned} \quad (8)$$

식(8)에서 $\Delta X(0) = 0$ 으로 가정하면 식(8)은 세가지 항으로 다음과 같이 분류된다.

$$\Delta X(k+1) = \Delta X_1(k+1) + \Delta X_2(k+1) + \Delta X_3(k+1) \quad (9)$$

$$\Delta X_1(k+1) = \sum_{j=0}^k (\tilde{\Phi})^{k-j} [\tilde{K} \Delta Z(j)] \quad (10a)$$

$$\Delta X_2(k+1) = \sum_{j=0}^k (\tilde{\Phi})^{k-j} [\Delta\Phi \hat{X}(j) + \Delta K Z(j)] \quad (10b)$$

$$\Delta X_3(k+1) = \sum_{j=0}^k (\tilde{\Phi})^{k-j} [\alpha(k) + \beta(k)] \quad (10c)$$

여기서 식 (10a)는 입력 양자화 오차이며, 식(10b)는 계수의 양자화 오차이고, 식 (10c)는 곱셈 후 반올림 오차에 해당하는 것이다. 곱셈 후 반올림 오차 σ_r^2 은 식 (10c)에 의하여 다음과 같이 유도할 수 있다. 여기서의 유도 과정은 부록에 자세히 나타내었다.

$$\sigma_r^2 = E[\Delta X_3^2(k+1)] = \sum_{j=0}^k (\tilde{\Phi})^j Q (\tilde{\Phi}^j)^T (E_r^2/12) \quad (11)$$

식(11)에서 ($E_r^2/12$)은 uniform 확률 밀도 함수를 갖는 반올림 오차나 절삭 오차의 분산을 나타내며 Q 는 대각(diagonal) 행렬로서 대각 행렬의 각 원소는 Φ 와 K 의 i 번째 열에서 0 이나 1 이 아닌 원소의 갯수가 된다. 다시 말하면

$$Q = \text{diag}(q_1, \dots, q_n) \quad (12)$$

여기서 q_i 는 Φ 와 K 의 i 번째열의 실수의 갯수이다. 그런데 식(11)은 Cauchy's limit theorem¹⁰⁾에 의해서 다음과 같이 쓸 수 있다.

$$\sigma_r^2 = \sum_{j=0}^{\infty} (\tilde{\Phi})^j Q (\tilde{\Phi}^j)^T (E_r^2/12) \quad (13)$$

한편 식(13)이 수렴하기 위한 필요조건은 다음과 같다.

$$\lim_{j \rightarrow \infty} (\tilde{\Phi})^j = 0 \quad (14)$$

식(14)은 Φ 의 특이값이 단위원 내에 존재해야 수렴함을 나타낸다. 다음은 상태변수의 크기를 스케일링했을 때의 관계를 알아보자. 스케일링은 overflow가 생기지 않도록 상태변수의 값을 normalize할때 필요한 것으로 상태변수의 각 원소를 적당한 값으로 나

누어 주는 것이다. 대각 행렬인 스케일링 행렬을 식 (15)과 같이 정의하자.

$$S = \begin{bmatrix} s_1, \dots, 0 \\ \vdots \\ 0, \dots, s_n \end{bmatrix} \quad (15)$$

S 행렬을 이용하여 상태변수의 크기를 스케일링하면

$$X^s(k) = S^{-1} X(k) \quad (16)$$

과 같다. 이것을 이용하여 앞에서 곱셈 후 반올림 오차의 분산을 유도하는 과정을 되풀이 하면 상태변수의 스케일링을 포함한 $E(\Delta X_s^2)$ 을 다음과 같이 유도할 수 있다.¹¹⁾

$$E(\Delta X_s^2) = \sum_{j=0}^{\infty} (\tilde{\Phi}^j)' S Q S' (\tilde{\Phi}^j)' (E_s^2/12) \quad (17)$$

2) 시뮬레이션 결과

본 절에서는 앞에서 제시한 필터 출력 오차식을 컴퓨터 시뮬레이션 결과와 비교 검토하였다. 본 논문에서 가정하고 있는 무한 정밀도(infinite precision) 연산은 double-precision 부동 소숫점 연산으로 처리하였으며, 고정 소숫점 연산은 부동 소숫점 연산과는 별도로 프로그래밍한 후 비트 길이에 따른 연산을 각각 수행하게 되며 이때 최대 비트 크기는 16비트로 제한하였다.

고정 소숫점 연산과 부동 소숫점 연산을 비교시 초기값 및 입력을 동일한 조건하에서 수행하였다. 오차의 분산은 칼만 필터가 안정화되는 10step 이후에서 매 step마다 500개의 앙상블 평균(ensemble average)을 구했으며, 다시 20개의 시간 평균(time average)을 가지고 오차의 통계를 구하였다. 고정 소숫점 연산의 시뮬레이션은 다음과 같은 3 가지 시스템을 가정하여 수행하였다.

- 스케일링이 필요없는 시스템 모델⁹⁾
- 상속도 직선 운동의 시스템 모델
- 가속도 운동의 시스템 모델

스케일 크기의 변화와 출력 오차와의 관계를 보기 위하여 입력의 스케일링이 전혀 필요없는 시스템1)을 정의하였다. 시스템2)와 시스템3)은 실제 목표물의 시나리오를 표현한 것으로 스케일링이 필요한 모델이다.

다음은 앞에서 유도한 출력 오차의 분산과 실제 시스템에서의 결과를 표 1과 표 2에서 제시하였다. 각각의 표에서 계산된 값은 식(13)과 식(17)에 의한 것이며 표1은 입력 크기의 값을 조정할 필요가 없는 시

표 1. 시스템 1)의 출력오차 분산식과 실제 시스템의 비교

Table 1. Comparison of system error variance and output error variance equation of system1).

# of bits	실험치	계산치
16	8.52 E-9	6.42 E-9
14	2.44 E-7	1.03 E-7
12	5.05 E-6	1.65 E-6
10	4.12 E-5	2.64 E-5
8	1.06 E-3	4.23 E-4
6	6.90 E-3	6.73 E-3

표 2. 시스템 2)의 출력오차 분산식과 실제 시스템의 비교

Table 2. Comparison of system error variance and output error variance equation of system 2).

# of bits	실험치	계산치
16	1.18 E-6	1.26 E-7
14	9.60 E-5	4.36 E-6
12	7.43 E-4	1.51 E-4
10	3.93 E-1	5.25 E-3
8	7.22 E+1	1.82 E-1
6	7.22 E+1	6.31 E+0

스템1)의 경우이고, 표 2는 시스템2)의 경우이다. 시스템3)의 경우는 출력 오차가 수렴할 조건식 (14)를 만족하지 못하는 경우이므로 계산값을 구할 수가 없게 된다. 제시된 표들은 유도된 출력 오차의 분산값과 실제 시스템의 실험치를 비교하여 유도된 식의 타당성을 보이기 위한 것이다. 실제 시스템에는 곱셈 후 반올림 오차만이 존재하는 것이 아니라 그 이외의 양자화 오차도 존재하게 되므로 그 오차 분산값이 유도식의 값보다 크게 나타났다. 또한 표1,2를 비교하여 보면 스케일링에 의한 영향이 나타난다. 표2와 같이 스케일링이 필요한 경우 식(17)에서 Q의 대각 원소의 값이 표1과 같이 스케일링이 필요없는 경우의 Q보다 크게 되므로 유도식의 값은 표1의 경우가 표2의 경우보다 크게 된다. 한편 실제 표적의 시나리오를 적용한 표2에서 14,16 비트를 사용하였을 때 FWL의 영향이 무시할 수 있을 정도로 작아서 고정 소숫점 연산으로 필터를 구현할 수 있게 된다. 가속도 운동의 표적에 대해서는 표2의 결과보다 더욱

큰 스케일링값이 필요하여 실험결과 16비트 이상의 고정 소숫점 연산에서만 추적이 가능함을 알 수 있었다.

Ⅲ. TWS 시스템의 하드웨어 설계 및 구현

1. 하드웨어의 구성

일반적인 신호처리 시스템은 많은 연산을 빠른 시간내에 처리할 수 있어야 한다. 또한 시스템 유연성 (flexibility)이 매우 중요한데, 이를 위해서는 될 수 있는한 많은 작업을 소프트웨어로 처리하여 소프트웨어의 변경만으로 시스템의 성능을 개선할 수 있어야 한다.

TMS320C25를 이용하여 일반적인 신호 처리 시스템을 설계할 경우에 기본적인 하드웨어 구성은 모두 비슷하며, 필요에 따라 일부 하드웨어만 수정해서 다른 신호처리 시스템에 이용할 수 있는 것이 큰 장점이라고 할 수 있다. 이와같은 DSP의 장점을 이용하여 추적 필터의 연산부를 구현하였다. 한편 추적 시스템의 출력 장치의 목적을 위하여 스캔 컨버터(scan converter)를 제작하였다. 이것은 CRT-TV 모니터의 임의의 위치에 원하는 도형이나 문자를 출력하는 기능을 담당한다. 여기서는 graphic display controller 인 NEC7220을 사용하였다. 스캔 컨버터의 또 다른 기능은 X-Y 좌표 변환을 하드웨어적으로 처리하는 것이다. 레이더의 입력 데이터는 극좌표계이므로 CRT 모니터의 입력형태에 맞는 X-Y 직교 좌표계로 변환할 필요가 있는 것이다. 한편 시스템 버스는 multi-bus를 채택하였다. Multi-bus는 1982년 IEEE에서 표준화 버스 규격으로 제공된 IEEE standard 796 버스이다.¹³⁾ 시스템 버스는 일반적인 버스 시스템의 기초로서 시스템의 유연성 (flexibility), 신뢰성 (reliability) 또는 성능과 비용에 큰 영향을 주게 된다. 이렇게 구성된 하드웨어는 동시에 최대 8개의 표적을 실시간으로 추적하는데 목적이 있다. 전체 시스템의 블록 선도는 그림 1과 같고, 그림 2는 실제 제작된 TWS 시스템의 실물 사진이다. 그러면 각 부분에 대한 설명은 다음과 같다.

1) TWS 연산부

TWS 연산부는 DSP인 TMS320C25를 이용하여 추적 알고리즘의 연산을 수행한다. TWS 연산부의 하드웨어의 주요 부분은 아래와 같이 크게 3부분으로 그 기능을 나눌 수 있다.

- TMS320C25
- 레이더의 데이터 입력부
- 메모리부

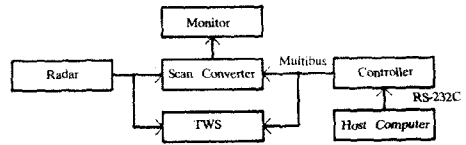


그림 1. TWS 시스템의 블록 선도
Fig. 1. Block diagram of TWS system.



그림 2. TWS 시스템의 실물 사진
Fig. 2. Photograph of TWS system.

TMS320C25를 제외한 이들 각 부분에 대한 설명은 다음과 같다.

○ 레이더의 데이터 입력부

레이더의 입력 데이터는 아날로그 신호를 8MHz로 샘플링한 4비트 데이터이다. 본 논문에서 가정한 레이더는 1800Hz의 PRF(pulse repetition frequency)를 갖는 탐지용 레이더이다. 샘플링된 4비트 데이터는 DPRAM에 입력되는데 여기서는 두개의 DPRAM을 사용하여 번갈아 가며 레이더의 데이터를 저장하게 하였다. 이때 TMS320C25는 소프트웨어로 주사 거리에 해당하는 윈도우를 적용하여 윈도우 내에서 표적 측정값을 얻게 된다. 표적의 측정값을 얻기 위하여 윈도우 방식을 도입한 이유는 쓸모없는 입력 데이터를 취하여 모두 처리함으로써 생기는 시스템의 비효율성을 최소화하기 위한 것이다.

○ 메모리부

TMS320C25의 내부 4K ROM은 masked ROM이므로 시스템 개발을 위하여 사용할 수 없다. 그러므로 외부에 EPROM을 사용하는데 4K×8비트의 고속

ROM을 2개 사용하여 프로그램을 수행시킨다. 고속 ROM은 WS57C49B로 access 시간은 35ns이다. 최대 8개의 윈도우 데이터와 프로그램 수행을 위한 데이터를 저장해야 하는 RAM도 역시 고속 RAM을 사용하였다. KM64256은 access 시간 25ns, 용량 64K × 4비트 크기이며, 16비트의 데이터 버스에 연결해야 하므로 4개를 사용하였다.¹⁴⁾

2) 스캔 컨버터 (scan converter)

시스템의 추적상태를 표시하기 위한 스캔 컨버터의 기본적인 기능은 크게 2부분으로 나뉘어 진다.

○X-Y 어드레스 발생부

○Graphic display 부

각 부분에 대한 자세한 설명은 다음과 같다.

○X-Y 어드레스 발생부

레이다에서 발생하는 표적의 위치는 레이다의 주사방식과 같은 (r, θ) 의 극 좌표계로 표시된다. 즉 일정한 방위각 θ 에 대하여 거리 r 이 계속 증가하며 주사하고 일정한 거리에 도달하면 PRF가 발생하여 다시 θ 가 다음값으로 증가하고 r 이 0부터 다시 증가한다. 이 때 PRF는 1800Hz로 일정하다. 이러한 (r, θ) 의 극 좌표계로 나타나는 신호를 CRT 모니터 입력 형태에 맞는 (X, Y) 직교 좌표계로 변환시킬 필요가 있다. X-Y 어드레스 발생기는 일정한 θ 값에서 r 이 계속 증가하면 각각의 r 에 대하여 그것에 해당하는 X-Y 어드레스를 발생시키는 작용을 한다. 14비트를 사용하여 0~360도를 θ 로 표시하고 매 클럭마다 일정한 Δr 값을 더하면서 거리값을 증가시켜 나간다. Y 어드레스의 경우도 마찬가지로 하면된다.

○Graphic display 부

CRT 모니터상의 임의의 위치에 원하는 모양의 도형이나 문자를 알맞는 크기로 display 해볼 필요가 있을 것이다. GDC(graphic display controller)인 NEC 7220은 위와 같은 기능을 하는 소자이다. 이 부분은 스캔 컨버터 중에서 80186 제어부에 의해서 access가 가능한 부분이다. 80186 제어부는 NEC7220을 초기화시키고 원하는 도형이나 문자를 모니터에 표시하도록 한다. NEC7220의 RAM의 일정한 어드레스는 모니터의 일정한 위치에 해당하고 데이터는 그 위치상의 밝기값을 나타낸다. NEC7220은 일반적으로 두가지 기능을 한다. 그중 하나는 이미 RAM에 들어 있는 내용을 모니터상에 나타내는 것과 다른 하나의 기능은 이미 RAM에 들어 있는 내용을 수정하여 새로운 데이터를 RAM에 써넣는 것이다.

3) 시스템 제어부

시스템 제어부는 TWS 시스템의 전체 동작을 주관하는 제어부로서 80186 CPU를 이용한 시스템을 구

성하였다. 시스템 제어부의 주요기능은 다음과 같다.

○Host 컴퓨터로부터 직렬 port를 이용하여 시스템 운용 프로그램의 downloading을 한다.

○NEC7220을 비롯한 전체 시스템의 초기화

○TWS 연산부의 결과를 받아 스캔 컨버터에 보내고 NEC7220을 제어하여 도형이나 문자 출력을 담당한다.

○시스템 버스인 multi-bus를 주관한다.

시스템 제어부의 시스템 프로그램의 내용은 그림 3의 흐름도와 같다.

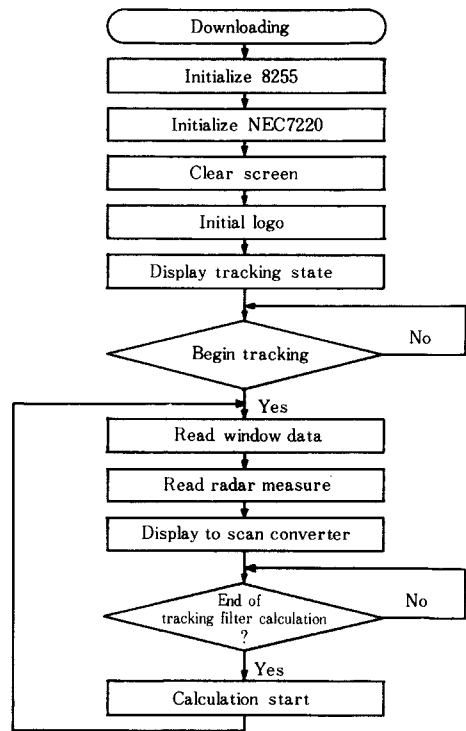


그림 3. 시스템 프로그램의 흐름도
Fig. 3. Flow chart of system program.

IV. 실험 결과

실험은 TWS 연산부의 고속 ROM에 추적 필터 계산과 데이터의 입, 출력에 관한 프로그램을 TMS320 C25 어셈블리어로 작성하여 수행하였다. 시스템 제어부가 전체 시스템을 초기화하여 실시간 추적 시스템을 동작시키기 위한 프로그램은 IBM PC/XT host 컴퓨터에서 RAM에 downloading하여 수행하였다.

실험에 사용한 표적의 시나리오는 X축 방향으로의 상속도 직선운동을 하는 것으로 초기위치(1000m, 1000 m)에서 X축 방향으로 -10m/sec의 제적을 갖게 하였으며, 이는 식(1)의 상태방정,을 따른다. 그러므로 실험에 사용한 표적의 데이터는 영 평균 백색 가우시안 잡음이 레이다에서 더해진 것으로 가정하여 실제의 제적에 잡음이 섞인 측정값을 추적 필터의 입력으로 사용하였다. 추적 필터의 초기값은 실제 제적과 똑같은 값을 주게 된다. 즉 레이다에서 표적이 발견되면 표적에 대한 위치 정보를 초기값으로 하여 추적기를 동작시키게 되는 것이다. 33rpm의 레이다를 측정시스템으로 가정하여 샘플링 시간은 1.8182s가 된다.

고정 소숫점 연산의 칼만 필터는 연산을 빨리 수행할 수 있으나 FWL의 영향 때문에 추적 성능이 저하되고 때로는 필터 발산의 원인이 되기도 한다. 반면 부동 소숫점 연산의 칼만 필터는 FWL의 영향이 매우 작아 추적 성능은 좋으나 연산 수행 시간이 길어서 실시간 처리에 불리하다는 단점을 갖고 있는 것으로 알려져 있다. 그러나 TMS320C25는 고속의 프로세서이기 때문에 실시간 조건을 허용하는 범위내에서 부동 소숫점 연산을 수행할 수 있다. 실험을 통하여 각 연산을 TMS320C25로 수행하는데 걸리는 계산시간은 덧셈의 경우 15.4μsec, 곱셈의 경우 7.8μsec, 나눗셈의 경우 22.8μsec이다.

실험에 의한 추적 결과를 표 3에 제시하였다. 표 3은 실제 표적의 시나리오와 예측된 윈도우의 중심좌표값을 직교 좌표계로 나타낸 것이다. 매 샘플링마다 X, Y 좌표값의 순서대로 나타내었고, 각 좌표값은 부동 소숫점 연산을 위한 데이터 형식을 따라 나타내었다.

실험 결과에서 추적 필터가 초기의 과도응답(transient response) 없이 잘 추적하는 것은 추적 필터의 초기 상태 변수를 실제 시스템과 똑같이 사용했기 때문이다.

표 4, 5는 전체 시스템의 실시간 처리능력을 보이기 위한 명령어수와 이에 따르는 계산 소요시간을 보이고 있다. 하나의 표적에 대해서 한번의 연산을 수행하는데 필요한 명령어수와 소요시간들이다. 표 5는 표 4의 필터 수행부분을 각 연산법에 따라 자세하게 나타낸 것이다.

추적 목표물의 측정값을 얻기 위해서는 64PRF 동안의 DPRAM의 데이터를 읽어야 하므로 그 수행 소요시간은 레이다의 PRF에 의해 결정된다. 추적 목표물의 측정값을 얻는 시간에 비하여 필터 수행시간은 극히 짧아 추적 필터 연산을 부동 소숫점으로 처

표 3. 추적결과

Table 3. Tracking result.

TARGET			WINDOW		
0	A	7D00 0	0	A	7D00 0000
0	A	7D00 0	0	A	7D00 7000
0	A	7A80 0	0	A	7A7F D000
0	A	7D00 0	0	A	7D00 5000
0	A	7800 0	0	A	7800 5580
0	A	7D00 0	0	A	7D00 9B60
0	A	7580 0	0	A	7581 51D8
0	A	7D00 0	0	A	7CF9 EE01
0	A	7300 0	0	A	72FF 8254
0	A	7D00 0	0	A	7CF3 E5A0
0	A	7080 0	0	A	707C A75C
0	A	7D00 0	0	A	7CEC B098
0	A	6E20 0	0	A	6E06 C7CE
0	A	7D00 0	0	A	7CE4 BE16
0	A	6BA0 0	0	A	6B8F 0358
0	A	7D00 0	0	A	7CDD 41AE
0	A	6940 0	0	A	6915 0A6D
0	A	7D00 0	0	A	7CD5 3C5C
0	A	66C0 0	0	A	669A 7702
0	A	7D00 0	0	A	7CCD 9050
0	A	6460 0	0	A	641D 92FC
0	A	7D00 0	0	A	7CC5 D8D6
0	A	61C0 0	0	A	61A1 0B30
0	A	7D00 0	0	A	7CBE CB48
0	A	5F60 0	0	A	5F24 A7B9
0	A	7D00 0	0	A	7CB7 F894

리하여도 실시간 구현이 가능함을 알 수 있다. 또한 여러가지 추적 성능의 저하 요인이 존재하는 상황에서도 좋은 추적 성능을 보일 수 있는 추적 알고리즘을 적용하여 필터 수행시간이 길어 지더라도 실시간

표 4. 시스템의 수행 시간표

Table 4. Computation time table of system.

용도		C25 명령어수	계산 소요 시간
추적 목표물 추출		91	약 35ms
필터 수행	고정 소숫점	714	0.284ms
	부동 소숫점	966	8.470ms
합계	고정 소숫점	805	약 35.284ms
	부동 소숫점	1057	약 43.470ms

표 5. 필터 연산 수행 시간표
Table 5. Computation time table of filter arithmetic.

연산법	용도	C25 명령어 수	계산 소요 시간
고정 소숫점	Kalman Gain	162	약 0.284ms
	Error Covari.	145	
	State Vector	198	
	Subroutines	141	
	기타	68	
부동 소숫점	Kalman Gain	231	약 8.470ms
	Error Covari.	138	
	State Vector	136	
	Subroutines	408	
	기타	53	

처리에 어려움이 없음을 알 수 있다. 33rpm레이다의 실시간 처리 시간이 1.8182s이다. 그런데 8개의 표적에 대한 추적을 동시에 수행하여도 고정 소숫점 처리를 하는 경우 약 0.28s, 부동 소숫점 처리를 하는 경우 약 0.35s정도의 시간이 소요된다.

V. 결 론

본 논문에서는 고속 신호처리 프로세서(digital signal processor)인 TMS320C25를 이용하여 실시간 TWS(track-while-scan) 시스템의 구현에 대하여 논하였다. 먼저 고정 소숫점 연산에 의해 칼만 필터를 구현할 때 생기는 FWL(finite word length)의 영향에 대하여 알아 보았다. 실험을 통하여 고정 소숫점 연산으로는 14, 16비트 이상의 레지스터 길이로 추적하여 FWL의 영향을 무시할 수 있음을 알았다. 실시간 TWS 시스템은 크게 3부분으로 구성되어 있다. 첫째는 칼만 필터의 수행을 위한 TWS 연산부이고, 둘째는 모니터상에 추적 상황을 표시하는 기능을 위한 스캔 컨버터(scan converter)이고, 마지막으로 전체 시스템의 제어를 위한 시스템 제어부로 구성하였다. 또한 멀티버스를 시스템 버스로 구성하였다. 실험을 통하여 실시간 TWS 시스템이 만족스럽게 동작함을 확인하였다. 구현된 시스템은 33rpm 레이다를 사용할 경우 실시간 요구시간이 1.8182s인데 반하여 부동 소숫점 연산에 의한 8표적의 추적 시간은 약 0.35s정도만이 요구된다. 고정 소숫점 연산에 의한 추적 시간은 약 0.28s가 요구된다. 그러므로 8개 정도의 표적을 실시간으로 추적하는 것은 충분히 가능하며 최대 20~30개 정도의 표적을 동시에 추적하

는 실시간 시스템 구현도 가능하리라 생각된다. 실제 해상 표적은 여러가지 장애 요인에 의하여 더욱 복잡한 추적 기법이 필요하게 될 것이다. 그러나 본 논문에서 구현된 시스템은 쉽게 알고리즘을 수정할 수 있고, 또한 알고리즘의 복잡성이 증가해도 8개 정도의 표적을 실시간 추적하는 것은 충분히 가능하다는 결론을 얻을 수 있다.

부 록

곱셈 후 반올림 오차 σ_r^2 은 식(10c)에 의하여 다음과 같이 유도할 수 있다.

$$\begin{aligned} \sigma_r^2 &= E[\Delta X_i^2(k+1)] \\ &= E\left[\sum_{j=0}^k (\tilde{\Phi})^{k-j} (\alpha(j) + \beta(j)) \sum_{i=0}^k (\alpha(i) + \beta(i))' (\tilde{\Phi}')^{k-1}\right] \end{aligned} \tag{A1}$$

식(A1)을 다시 정리하면 다음과 같다.

$$\begin{aligned} &= \sum_{j=0}^k \sum_{i=0}^k (\tilde{\Phi})^{k-j} E \left[\begin{matrix} (\alpha_1(j) + \beta_1(j)) (\alpha_1(i) + \beta_1(i)), \dots \\ \vdots \\ (\alpha_N(j) + \beta_N(j)) (\alpha_N(i) + \beta_N(i)), \dots \\ \vdots \\ (\alpha_1(j) + \beta_1(j)) (\alpha_N(i) + \beta_N(i)) \end{matrix} \right] (\tilde{\Phi}')^{k-1} \\ &\quad \dots (\alpha_N(j) + \beta_N(j)) (\alpha_N(i) + \beta_N(i)) \end{aligned} \tag{A2}$$

반올림 오차의 가정에 따라 $\alpha(i)$ 와 $\beta(i)$ 는 백색 잡음이다. 그러므로 식(A2)는 다음과 같다.

$$\begin{aligned} E[\Delta X_i^2(k+1)] &= \sum_{j=0}^k (\tilde{\Phi})^{k-j} \begin{bmatrix} \sigma^2 [\alpha_1(j) + \beta_1(j)], \dots, 0 \\ \vdots \\ 0, \dots, \sigma^2 [\alpha_N(j) + \beta_N(j)] \end{bmatrix} (\tilde{\Phi}')^{k-1} \\ &= \sum_{j=0}^k (\tilde{\Phi}) Q (\tilde{\Phi}')' (E_i^2/12) \end{aligned} \tag{A3}$$

參 考 文 獻

[1] R.E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng. Trans.*, ASME, vol. 82D, pp. 35-45, 1960.
[2] A.H. Jazwinski, *Stochastic Process and Filtering Theory* N.Y., Academic Press, 1970.

- [3] P.S. Maybeck, *Stochastic Models, Estimation and Control; volume1*, Academic Press, 1979.
- [4] B.D.O. Anderson and J.B. moore, *Optimal Filtering*, Prentice-Hall, 1979.
- [5] 이만형, 김종화, "칼만 필터링을 사용한 목표물 추적 시스템의 설계," 대한 전기학회 논문집, 제37권, 제29호, pp. 636-645, 1988.
- [6] Texas Instruments, *TMS320c25 User's Guide*, 1986.
- [7] L.B. Jackson, "On the interaction of roundoff noise and dynamic range in digital filters," *Bell Syst. Tech. J.*, vol. 49, pp. 159-184, 1970.
- [8] A. Gelb, *Applied Optimal Estimation*, The M.I.T. Press, 1974.
- [9] A.B. Stripad, "Performance degradation in digitally implemented kalman filters," *IEEE Trans. Aero. and Elec. Sys.*, vol. AES-17, pp. 626-634, Sept. 1981.
- [10] S.Y. Hwang, "Performance degradation in digitally implemented kalman filters," *IEEE Trans. Aero. and Elec. Sys.*, vol. AES-17, pp. 626-634, Sept. 1981.
- [11] 기석철, "DSP를 사용한 실시간 TWS 시스템 구현 연구," 석사학위 논문, 서울대학교 대학원 제어계측 공학과, 1989, 1.
- [12] J. Coonen et al, "A proposed standard for binary floating-point arithmetic," *ACM Signum Newsletter*, pp. 4-12, Oct. 1979.
- [13] IEEE Computer society, "IEEE Standard Microcomputer Systems Bus," 1983.
- [14] SAMSUNG, *MOS Memory Data Book*, 1988. *

 著 者 紹 介



奇錫哲(準會員)

1964年 4月 28日生. 1983年 3月
~1987年 2月 서울대학교 제어계
측 공학과 졸업. 1987年 3月~
1989年 2月 동대학원 석사과정.
1989年 3月~현재 삼성전자 정보
통신부문 통신연구소 응용 연구실

근무중. 주관심분야는 디지털 신호처리 등임.

李商郁 (正會員) 第25卷 第12號 參照

현재 서울대학교 제어계측공
학과 부교수