

GPS 수신기를 위한 항법필터의 설계 및 구현

Design and Implementation of a Navigation Filter for GPS Receiver

홍진석*, 지규인*, 이장규**, 오종택***

Jin Seok Hong*, Gyu In Jee*, Jang Gyu Lee**, and Jong Taek Oh***

요 약

GPS 수신기는 잡음이 섞인 의사거리와 도플러 측정값으로부터 사용자의 위치, 속도 및 시간을 결정하여야 한다. GPS 수신기는 일반적으로 최소자승법(Least Square Method)을 사용한다. 칼만필터(Kalman Filter)는 시스템의 동적특성과 시스템 잡음의 확률적 특성을 이용하는 최적 예측기로 알려져 있다. 본 논문에서는 GPS 수신기를 위한 8차 칼만 필터를 설계하고, 필터의 계산 부담을 줄이고 안정도를 향상시키기 위해 UD칼만필터, Steady-State 칼만필터, Schmidt 칼만필터 등의 기법들을 적용하였다. 설계된 필터의 성능을 실제 수신기에 구현하여 검증하고 결과를 분석하였다.

Abstract

Determination of position, velocity and time from the satellite signals is the central problem of the GPS receiver. Generally, GPS receiver uses least square method for navigation filter algorithm. The Kalman filter has known as an optimal linear estimator, which uses the knowledge of the system dynamics and the statistical characteristics of the system noises and measurement error. In this paper, we design an 8 state Kalman filter for GPS navigation and employ some techniques that can reduce computational burden, increase numerical stability, and improve filtering performance. To show its effectiveness, the designed filter is implemented in a GPS receiver and tested in real-time.

I. 서 론

대표적인 위성항법 시스템인 GPS(Global Positioning System)의 수신기는 위성이 보내주는 신호로부터 위치, 속도, 시각 등의 정보를 결정해야 한다. 위성으로 받은 신호는 여러 종류의 오차와 잡

음을 가지고 있기 때문에 이러한 잡음이 섞인 신호의 측정값으로부터 정확한 위치, 속도, 시각 등을 정확하게 추정하는 것은 중요한 문제이다. 보통의 수신기에서는 의사거리와 도플러 측정값을 이용하여 사용자의 위치와 속도를 추정하기 위해서 최소자승법(Least Square Method) 알고리즘을 사용한다[2],[4]. 최소자승법을 사용하는 경우에는 보

*건국대학교 전자공학과(Dept. of Electronics, KonKuk Univ.)

**서울대학교 전기공학부(School of Electrical Engineering, Seoul National Univ.)

***한국통신 무선통신연구소(Wireless Communication Research Lab, Korea Telecom.)

· 논문번호 : 97-1-2

· 접수일자 : 1997년 12월 4일

· 이 논문은 한국통신에서 지원한 '97 정보통신 기초연구과제의 연구결과임.

통 위치 필터를 이용하여 사용자의 위치와 수신기 시계의 바이어스 부분을 추정하고 속도 필터를 이용하여 사용자의 속도와 수신기 시계의 드리프트를 결정한다. 칼만필터를 이용하면 사용자의 동적 특성과 측정값의 확률적 특성을 이용하여 최적의 추정치를 얻을 수 있음이 알려져 있다[3],[5]. 그러나 실질적인 문제에 칼만필터를 적용할 경우에는 추정하는 상태변수의 차수가 증가함에 따라 계산상의 어려움이 커진다. 본 논문에서는 GPS 수신기에 내장되어 사용되기 위한 8개의 상태 변수를 가지는 GPS 항법용 필터를 설계하고 필터의 계산 부담을 줄이고 안정도 및 정확도를 향상시키는 칼만필터 기법에 대해서 구현하여 실제로 GPS 수신기에 적용한 실험결과를 보여준다.

II. GPS를 위한 칼만 필터 구성

GPS가 단독으로 사용될 경우에는 INS와 같이, 기준이 되는 시스템이 없기 때문에 선형 칼만 필터가 아닌 확장 칼만 필터가 사용된다[5]. 따라서 그림 1에서와 같이 칼만 필터의 추정된 값을 이용하여 가상의 기준에 대한 오차를 추정한다. 이런 경우 추정할 오차는 사용하는 응용에 따라 달라지게 된다. 움직이지 않는 사용자가 되면 5차의 랜덤 바이어스 오차 모델이 최적의 모델이 된다. 본 논문에서는 저속으로 움직이는 사용자를 가정하여 8차의 모델을 사용하여 움직이는 상태를 추정하였다.

2-1 GPS 모델링

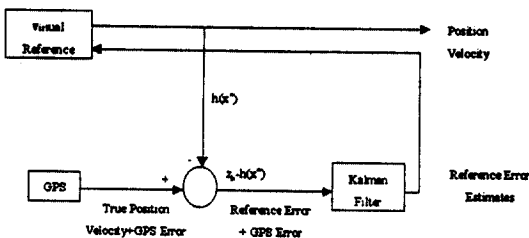


그림 1. Stand-alone GPS Kalman filter.
Fig. 1. Stand-alone GPS Kalman filter.

2-1-1 이동체 운동 모델링(Kinematic Modeling)

사용자의 운동을 그림 2에 보이는 위치, 속도 모델을 사용하였다. 위치, 속도 모델을 상태변수로 표시하면 식 (1)과 같다. 입력 스펙트럼의 크기 S_p 는 사용자의 동적 특성을 고려하여 결정한다.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ w_p \end{bmatrix} \quad (1)$$

2-1-2 수신기 시계 모델링

GPS 수신기는 항법을 위해서는 수신기 시계의 바이어스와 드리프트를 추정해야 한다. 일반적으로 사용되는 수신기 시계 모델은 2차 랜덤 프로세스 모델이다. 2개의 상태 변수는 수신기 시계오차의 위상과 주파수가 된다. 그림 3에 수신기 시계의 모델을 나타냈고 모델은 식 (2)로 주어진다.

입력 잡음의 스펙트럼 크기 S_g 와 S_f 는 Allan variance값과 관련이 있다. 수신기 시계는 보통 수정 발진자이므로 사용된 값은 $S_f = 2h_0$, $S_g = 8\pi^2 h_{-2}$ 으로 $h_0 = 2(10^{-19})$, $h_{-2} = 2(10^{-20})$ 이다.[5] 이 값을 수신기에서 사용할 때는 거리의 단위로 환산을 해야 하므로 광속의 제곱을 곱해서 사용하여야 한다.

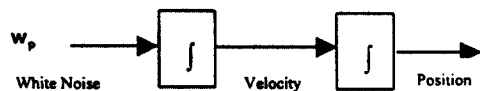


그림 2. Integrated random walk model.
Fig. 2. Integrated random walk model.

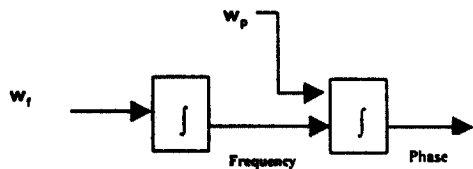


그림 3. 수신기 시계 모델
Fig. 3. Receiver clock model.

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_f \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_p \\ x_f \end{bmatrix} + \begin{bmatrix} w_p \\ w_f \end{bmatrix} \quad (2)$$

2-1-3 전체 시스템

3차원 위치 오차성분, 속도 오차성분, 그리고 2차의 시계 오차모델로 구성된 8차 상태변수 모델은 식 (3)이 된다[5].

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 \\ w_2 \\ 0 \\ w_4 \\ 0 \\ w_6 \\ w_7 \\ w_8 \end{bmatrix} \quad (3)$$

여기서 $x_1 = x$ position, $x_2 = x$ velocity, $x_3 = y$ position, $x_4 = y$ velocity, $x_5 = z$ position, $x_6 = z$ velocity, $x_7 =$ clock bias error, $x_8 =$ clock drift error가 된다.

2-2 측정 방정식(Measurement Equation)

GPS 위성으로부터 얻을 수 있는 측정치에는 의사거리, 도플러, 반송파 위상 등의 정보가 있다. 후처리를 이용하여 위치만을 구하기 위해서는 의사거리 정보만을 사용할 수 있으나 실시간으로 수신기가 동작하기 위해서는 수신기의 시계가 GPS 기준 시각과 동기되어야 하기 때문에 수신기 시계의 바이어스 및 드리프트를 결정해야 하므로 의사거리와 도플러 측정값을 필터의 측정값으로 이용하여 시계 모델의 계수를 추정하여야 한다.

2-2-1 의사거리 측정 방정식

위성과 사용자 사이의 거리는 다음식으로 표현된다.

$$\rho = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2 + (z_s - z_u)^2} + ct_u \quad (4)$$

여기서 (x_s, y_s, z_s) 는 위성의 위치, (x_u, y_u, z_u) 는 사용자의 위치, t_u 는 수신기 시계 바이어스이다. 위식은 비선형이므로 기준위치 $\hat{\rho}_0(x_0, y_0, z_0)$ 에 관하여 선형화하여 다음의 식을 얻을 수 있다.

$$\rho - \hat{\rho}_0 = \frac{\partial \rho}{\partial x_0} \Delta x + \frac{\partial \rho}{\partial y_0} \Delta y + \frac{\partial \rho}{\partial z_0} \Delta z + c\Delta t_u \quad (5)$$

선형화된 측정식을 위성이 n 개인 경우에 대하여 정리하면 다음과 같다.

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} h_{1x} & 0 & h_{1y} & 0 & h_{1z} & 0 & 1 & 0 \\ h_{2x} & 0 & h_{2y} & 0 & h_{2z} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{nx} & 0 & h_{ny} & 0 & h_{nz} & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} v_{\rho 1} \\ v_{\rho 2} \\ \vdots \\ v_{\rho n} \end{bmatrix} \quad (6)$$

여기서 $h_{ix} = \frac{\partial \rho_i}{\partial x_0}$, $h_{iy} = \frac{\partial \rho_i}{\partial y_0}$, $h_{iz} = \frac{\partial \rho_i}{\partial z_0}$ 는 위성과 사용자의 시선(Line of Sight)벡터, r_i 는 $\rho - \hat{\rho}_0$ 로 수신기 채널의 의사거리 측정값과 예측값의 차, v_{ρ_i} 는 측정잡음이다.

2-2-2 도플러 측정 방정식

위성과 사용자의 운동에 의한 도플러의 모델은 다음과 같이 표시할 수 있다.

$$\dot{\rho} = \frac{(x_s - x_u)}{\rho} (\dot{x}_s - \dot{x}_u) + \frac{(y_s - y_u)}{\rho} (\dot{y}_s - \dot{y}_u) + \frac{(z_s - z_u)}{\rho} (\dot{z}_s - \dot{z}_u) + c\Delta \dot{t} \quad (7)$$

측정값과 계산이 가능한 값을 우변으로 정리하고 미지수 항을 우변으로 정리하면 다음과 같다.

$$\begin{aligned} \dot{\rho} - \dot{x}_s \frac{(x_s - x_u)}{\rho} - \dot{y}_s \frac{(y_s - y_u)}{\rho} - \dot{z}_s \frac{(z_s - z_u)}{\rho} \\ = \dot{x}_u \frac{(x_s - x_u)}{\rho} + \dot{y}_u \frac{(y_s - y_u)}{\rho} + \dot{z}_u \frac{(z_s - z_u)}{\rho} + c\Delta i_u \\ \dot{\rho} - h_x \dot{x}_s - h_y \dot{y}_s - h_z \dot{z}_s = h_x \dot{x}_u + h_y \dot{y}_u + h_z \dot{z}_u + c\Delta i_u \\ d = h_x \dot{x}_u + h_y \dot{y}_u + h_z \dot{z}_u + c\Delta i_u \end{aligned} \quad (8)$$

여기서 $d = \dot{\rho} - h_x \dot{x}_s - h_y \dot{y}_s - h_z \dot{z}_s$
 n 개의 위성에 대해 정리하면 식 (9)와 같다.

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} 0 & h_{1x} & 0 & h_{1y} & 0 & h_{1z} & 0 & 1 \\ 0 & h_{2x} & 0 & h_{2y} & 0 & h_{2z} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & h_{nx} & 0 & h_{ny} & 0 & h_{nz} & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} v_{d1} \\ v_{d2} \\ \vdots \\ v_{dn} \end{bmatrix} \quad (9)$$

의사거리 측정값과 도플러 측정값을 모아서 n 개의 위성에 대해 8차 위치, 속도모델의 상태벡터로 표현하면 다음과 같다.

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} h_{1x} & 0 & h_{1y} & 0 & h_{1z} & 0 & 1 & 0 \\ h_{2x} & 0 & h_{2y} & 0 & h_{2z} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{nx} & 0 & h_{ny} & 0 & h_{nz} & 0 & 1 & 0 \\ 0 & h_{1x} & 0 & h_{1y} & 0 & h_{1z} & 0 & 1 \\ 0 & h_{2x} & 0 & h_{2y} & 0 & h_{2z} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & h_{nx} & 0 & h_{ny} & 0 & h_{nz} & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} v_{\rho 1} \\ v_{\rho 2} \\ \vdots \\ v_{\rho n} \\ v_{d1} \\ v_{d2} \\ \vdots \\ v_{dn} \end{bmatrix} \quad (10)$$

행렬식으로 표기하면 다음과 같다.

$$Z = Hx + v \quad (11)$$

III. 실제 시스템에서의 Kalman Filter

일반적인 형태의 칼만 필터를 2장에서 정의된 시스템에 적용한다면 매 순간 변하는 측정치에 따라 시스템이 변하게 되므로 이에 따른 고려를 해야 한

다. 또한 8개의 위성이 보이는 경우를 가정하면 매 순간 16×16 행렬의 역행렬을 매 순간 구해야 한다. 그러나 GPS와 같이 측정값이 서로 독립인 경우는 순차형 칼만 필터를 사용하여 이 문제를 해결할 수 있다. GPS의 경우 시스템 행렬의 성감성(sparse matrix)이 크므로 이를 이용하여 계산량을 줄일 수 있다[6]. 또한 수치적 안정도를 증가시키기 위해서는 U-D 칼만 필터를 사용할 수 있고[3],[6] 유색잡음 측정잡음 문제를 해결할 수 있는 Schmidt 칼만 필터와 계산량을 줄이는 Steady-State 칼만필터 등을 고려할 수 있다[7],[8].

3-1 이산형 칼만필터

아래의 식으로 표시되는 이산 시스템을 고려하면

$$\begin{aligned} X_{k+1} &= \Phi_k X_k + w_k \\ Z_k &= H_k X_k + v_k \end{aligned} \quad (12)$$

여기서

$$\begin{aligned} E(w_k w_l^T) &= \begin{cases} Q_k & k=l \\ 0_{p \times p} & otherwise \end{cases} \\ E(v_k v_l^T) &= \begin{cases} R_k & k=l \\ 0_{m \times m} & otherwise \end{cases} \\ E(w_k v_l^T) &= 0_{p \times m} \end{aligned} \quad (13)$$

다음과 같은 칼만 필터식을 유도할 수 있다.

$$\begin{aligned} \hat{x}_k &= \hat{x}_{\bar{k}} + K_k (z_k - H_k \hat{x}_{\bar{k}}) \\ P_k &= (I - K_k H_k) P_{\bar{k}} \\ \hat{x}_{k+1} &= \Phi_k \hat{x}_k \\ P_{k+1} &= \Phi_k P_k \Phi_k^T + Q_k \\ K_k &= P_{\bar{k}} H_k^T (H_k P_{\bar{k}} H_k^T + R_k)^{-1} \end{aligned} \quad (14)$$

3-2 Sequential filter

GPS의 측정값이 독립이라고 가정을 하면 측정값 잡음의 공분산 행렬이 양의 대각선 행렬이므로

측정값 갱신을 순차 칼만 필터 알고리즘을 사용할 수 있다.

Sequential Kalman filter algorithm:

q 개의 측정값을 가정하면 식 (15)의 벡터와 행렬을 정의할 수 있다.

$$\begin{aligned} z_k &= [z^1 \dots z^q]^T \\ H_k &= [h^1 \dots h^q]^T \\ R_k &= \text{diag}[r^1 \dots r^q]^T \end{aligned} \quad (15)$$

초기 공분산 행렬과 상태벡터를 식 (16)으로 정의한 후 식 (17)의 반복 계산을 수행하여 측정값 갱신을 할 수 있다.

$$P^0 = P_k, \hat{x}_0 = \hat{x}_k \quad (16)$$

For $i = 1 : q$

$$\left\{ \begin{aligned} k^i &= \frac{1}{(h^i)^T P^{i-1} h^i + r^i} P^{i-1} h^i \\ \hat{x}^i &= \hat{x}^{i-1} + (z^i - (h^i)^T \hat{x}^{i-1}) k^i \\ P^i &= P^{i-1} - k^i (h^i)^T P^{i-1} \end{aligned} \right. \quad (17)$$

최종 측정값 갱신은 $\hat{x}_k = \hat{x}^q$ 이다.

3-3 U-D covariance factorization filter

U-D 칼만 필터를 사용함으로써 매순간 변하는 측정치 벡터에 따른 변화에 능동적으로 대처할 수 있고 또한 정밀도에서도 일반적인 칼만 필터에 비하여 2배의 효과를 얻을 수 있다.

3-3-1 U-D 칼만필터의 공분산 행렬

측정값 전후의 공분산 행렬을 다음의 관계식을 다음으로 정의할 수 있다.

$$\begin{aligned} P(t_i^-) &= U(t_i^-) D(t_i^-) U^T(t_i^-) \\ P(t_i^+) &= U(t_i^+) D(t_i^+) U^T(t_i^+) \end{aligned} \quad (18)$$

여기서 U 행렬은 대각선 원소는 1인 상위 삼각 대각 행렬, D 행렬은 대각선 행렬이다.

3-3-2 U-D 칼만필터의 측정값 갱신

측정값 z_i 에 대한 UD칼만 필터의 측정값 갱신은 아래의 알고리즘에 의하여 이뤄진다. 여기서 $H(t_i)$ 는 1-by- n 벡터 그리고 R 은 측정값에 대한 공분산이 된다.

$$\begin{aligned} f &= U^T(t_i^-) H^T(t_i) \\ v_j &= D_{jj}(t_i^-) f_j \quad j = 1, 2, \dots, n \\ a_n &= R \end{aligned} \quad (19)$$

$k = 1, 2, n$ 에 대해 다음을 계산한다.

$$\left. \begin{aligned} a_k &= a_{k-1} + f_k v_k \\ D_{kk}(t_i^+) &= D_{kk}(t_i^-) a_{k-1} / a_k \\ b_k &\leftarrow v_k \\ p_k &= -f_k / a_{k-1} \\ U_{jk}(t_i^+) &= U_{jk}(t_i^-) + b_j p_k \\ b_j &\leftarrow b_j + U_{jk}(t_i^-) v_k \end{aligned} \right\} j = 1, 2, \dots, (k-1) \quad (20)$$

식 (20)에서 \leftarrow 는 효율적인 변수 사용을 위하여 이전의 값에 덮어쓰음을 의미한다. 첫번째 순간에는 처음 3개의 식만을 계산하고 n 번 반복 후 $U(t_i^+)$ 와 $D(t_i^+)$ 가 계산된다. 칼만 필터의 이득 $K(t_i)$ 는 n 차원의 벡터 b 를 이용하여 다음과 같이 구하고 상태 변수를 식 (21)을 이용하여 갱신한다.

$$\begin{aligned} K(t_i) &= b/a_n \\ \hat{x}(t_i^+) &= \hat{x}(t_i^-) + K(t_i)[z_i - H(t_i)\hat{x}(t_i^-)] \end{aligned} \quad (21)$$

3-3-3 U-D 칼만필터의 time propagation

일반적인 칼만 필터의 공분산 행렬의 시간 천이식은 다음과 같다.

$$P(t_{i+1}^-) = \Phi(t_{i+1}, t_i)P(t_i^+)\Phi^T(t_{i+1}, t_i) + G_d(t_i)Q_d(t_i)G_d^T(t_i) \quad (22)$$

본 논문에서는 Modified Weighted Gram-Schmidt(MWGS)방법을 이용하여 시간에 대한 공분산 행렬의 천이를 구현하였다. 다음과 같은 $n \times (n+s)$ 행렬 $Y(t_{i+1}^-)$ 와 $(n+s) \times (n+s)$ 대각 행렬 $\tilde{D}(t_{i+1}^-)$ 을 정의한다.

$$Y(t_{i+1}^-) = [\Phi(t_{i+1}, t_i)U(t_i^+); G_d(t_i)]$$

$$\tilde{D}(t_{i+1}^-) = \begin{bmatrix} D(t_i^+) & 0 \\ 0 & Q_d(t_i) \end{bmatrix} \quad (23)$$

각각의 크기가 $n+s$ 인 n 개의 벡터 a_n 을 초기화 한다.

$$[a_1; a_2; \dots; a_n] = Y^T(t_{i+1}^-) \quad (24)$$

$k = n, n-1, \dots, 1$ 에 대해 다음을 계산한다.

$$c_k = \tilde{D}(t_{i+1}^-)a_k$$

$$(c_{kj} = \tilde{D}_{jj}(t_{i+1}^-)a_{kj}, \quad j = 1, 2, \dots, n+s)$$

$$D_{kk}(t_{i+1}^-) = a_k^T c_k$$

$$d_k = c_k / D_{kk}(t_{i+1}^-)$$

$$U_{jk}(t_{i+1}^-) = a_j^T d_k$$

$$a_j \leftarrow a_j - U_{jk}(t_{i+1}^-)a_k \quad \left. \right\} j = 1, 2, \dots, k-1 \quad (25)$$

상태변수에 대한 천이식은 다음식으로 주어진다.

$$\hat{x}(t_{i+1}^-) = \Phi(t_{i+1}, t_i)\hat{x}(t_i^+) \quad (26)$$

3-4 Schmidt Kalman filter

칼만필터가 최적으로 동작하기 위해서는 시스템 잡음 및 측정잡음의 특성이 가우시안 백색잡음이어야 한다. 그러나 GPS 칼만필터의 측정값으로 사용되는 의사거리에는 식 (4)와 달리 다음과 같이 주요한 오차 요인인 S/A가 포함되어 있다.

$$\rho_{\text{measure}} = \rho_{\text{true}} + ct + S/A + v \quad (27)$$

S/A는 인위적으로 작용되는 의사거리 오차항으로 일반적으로 수분의 주기를 갖는 유색잡음으로 알려져 있다. 그러므로 유색잡음인 S/A를 측정잡음에 포함시켜 백색잡음으로 간주한다면 칼만필터에 의한 최적추정이 불가능하다. 이를 해결하기 위한 방법으로 의사거리에 포함되는 S/A를 Shaping Filter를 통해 모델링하여 이를 추정하는 방법이 있다. 그러나 이 방법을 사용하여 S/A를 2차 마코프프로세스로 모델하면 4개의 위성이 보이는 경우 전체 필터의 차수는 8(system)+8(S/A)=16차가 되며 8개의 위성이 보이는 경우에는 8+16=24차가 된다. 이러한 큰 차수의 칼만필터를 수신기 내부에 사용하기에는 계산상의 문제가 있으며 또한 필터의 목적이 S/A추정에 있는 것이 아니므로 S/A를 직접 추정하지 않고 S/A에 의한 불확정성의 정도만을 상태변수 추정에 이용하는 방법을 사용하는 것이 더욱 효율적이라 할 수 있다.

이러한 경우 사용할 수 있는 것이 Schmidt 칼만 필터이다[7],[8]. Schmidt 필터에서는 직접추정하지 않는 상태변수를 고려상태변수(Consider State)로 정의하여 이 고려상태변수의 공분산 값을 상태 변수 추정에 반영하는 구조를 가지고 있다. 이 Schmidt 칼만 필터를 GPS필터에 이용하는 경우에도 각 의사거리에 존재하는 S/A를 2차로 모델링하면 고려상태변수의 차수가 역시 증가하게 된다. 본 논문에서는 위치해에 존재하게 되는 S/A의 영향을 고려상태변수로 모델한다. 위치해에 존재하는 S/A의 영향을 간단하게 모델링하면 다음과 같이 1차 가우스 마코프프로세스로 모델할 수 있다.

$$\dot{x}_c = -\beta x_c + \sqrt{2\sigma^2\beta} \quad (28)$$

3차원 위치성분마다 S/A의 영향을 모델하면 3차의 고려상태변수가 구성된다.

Schmidt Kalman filter algorithm:

시스템 모델과 측정값 모델을 상태변수와 고려상

태변수(consider state)로 구성하여 다음식으로 나타낼 수 있다.

$$\begin{bmatrix} x \\ y \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi_x & 0 \\ 0 & \Phi_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_k + \begin{bmatrix} w_x \\ w_y \end{bmatrix}_k$$

$$z_k = [H \quad J]_k \begin{bmatrix} x \\ y \end{bmatrix}_k + v_k \quad (29)$$

여기서 x 는 상태변수 y 는 고려상태변수가 된다. 이 때 공분산 행렬은 다음 식이 된다.

$$Q_k = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix}_k, P_k = \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix}_k \quad (30)$$

Schmidt 칼만 필터 알고리즘은 다음과 같다. 칼만 필터의 이득은 다음과 같이 계산된다.

$$K_x = (P_{xx} H^T + P_{xy} J^T) \alpha^{-1} \quad (31)$$

여기서 $\alpha = HP_{xx}H^T + HP_{xy}J^T + JP_{yx}H^T + JP_{yy}J^T + R$ 이다.

다음으로 상태 변수를 갱신한다. 이때 고려 상태 변수는 추정하지 않는다.

$$\hat{x} = \hat{x}^- + K_x(z - H\hat{x}^-) \quad (32)$$

공분산 행렬을 갱신한다.

$$\begin{aligned} P_{xx} &= (I - K_x H) P_{xx}^- - K_x J P_{yx}^- \\ P_{xy} &= (I - K_x H) P_{xy}^- - K_x J P_{yy}^- \\ P_{yx} &= P_{yx}^T \\ P_{yy} &= P_{yy}^- \end{aligned} \quad (33)$$

시간 천이에 의한 상태변수와 공분산행렬은 다음식으로 계산된다.

$$\begin{aligned} \hat{x}_{k+1} &= (\Phi_x)_k \hat{x}_k \\ (P_{xx})_{k+1} &= (\Phi_x)_k (P_x)_k (\Phi_x)_k^T + (Q_x)_k \\ (P_{xy})_{k+1} &= (\Phi_x)_k (P_{xy})_k (\Phi_y)_k^T \end{aligned} \quad (34)$$

$$\begin{aligned} (P_{yx})_{k+1} &= (P_{xy})_{k+1}^T \\ (P_{yy})_{k+1} &= (\Phi_y)_k (P_{yy})_k (\Phi_y)_k^T + (Q_y)_k \end{aligned}$$

3-5 Steady-State Kalman filter

칼만필터가 시스템 모델의 변화없이 정상상태에 도달하면 필터의 공분산 행렬의 크기가 일정해지므로 필터의 이득 행렬이 변하지 않게 된다. 따라서 이 때에는 필터의 모든 알고리즘을 수행하지 않고 정상상태 칼만 이득을 사용하여 필터를 동작시킬 수 있다. 본 논문에서는 식 (35)로 주어진 필터의 공분산 행렬의 대각선 항의 합의 크기를 이전 상태와 비교하여 정상상태를 판단하고 모델이 변하지 않는다면, 즉 가시 위상이 변화하지 않는다면 정상상태 알고리즘을 이용하여 항법해를 구하는 방법을 사용하였다.

$$abs(sum(diag(P_k - P_{k-1}))) < Threshold \quad (35)$$

IV. 실험 및 분석

구현된 알고리즘의 성능을 평가하기 위하여 후처리를 통하여 일반적으로 사용되고 있는 최소 자승법 알고리즘, GPS의 경우 시스템 천이 행렬이 성검 행렬(sparse matrix)이고 각 채널의 측정값이 독립임을 이용한 순차형 칼만필터, UD 칼만필터, Schmidt 칼만 필터, 정상상태 칼만필터 등의 형태로 구현하여 비교하였다. 또한 UD 칼만 필터를 이용하여 구한 항법용 알고리즘을 GPS 개발용 보드인 GEC Plessey사의 Architect에 구현하여 도심 구간을 실시간 주행을 하였다. Architect는 ARM 60마이크로 프로세서를 사용한다.

4-1 알고리즘 성능 비교

구현된 알고리즘이 실제 시스템에서 사용될 때는 알고리즘의 정확도, 또한 사용하는 메모리의 양, 계산 시간 등 여러가지 사항을 고려하여야 한다. 표 1은 MATLAB상에서 알고리즘을 구현하여 실시 주

표 1. 알고리즘 성능 비교

Table 1. Comparison of the algorithm performance.

알고리즘	Flops
LSQ	3821801
Conventional KF	27534073
Sequential KF	31456675
Schmidt KF	56872795
Steady State KF	9456164
UD KF	11449715

행 데이터를 후처리하여 비교한 것이다. 성능을 살펴보면 최소자승법(LSQ)은 계산과 스택의 사용면에서 이점을 가지고 있다. 일반적인 칼만 필터의 경우는 계산시간의 장점을 가지고 있지만 계산량과 특히 스택의 사용량이 많기 때문에 실지 시스템에 구현하기에는 적합하지 않다. 순차형 칼만 필터의 경우는 시간과 특히 스택의 사용면에서 그리고 역행렬을 구하지 않아도 된다는 장점을 가지고 있다. 하지만 계산량이 증가하는 단점을 가지고 있다. Schmidt 칼만 필터의 경우는 일반적인 칼만 필터에 비하여 안정도와 강인성이 증가하는 장점을 가지고 있다. 그러나 일반적인 칼만 필터보다 많은 스택과 계산 시간을 필요로 한다. UD칼만필터의 경우는 스택의 사용량과 계산량에서 큰 장점을 가지고 있고 또한 정밀도에서 다른 필터에 비하여 2배의 정밀도를 가지고 필터의 안정도도 강하다. 하지만 최소자승법에 비해 계산량이 증가하는 단점이 있다.

4-2 정점실험

UD필터로 구현된 칼만 필터의 성능을 확인하기 위하여 건국대학교 공과대학 옥상에 설치된 안테나를 이용하여 데이터를 수신하여 후처리하였고 또한 GPS수신기 보드인 Architect에 칼만 필터를 구현하였다. 그림 4는 5차 선형 칼만필터에 정지한 사용자를 모델하여 처리한 결과이다. 칼만 필터의 모델이 정확하기 때문에 필터의 정확도가 최소자승법보

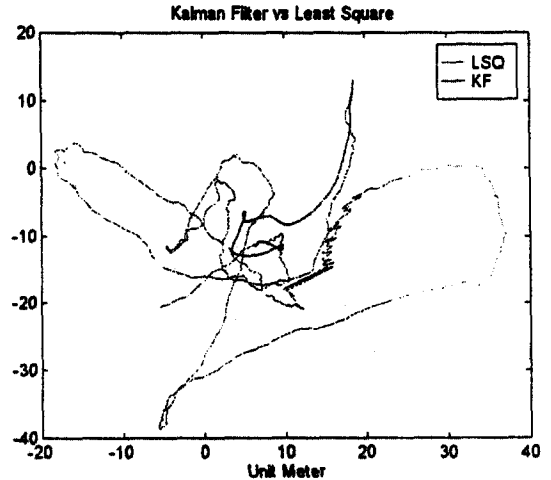


그림 4. 칼만필터와 최소자승법 비교
Fig. 4. Kalman filter vs. least square method.

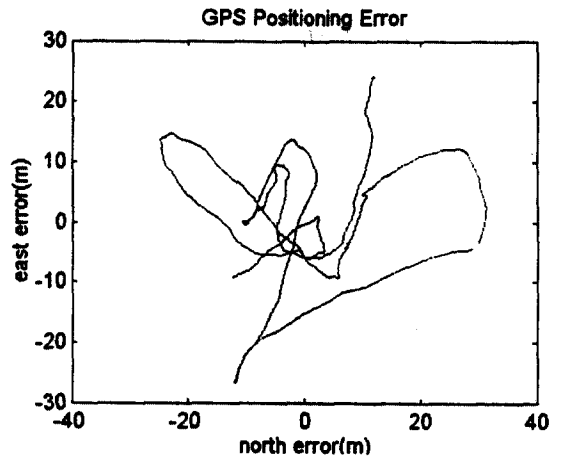


그림 5. 칼만필터 위치해(정점)
Fig. 5. Position using Kalman filter.

다 우수한 결과를 나타낸다. 그림 5는 같은 데이터를 8차의 UD칼만필터를 이용하여 처리한 결과이다. 결과를 살펴보면 칼만필터에 의한 위치해가 최소자승법에 의한 해에 비해 보다 부드러운 궤적을 그림을 알 수 있다. 그림 6은 Architect의 실시간 동작 결과이다. 결과를 살펴보면 설계된 필터가 일반적으로 알려진 GPS오차 범위 내에서 동작하는 것을 알 수 있다.

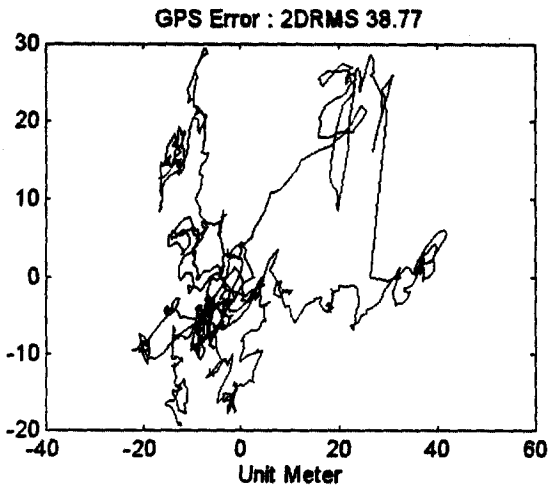


그림 6. Architect 위치해 (정점)
Fig. 6. Position using architect receiver (static).

4-3 동적실험

동적 환경에서의 성능을 알아보기 위하여 정점에서와 마찬가지로 데이터를 받아서 후처리로 성능을 알아봤다. 그림 7은 실험 중 보인 위성의 개수이다. 실험 중 건물, 주변의 차량, 가로수 등의 영향으로

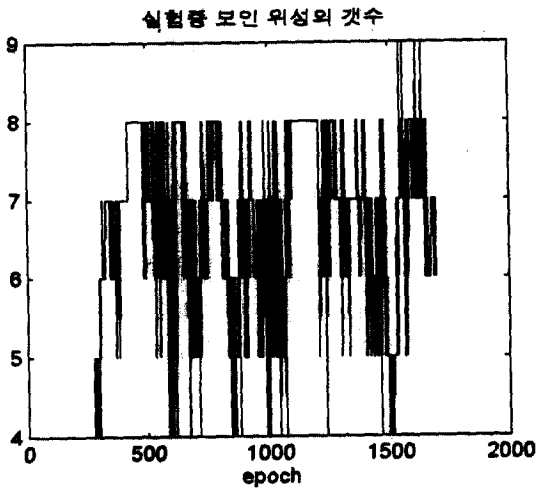


그림 7. 동적 실험중 보인 위성의 개수
Fig. 7. Number of satellite during dynamic experiment.

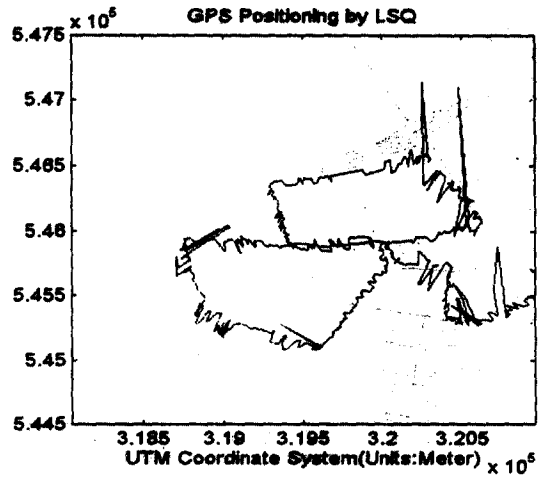


그림 8. 최소자승법 위치해
Fig. 8. Position using least square method.

실험 중 위성이 심하게 변하게 되는 것을 알 수 있다. 그림 8의 최소자승법에 의한 결과를 살펴보면 위치해가 심하게 흔들리는 것을 알 수 있다. 이러한 현상의 주요한 원인은 위치필터에 사용된 위성 집합이 급작스럽게 변하기 때문이다. 그림 10은 같은 데이터를 UD칼만필터를 이용하여 처리한 결과이다. 결과를 살펴보면 최소자승법에 의한 결과보다 우수한 성능을 나타내는 것을 알 수 있다. 그러나 그림 11의 칼만 필터를 이용하여 구한 속도정보가

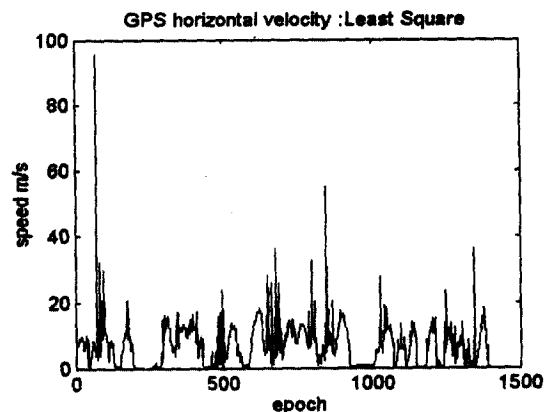


그림 9. 최소자승법 속도
Fig. 9. Velocity using least square method.

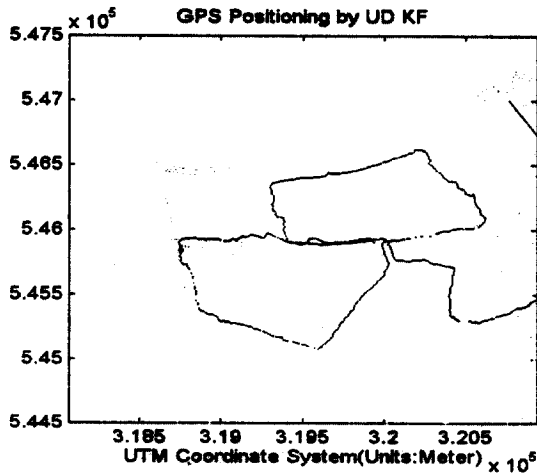


그림 10. UD 칼만 필터 위치해
Fig. 10. Position using UD Kalman filter.

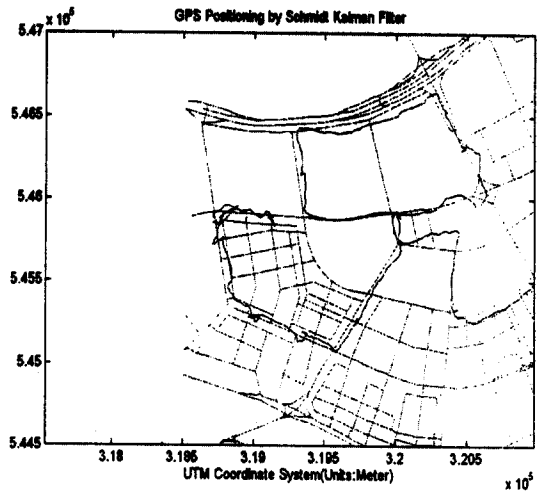


그림 12. Schmidt 칼만 필터 위치해
Fig. 12. Position using Schmidt Kalman filter.

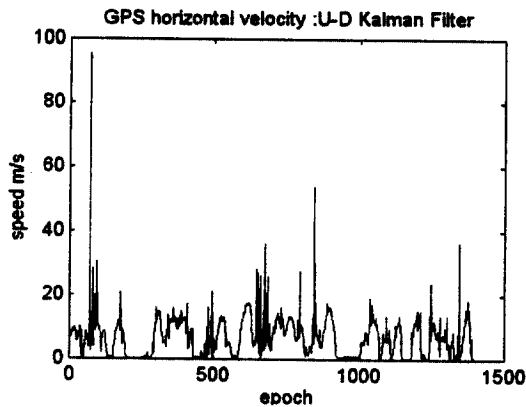


그림 11. UD 칼만 필터 속도
Fig. 11. Velocity using UD Kalman filter.

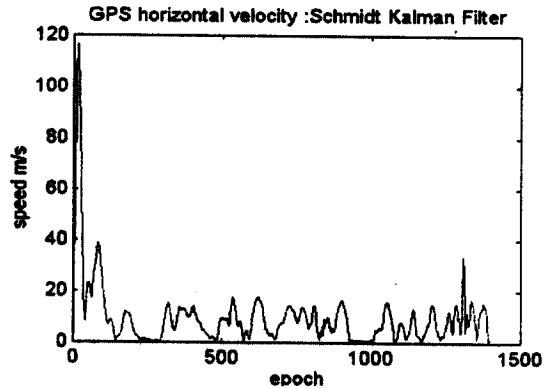


그림 13. Schmidt 칼만 필터 속도
Fig. 13. Velocity using Schmidt Kalman filter.

그림 9의 최소자승법의 결과와 같이 심하게 흔들리는 것을 알 수 있다. 이것은 위성집합이 변하는 순간 S/A등의 영향을 고려하지 못했기 때문이다. 그러나 그림 13의 Schmidt 칼만 필터를 이용한 속도 정보는 이러한 부분을 고려했기 때문에 급격한 속도 변화가 줄어 든 것을 알 수 있다. 그림 14의 정상 상태 칼만 필터 기법을 사용한 위치해를 살펴보면 일반적인 칼만필터를 사용할 때에 비하여 성능이 크게 나빠지지 않는 것을 알 수 있다. 이상의 결과

에서 수신기에서 고려할 수 있는 좋은 방법은 U-D 칼만 필터와 Schmidt칼만 필터를 결합한 후 정상 상태 기법을 추가하는 것임을 알 수 있다.

Architect 보드에 UD칼만 필터를 구현하여 실시간 주행을 한 결과는 그림 15와 그림 16이다. 설계된 알고리즘이 실제 수신기에서 정상적으로 동작하는 것을 알 수 있다.

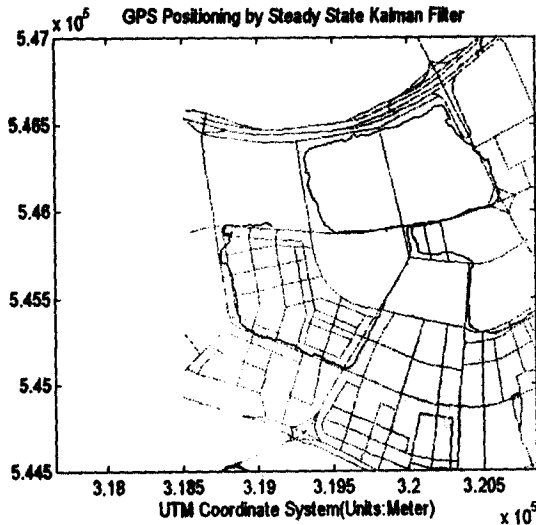


그림 14. Steady-State 칼만 필터 위치해
Fig. 14. Position using Steady-State Kalman filter.

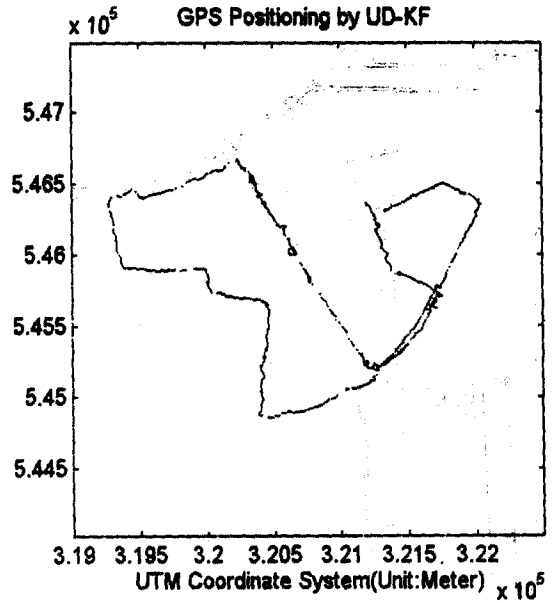


그림 16. UD 칼만 필터 위치해(실시간 동적 환경)
Fig. 16. Position using UD Kalman filter.

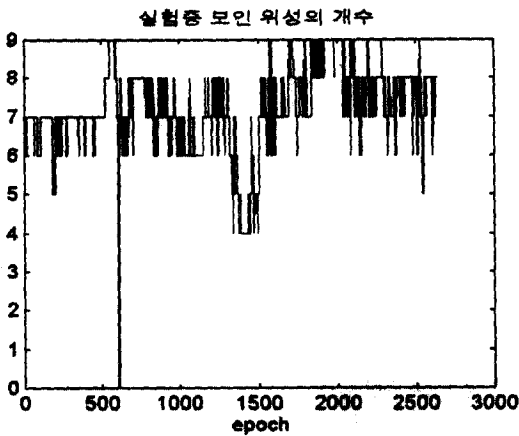


그림 15. Architect실시간 실험 중 보인 위성의 개수
Fig. 15. Number of Satellite during real-time experiment.

V. 결 론

본 논문에서는 GPS 수신기를 위한 칼만필터를 설계하였고 이를 실제 수신기에 구현하였다. 실제로 알고리즘을 구현할 때 생기는 여러 문제를 극복할 수 있는 필터 기술과 필터의 성능을 살펴본 후

GPS 수신기 보드인 Architect보드에 설계된 칼만 필터를 구현하였다. 구현된 알고리즘이 정점 및 동적인 환경에서도 효율적이고 안정적으로 동작하는 것을 확인하였다.

참고문헌

- [1] *Global Positioning System*, vol. I, II, III, IV, The Insitute of Navigation.
- [2] B. Hofmann-Wellenhof, *Global Positioning System Theory and Practice*, Springer-Verlag, New York.
- [3] Peter S. Maybeck, *Stochastic models, estimation, and control*, vol. 1, Navtech Book and Software Store, 1994.
- [4] Elliott. D. Kaplan, *Understanding GPS Principles and Applications*, Artech House, 1996.
- [5] Robert Grover Brown, Patrick Y. C. Hwang, *Introduction to Random Signals*

and Applied Kalman Filtering, John Wiley & Sons, 1985.

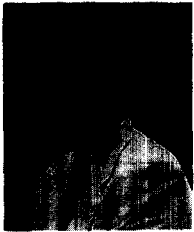
[6] C. K. Chui, G. Chen, *Kalman Filtering with Real-Time Applications*, Springer-Verlag, New York, 1987.

[7] Paul W. McBurney, "A Robust Approach To Reliable Real-Time Kalman Filtering,"

Proceeding of IEEE PLANS90, March 20-23,1990, pp. 549-556.

[8] Spiro P. Karatsinides, "Enhancing Filter Robustness in Cascaded GPS-INS Integrations," *IEEE Transactions on Aerospace and Electronics System*, vol. 30, no. 4, October, 1994.

지 규 인 (池圭仁)



1959년 11월 24일생.
 1982년 서울대 공대 제어계측공학과 학사.
 1984년 서울대 대학원 제어계측공학과 석사.
 1989년 Case Western Reserve University 시스템공학과 박사.

1990년 Rensselaer Polytechnic Institute 연구원.
 1990년 9월~1992년 2월 강원대학교 제어계측공학과 전임강사.
 1992년 3월 ~ 현재 건국대학교 전자공학과 부교수
 관심분야 : GPS/GLONASS 위성항법시스템, GPS/INS 항법시스템, 차량항법시스템, 자동주행차량 항법 및 제어

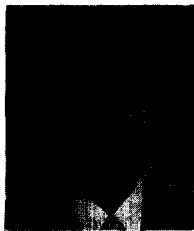
홍 진 석 (洪鎭錫)



1970년 9월 5일생.
 1994년 건국대 전자공학과 학사
 1996년 건국대 전자공학과 석사
 1996년 ~ 현재 건국대학교 전자공학과 박사과정.
 관심분야 : GPS 수신기, GPS 자

세측정 시스템, GPS/INS 항법시스템, 개인휴대 항법 시스템

이 장 규 (李章揆)



1946년 3월 28일생.
 1971년 서울대 공대 전기공학과 졸업
 1974년 Univ. of Pittsburgh 전기공학과 졸업(공학석사)
 1977년 Univ. of Pittsburgh 전기공학과 졸업(공학박사)

1977년 4월 ~ 1981년 6월 The Analytic Sciences Corporation 연구원(Technical Staff)
 1981년 6월 ~ 1982년 7월 Charles Stark Draper Laboratory 연구원(Technical Staff)
 1982년 9월 ~ 현재 서울대 공대 제어계측공학과 교수
 1994년 12월 ~ 현재 서울대 자동제어특화연구센터 소장

오 종 택 (吳宗澤)



1963년 11월 4일생
 1986년 2월 한양대학교 전자통신공학과 학사
 1989년 2월 한국과학기술원 전기 및 전자공학과 석사
 1993년 2월 한국과학기술원 전기

및 전자공학과 박사
 1995년 5월 ~ 현재 한국전기통신공사 연구개발원 무선통신개발단 선임연구원
 관심분야 : 신호처리분야, 이동통신관련분야