

이산 선형 시불변시스템에 대한 병렬칼만필터

김 용 준* 이 장 규** 김 형 중***

A Parallel Kalman Filter for Discrete Linear Time-invariant System

Yong Joon Kim, Jang Gyu Lee, and Hyoung Joong Kim

ABSTRACT

A parallel processing algorithm for discrete Kalman filter, which is one of the most commonly used filtering technique in modern control, signal processing, and communication, is proposed. Previously proposed parallel algorithms to decrease the number of computations needed in the Kalman filter are the hierarchical structures by distributed processing of measurements, or the systolic structures to disperse the computational burden. In this paper, a new parallel Kalman filter employing a structure similar to recursive doubling is proposed. Estimated values of state variables by the new algorithm converge with two times faster data processing speed than that of the conventional Kalman filter. Moreover it maintains the optimality of the conventional Kalman filter.

1. 서 론

칼만필터¹⁾는 현대제어, 신호처리, 그리고 통신 등의 분야에서 널리 적용되는 필터기법 중의 하나이다. 이것은 백색 가우스 잡음을 가지고 있는 시스템에 있어서 최적필터이며, 현재와 과거의 측정치 z_1, \dots, z_k 로부터 가능한 한 가장 정확한 시스템 상태변

수 \hat{x}_k 의 추정치를 순환적으로 얻는 기술이다.

그러나 칼만필터의 실시간 응용은 복잡한 행렬 계산식으로 구성된 필터 알고리즘의 계산량으로 인하여 제한되고 있다. 이의 해결 방법으로는 모델의 크기를 줄여서 계산량을 줄이는 방법이 있으므로, 칼만필터는 모델의 정확도에 민감하여 오차가 발생할 여지가 많다.

그러므로 충분한 크기의 정확한 모델에 대하여 실시간 칼만필터링을 하기 위해서 병렬성을 도입하여 계산량을 줄이려는 방법들³⁻⁷⁾이 제안되었다.

지금까지 이산 칼만필터에 병렬성을 도입한 연구들은 크게 계층적 구조의 알고리즘과 시스톨릭 구조

* 금성사 중앙연구소 연구원

** 서울대학교 제어계측공학과 부교수

*** 강원대학교 제어계측공학과 전임강사

를 사용한 알고리즘으로 나눌 수 있다. 계층적 구조의 알고리즘은 측정치를 여러 개의 작은 차수의 모델들로 나누거나 시간적으로 여러 개의 부분으로 나누고 하부프로세서들을 이용해 처리하여 각각의 추정치를 구하고, 이 값들을 상부프로세서로 전송하여 이곳에서 최적 추정치를 얻는 구조를 가지고 있다.

Hassan³⁾, Desai와 Das⁴⁾, Hashemipour⁵⁾ 등이 이러한 계층적 구조의 알고리즘을 제안하였다. VLSI 기술의 발달과 더불어 시스톨릭 구조⁸⁾라는 새로운 구조의 병렬처리 기법이 최근에 개발되었다. 시스톨릭 구조를 사용하면 행렬의 연산(행렬과 벡터의 곱셈, 행렬과 행렬의 곱셈, 행렬의 삼각화, 행렬의 회전)을 고속으로 처리할 수 있다.^{7, 9)}

이러한 장점 때문에 칼만필터의 복잡한 행렬의 연산에 시스톨릭 구조를 이용하여 성능향상을 얻으려는 연구들이 있었다. 이러한 연구는 Andrews⁵⁾가 Bierman의 LDU 분할에 적용한 것이 최초이고, Jover와 Kailath,⁶⁾ Sung과 Hu⁷⁾ 등이 이러한 연구들을 하였다. 이들은 제곱근필터의 특징을 이용하여 시스톨릭 구조의 병렬 칼만필터들을 제안하였다.

본 논문에서는 이러한 기존 알고리즘의 형태와는 다르게, Kogge와 Stone¹⁰⁾이 제안한 순환방정식에 대한 병렬처리 알고리즘의 기본개념을 사용하여 새로운 병렬 칼만필터 알고리즘을 제안한다. Kogge와 Stone은 반복이분법을 사용하여 일반적인 순환방정식에 대한 병렬 알고리즘을 제안하였다.

이들의 알고리즘에 의하여 연속된 N 개의 해를 구하는 순환방정식 문제를, N 개의 프로세서를 사용할 경우 $O(\log_2 N)$ 시간에 해결할 수 있다. 그러나 이들의 알고리즘을 적용하려면 모든 프로세서가 각각 자기의 데이터를 동시에 가지고 있어야 한다.

따라서 만일 데이터가 시간에 따라 순차적으로 입력되는 문제라면, 이들의 알고리즘을 적용할 수 없게 된다. 이산 칼만필터는 순환방정식의 형태이나, 실시간 운영에 적용할 때 시간에 따라 순차적으로 입력되는 측정치가 포함되므로, 실시간으로 운영되는 칼만필터에는 이들의 알고리즘을 적용할 수 없다.

본 논문은 Kogge와 Stone의 알고리즘이 반복이분법에 의하여 순환방정식을 병렬로 해결한 것과 유사하지만, 실시간으로 운영되는 칼만필터의 제약조

건에 알맞게 이산 칼만필터에 대한 새로운 병렬처리 알고리즘을 제안한다.

2. 병렬 칼만필터 알고리즘

본 절에서는 새로운 병렬 칼만필터 알고리즘을 제안한다. 다음과 같은 이산 선형 시불변시스템(discrete linear time invariant system)과 측정방정식(observation equation)을 생각해보자.

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (1)$$

$$\mathbf{z}_k = H \mathbf{x}_k + \mathbf{v}_k \quad (2)$$

식(1)과 식(2)는 이산 선형시스템과 측정방정식의 가장 일반적인 식으로, 식에서 \mathbf{w}_k 와 \mathbf{v}_k 는 정규분포를 갖는 백색잡음으로, 평균은 0이고 공분산행렬은 각각 Q , R 이다.

칼만필터는 상태변수 \mathbf{x}_k 와 추정상태변수 $\hat{\mathbf{x}}_k$ 의 오차분산 $E\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\}$ 을 최소로 하는 알고리즘으로 다음과 같은 두 단계에 의하여 최적 추정치를 찾게 된다.

$$\hat{\mathbf{x}}_k(-) = \Phi \hat{\mathbf{x}}_{k-1}(+) \quad (3)$$

$$P_k(-) = \Phi P_{k-1}(+) \Phi^T + Q \quad (4)$$

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + K_k [\mathbf{z}_k - H \hat{\mathbf{x}}_k(-)] \quad (5)$$

$$P_k(+) = [I - K_k H] P_k(-) \quad (6)$$

$$K_k = P_k(-) H^T [H P_k(-) H^T + R]^{-1} \quad (7)$$

알고리즘 유도의 편의상 우선 칼만필터 전개식에 서 상태변수에 관한 전개식만을 고려하기로 한다. 식(3)과 식(5)를 이용하여 $\hat{\mathbf{x}}_{h+i+1}(+)$ 를 $\hat{\mathbf{x}}_h(+)$ 의 함수로 표시하면 식(8)과 같이 표현된다.

$$\hat{\mathbf{x}}_{h+i+1}(+) = S_{i+1} \hat{\mathbf{x}}_h(+) + T_{i+1} \quad (8)$$

식(8)에서 S_{i+1} 과 T_{i+1} 은 다음과 같이 구해진다.

$$S_0 = I, T_0 = 0$$

$$S_{i+1} = [\Phi - K_{h+i+1} H \Phi] S_i \quad (9)$$

$$T_{i+1} = [\Phi - K_{h+i+1} H \Phi] T_i + K_{h+i+1} \mathbf{z}_{h+i+1} \\ \text{for } i = 0, 1, 2, \dots \quad (10)$$

이상으로부터 $\hat{\mathbf{x}}_{h+i+1}(+)$ 는 S_{i+1} , T_{i+1} 을 먼저 계산하고, 식(8)에 의하여 $\hat{\mathbf{x}}_h(+)$ 로부터 직접 구

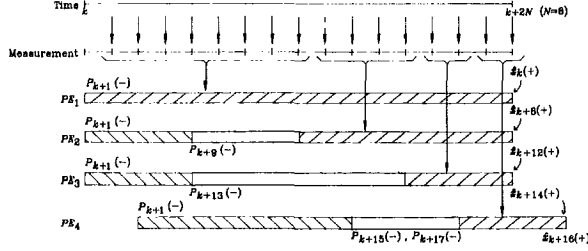


그림 1. 이산 시불변시스템에 대한 병렬칼만필터

할 수 있음을 알 수 있다.

식(8)에서부터 식(10)까지의 관계로부터 새로운 병렬칼만필터 알고리즘을 유도할 수 있다. 그림 1은 병렬 칼만필터의 시나리오를 나타내고 있는데, 먼저 그림을 사용하여 제안하는 알고리즘을 예를 들어 설명하고, 다음에 일반적으로 정리한다.

본 논문에서 제안하는 병렬 알고리즘은 일정시간 구간을 한 계산단계로 삼아, 칼만필터를 실시간으로 운영한다. 그림 1에서 첫번째 선은 시간축을 나타내며, $2N$ 은 병렬 알고리즘이 한 단계 실행될 때 소요되는 시간을, k 는 그 단계의 시작시간을 나타낸다. $2N$ 의 시간에 측정치는 $2N$ 개 입력된다고 가정하고, 각 프로세서들의 계산능력은 하나의 측정치를 2의 단위시간에 처리할 수 있다고 가정한다.

그러므로 일반적인 순차적 칼만필터를 사용하면, $2N$ 의 시간에 N 개의 측정치만을 처리할 수 있다. 그러나 본 논문은 $2N$ 의 시간에 $2N$ 개의 측정치를 처리할 수 있는 새로운 병렬 알고리즘을 제안한다. 설명을 쉽게 하기 위하여 $N=8$ 의 경우에 대하여 병렬 칼만필터 알고리즘을 전개해 본다.

PE_1 (processing element 1)은 시간 k 부터 $k+8$ 까지에 입력되는 8개의 측정치를 시간 k 부터 $k+16$ 까지의 시간에 처리한다. 식(9)와 식(10)에서 $h=k$ 로 하여, S_8 , T_8 을 계산하고, 식(8)의 관계에 의하여 $\hat{x}_k(+)$ 로부터 $\hat{x}_{k+8}(+)$ 를 계산한다. 그 다음 $\hat{x}_{k+8}(+)$ 를 PE_2 에 전송한다. PE_2 는 9번째 측정치부터 12번째 측정치까지 4개의 측정치를 시간 $k+8$ 부터 $k+16$ 까지의 시간에 처리한다.

$h=k+8$ 로 하여, S_4 , T_4 를 계산하고, PE_1 에서 받은 $\hat{x}_{k+8}(+)$ 로부터 $\hat{x}_{(k+8+4)}(+)$ 를 계산한다. 그 다음 $\hat{x}_{k+12}(+)$ 를 PE_3 에 전송한다. 비슷한 과정에

의하여 PE_3 는 PE_2 에서 받은 $\hat{x}_{k+12}(+)$ 로부터 $\hat{x}_{k+14}(+)$ 를 계산하여 PE_4 에 전송하며, PE_4 는 이것으로부터 $\hat{x}_{k+16}(+)$ 를 계산하며, 이 값을 다음 계산단계에서 사용하기 위해 PE_1 으로 전송한다.

이상과 같이 예를 들어 설명한 병렬 칼만필터 알고리즘을 수행하기 위하여 필요한 프로세서 갯수를 p 라고 하면, $p=(\log_2 N)+1$ 이다. 제안한 병렬 칼만필터 알고리즘을 일반적으로 정리하면 다음과 같다.

$$PE_i (i=1, \dots, (\log_2 N)+1) :$$

$$\text{계산시작시간} : h=k+2\sum_{j=1}^{i-1} \frac{N}{2^j}$$

$P_{h+1}(-)$ 는 주어짐

$$S_0=I, T_0=0$$

$$l=N-\sum_{j=1}^{i-1} \frac{N}{2^j}$$

(단, $i=(\log_2 N)+1$ 이면, $l=2$)

$$K_{h+1}=P_{h+1}(-)H^T[HP_{h+1}(-)H^T+R]^{-1}$$

$$P_{h+1}(+)= [I-K_{h+1}H]P_{h+1}(-)$$

$$S_1=[\Phi-K_{h+1}H\Phi]S_0$$

$$T_1=[\Phi-K_{h+1}H\Phi]T_0+K_{h+1}z_{h+1}$$

for $j=2, \dots, l$

$$P_{h+j}(-)=\Phi P_{h+j-1}(+) \Phi^T+Q$$

$$K_{h+j}=P_{h+j}(-)H^T[HP_{h+j}(-)H^T+R]^{-1}$$

$$P_{h+j}(+)= [I-K_{h+j}H]P_{h+j}(-)$$

$$S_j=[\Phi-K_{h+j}H\Phi]S_{j-1}$$

$$T_j=[\Phi-K_{h+j}H\Phi]T_{j-1}+K_{h+j}z_{h+j}$$

$(i-1)$ 번째 프로세서에서 $\hat{x}_h(+)$ 를 전송받는다.

$$\hat{x}_{h+l}(+) = S_l \hat{x}_h(+) + T_l$$

$\hat{x}_{h+1}(+)$ 를 $(i+1)$ 번째 프로세서로 전송한다.

단, 첫번째 프로세서는 상태변수의 초기치를 전 계산단계에서 마지막 프로세서로부터 전송받으며, 마지막 프로세서는 계산된 상태변수를 다음 계산단계에서 사용하기 위하여 첫번째 프로세서로 전송한다.

3. 공분산행렬의 초기치 계산

2절에서 실시간 운영의 칼만필터에 대한 병렬처리 알고리즘을 유도하였다. 이 알고리즘에서 보면, 각 프로세서가 S_j, T_j 를 계산할 때 칼만이득 K_{h+j} 를 필요로 하며, 이것은 공분산행렬 P_{h+j} 로부터 계산된다.

따라서 P_{h+j} 의 전달을 위하여 각 프로세서는 알고리즘의 매 계산단계의 계산시작 전에 각자의 공분산행렬의 초기치를 가지고 있어야 함을 알 수 있다. 즉, 그림 1의 경우 PE_1 은 $P_{k+1}(-)$ 를, PE_2 는 $P_{k+9}(-)$ 를, PE_3 는 $P_{k+13}(-)$ 를 그리고 PE_4 는 $P_{k+15}(-)$ 를 초기값으로 계산시작 전에 가지고 있어야 한다.

본절에서는 병렬 칼만필터 알고리즘을 수행하는 프로세서의 비동작시간을 이용하여, 이 값들을 계산하는 방법을 설명한다.

식(4), (6), (7)로부터 식(11)이 유도된다.¹¹⁾

$$\begin{aligned} P_{k+1}(-) &= \Phi[P_k(-) \\ &\quad - P_k(-)H^T(HP_k(-)H^T \\ &\quad + R)^{-1} \cdot HP_k(-)]\Phi^T + Q \\ &= [Q\Phi^{-T} \\ &\quad + (\Phi + Q\Phi^{-T}H^TR^{-1}H)P_k(-)] \\ &\quad \cdot [\Phi^{-T} + \Phi^{-T}H^TR^{-1}HP_k(-)]^{-1} \\ &\equiv [C + DP_k(-)][A + BP_k(-)]^{-1} \end{aligned} \quad (11)$$

식(11)에서의 A, B, C, D 는 모두 $n \times n$ 행렬이다. 이를 이용하여 식(12)와 같이 표현되는 선형시스템을 가정하자.

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} \quad (12)$$

식(12)에서 X_k, Y_k 는 각각 $n \times n$ 행렬로서 이것으로부터 식(13)과 같은 관계식을 얻을 수 있다.

$$\begin{aligned} Y_{k+1} X_{k+1}^{-1} &= [CX_k + DY_k][AX_k + BY_k]^{-1} \\ &= [CX_k + DY_k]X_k^{-1}[A + BY_k X_k^{-1}]^{-1} \\ &= [C + DY_k X_k^{-1}][A + BY_k X_k^{-1}]^{-1} \end{aligned} \quad (13)$$

식(11)과 식(13)으로부터 $Y_k X_k^{-1} = P_k(-)$ 를 가정하면, $Y_{k+1} X_{k+1}^{-1} = P_{k+1}(-)$ 의 관계가 성립함을 알 수 있다. 따라서 식(11)로 표현되는 이산 비선형방정식을 식(12)와 같은 선형방정식의 전개로 변형시킬 수 있으며, X_k, Y_k 로부터 $P_k(-)$ 를 계산할 수 있음을 알 수 있다. 실제로 $X_0 = I, Y_0 = P_0(-)$ 로 하면 $P_0(-) = Y_0 X_0^{-1}$ 이며, 이 X_0, Y_0 로부터 식(12)로 이용하여 선형차분방정식으로 X_k, Y_k 를 순환적으로 구할 수 있으며, 이로부터 $P_k(-)$ 를 구할 수 있다.

이상의 결과를 이용하여 공분산행렬의 초기치들의 계산하는 방법을 설명한다. 그림 1에서 $N=8$ 로 가정하면 한 계산단계에서 필요한 공분산행렬의 초기치는 4개이다. 알고리즘의 시작시간은 k 이고, 또 $x_k(+), P_{k+1}(-)$ 는 주어졌다고 가정한다. 특히 $P_{k+1}(-)$ 는 모든 프로세서에 주어졌다고 가정한다. 따라서 계산해야 할 분산행렬은 다음 계산단계에서 필요한 초기분산행렬을 포함하여 모두 4개로 $P_{k+9}(-), P_{k+13}(-), P_{k+15}(-), P_{k+17}(-)$ 이다.

PE_2 는 병렬 칼만필터 알고리즘의 비동작 시간을 이용하여, 다음의 계산과정에 의하여 $k+9$ 시간에 필요한 $P_{k+9}(-)$ 을 계산한다.

$$\begin{bmatrix} X_{k+9} \\ Y_{k+9} \end{bmatrix} = \Psi \begin{bmatrix} I \\ P_{k+1}(-) \end{bmatrix}, \quad \Psi = M^8 \quad (14)$$

$$P_{k+9}(-) = Y_{k+9} X_{k+9}^{-1} \quad (15)$$

유사한 과정에 의하여 PE_3 는 $P_{k+13}(-)$ 를 계산하고, PE_4 는 $P_{k+15}(-), P_{k+17}(-)$ 를 계산한다.

일반적으로 이 공분산행렬의 병렬 알고리즘을 정리하면 다음과 같다.

PE_i ($i=2, \dots, (\log_2 N)+1$) :

$P_{k+1}(-)$ 는 주어짐

$$l \equiv k+2 \left(\sum_{j=1}^{i-1} \frac{N}{2^j} \right) + 1$$

$$\begin{bmatrix} X_l \\ Y_l \end{bmatrix} = \Psi \begin{bmatrix} I \\ P_{k+1}(-) \end{bmatrix}, \quad \Psi = M^{l-k-1}$$

$$P_l(-) = Y_l X_l^{-1}$$

$PE_{(\log_2 N)+1}$ 은 다음을 더 계산한다 :

$$\begin{bmatrix} X_{k+2N+1} \\ Y_{k+2N+1} \end{bmatrix} = \Psi \begin{bmatrix} I \\ P_{k+1}(-) \end{bmatrix}, \quad \Psi = M^{2N}$$

$$P_{k+2N+1}(-) = Y_{k+2N+1} X_{k+2N+1}^{-1}$$

4. 결 론

본 논문에서는 이산 선형 시불변시스템에 대한 칼만필터의 병렬 알고리즘을 제안하였다. 본 논문이 제안한 병렬 칼만필터는 Kogge와 Stone³⁾이 제안한 순환방정식에 대한 병렬처리 알고리즘의 기본개념을 사용한 새로운 형태의 알고리즘이다.

본 논문이 제안한 병렬 칼만필터는 일정시간 구간을 한 계산단계로 하여, 칼만필터를 실시간으로 운영한다. 그리고 일반적인 칼만필터보다 두배의 빠르기로 추정치를 처리하기 때문에 상태변수 추정치가 참값에 더욱 빨리 수렴한다.

병렬 칼만필터 알고리즘을 수행하기 위하여 각 프로세서는 매 계산단계의 계산시작 전에 각각 자신의 공분산행렬의 초기치를 가지고 있어야 하는데, 이를 위하여 병렬 칼만필터 알고리즘을 수행하는 프로세서들의 비동작시간을 이용하여 이 값들을 계산하는 방법을 제안하였다.

본 논문이 제안한 병렬 칼만필터 알고리즘은 실시간 운영이지만, 실제로 실시간으로 사용될 수 있는 상태변수는 각 계산단계의 마지막 상태변수 뿐이다. 이러한 이유로 본 논문이 제안한 병렬 칼만필터는 매 샘플링 시간마다 계속적으로 상태변수 추정치를

구해야 하는 문제에는 적용 불가능하고, 일정한 시간간격으로 정확한 상태변수 추정치를 필요로 하는 문제에 적합하다.

참 고 문 헌

1. A. Gelb, *Applied Optimal Estimation*, M.I. T. Press, 1974.
2. Mohamed F. Hassan, G. Salut, Madan G. Singh, and Andre Titli, "A Decentralized Computational Algorithm for the Global Kalman Filter," *IEEE Trans. Automat. Contr.*, Vol. AC-23, No. 2, pp. 262-268, 1978.
3. U. B. Desai and B. Das, "Parallel Algorithms for Kalman Filtering," in *Proc. 1985 Amer. Contr. Conf.*, Boston, MA, pp. 920-921, 1985.
4. Hamid R. Hashemipour, Sumit Roy, and Alan J. Laub, "Decentralized Structures for Parallel Kalman Filtering," *IEEE Trans. Automat. Contr.*, Vol. 33, No. 1, pp. 88-94, 1988.
5. A. Andrews, "Parallel Preprocessing of the Kalman Filter," in *Proc. Int. Conf. Parallel Processing*, Columbus, Ohio, pp. 216-220, 1981.
6. J. M. Jover and T. Kailath, "A Parallel Architecture for Kalman Filter Measurement Update and Parameter Estimation," *Automatica*, Vol. 22, No. 1, pp. 43-57, 1986.
7. Tze-Yun Sung and Yu-Hen Hu, "Parallel VLSI Implementation of the Kalman Filter," *IEEE Trans. Aero. Elec. Syst.*, Vol. AES-23, No. 2, pp. 215-224, 1987.
8. H. T. Kung, "Why Systolic Architecture?," *IEEE Computer*, pp. 37-46, Jan., 1982.
9. C. Mead and L. Conway, *Introduction to*

- VLSI Systems*, Addison Wesley, 1980.
10. Peter M. Kogge and Harold S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Trans. Computers*, Vol. C-22, No. 8, pp. 786-793, 1973.
 11. Brian D. O. Anderson and John B. Moore, *Optimal Filtering*, Prentice-Hall, 1979.