GPS/INS Integration using Vector Delay Lock Loop Processing Technique

Hyun-Soo Kim*, Sung-Chun Bu** and Gyu-In Jee***

* Department of Electronics Engineering, Konkuk University, Seoul, Korea
(Tel : +82-2-452-7407; E-mail: drhskim@konkuk.ac.kr)*
**Department of Electronics Engineering, Konkuk University, Seoul, Korea
(Tel : +82-2-452-7407; E-mail: scoolde@hotmail.com)
*** Department of Electronics Engineering, Konkuk University, Seoul, Korea
(Tel : +82-2-450-3070; E-mail: gijee@konkuk.ac.kr)

Abstract: Conventional DLLs estimate the delay times of satellite signals individually and feed back these measurements to the VCO independently. But VDLL estimates delay times and user position directly and then estimate the feedback term for VCO using the estimated position changes. In this process, input measurements are treated as vectors and these vectors are used for navigation. First advantage of VDLL is that noise is reduced in all of the tracking channels making them less likely to enter the nonlinear region and fall below threshold. Second is that VDLL can operate successfully when the conventional independent parallel DLL approach fails completely. It means that VDLL receiver can get enough total signal power to track successfully to obtain accurate position estimates under the same conditions where the signal strength from each individual satellite is so low or week that none of the individual scalar DLL can remain in lock when operating independently. To operate VDLL successfully, it needs to know the initial user dynamics and position and prevents total system from the divergence. The suggested integration method is to use the inertial navigation system to provide initial dynamics for VDLL and to maintain total system stable. We designed the GPS/INS integrated navigation system. This new type of integrated system contained the vector pseudorange format generation block, VDLL signal processing block, position estimation block and the conversion block from position change to delay time feedback term aided by INS.

Keywords: GPS, Inertial Navigation System, delay lock loop, Vector DLL

1. INTRODUCTION

There are various types of integrating inertial sensors with GPS. One of these is the loosely coupled integration that the filtered navigation solution tends to have a higher bandwidth and better noise characteristics than the GPS solution alone. This configuration is best implemented with higher quality inertial sensors if the GPS outages are long in duration.

A more complex level of coupling is tight integration, where GPS PR, Doppler, or carrier phase measurements are filtered with the navigation solution generated by the inertial sensors. In addition to the benefits of loose coupling, a tightly integrated system can have a more accurate navigation solution because the basic GPS observable used in the filtering process are not as correlated as the position and velocity solutions used in loose integration.

A more complex and potentially most beneficial level of GPS-inertial integration occurs at the GPS tracking-loops level. This type of coupling is referred to as ultra-tight integration. This configuration is more complex than the other architectures discussed above because it changes the structure of the traditional GPS tracking loops. In terms of performance, ultra-tight integration also offers the most benefits in terms of accuracy and robustness improvements to the GPS receiver and overall system.

Ultra-tight integration can improve acquisition time as well as the tracking performance of the delay-lock loop in terms of dynamics stress and noise rejection, thus producing more accurate Doppler and pseudo-range measurements.

A conventional GPS receiver consists of several parallel scalar DLLs, each of which independently estimates the individual pseudoranges. When the signal strength from each individual satellite is so low, none of the individual scalar DLL can remain in lock when operating independently. As we know, all pseudoranges between all satellites and a user position change simultaneously whenever satellites move or user moves. In addition, current GPS receiver cannot track the week signals of satellites in bad environments. It means that GPS receiver cannot measure pseudoranges and cannot provide a position. So as to use the correlation of each satellite signal and user dynamics, we use vector delay-lock loop technique [1]. The advantage of VDLL is that noise is reduced in all of the tracking channels making them less likely to enter the nonlinear region and fall below threshold. Second is that VDLL can operate successfully when the conventional independent parallel DLL approach fails completely. It means that VDLL receiver can get enough total signal power to track successfully to obtain accurate position estimates under the same conditions where the signal strength from each individual satellite is so low or week that none of the individual scalar DLL can remain in lock when operating independently.

In this paper, we suggested the new tightly coupled GPS/INS integration to improve the performance of GPS receiver aided by INS and vector delay lock loop (VDLL) technique. As a result, the proposed integration system estimated delay time accurately and power of tracked signal is increased by VDLL technique. In a week signal or unstable environment like as indoors and dense urban area, the new system tracked satellite signals and estimated user position accurately when conventional GPS system could not estimate delay time and failed to fix a user position. In this paper, we will present all of these results compared with the results of the conventional GPS receiver.

2. SIGNAL PROCESSING IN GPS RECEIVER

2.1 GPS Signal Processing

The process of receiving GPS signals may be divided into three steps. First is the acquisition. The signal acquisition process consists of a three-dimensional search in time (code phase), frequency and satellite-specific PRN code. Next step is the tracking. After the acquisition process is accomplished the receiver enters the tracking mode. A delay-lock loop (DLL) is used to track the code phase (or frequency) and a phase-lock loop (PLL) or a frequency-lock loop (FLL) is used to track the carrier phase or frequency, respectively. Last step is the data decoding and position solution. When both tracking loops are in lock, it is possible to decode the 50 bps data and determine the receiver's position.

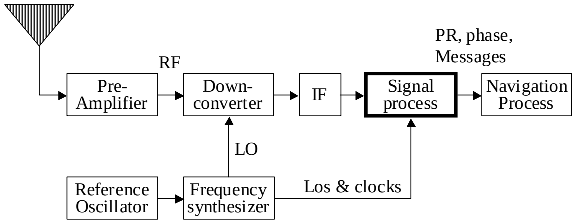


Fig. 1 GPS receiver block diagram

The RF signal from satellite is first down-converted to the IF signal and IF signal is then down-converted by the carrier frequency to the base-band in I and Q components. The I and Q multiplied by the early, late and prompt PRN codes to get the correlation values. The sampling model of prompt I and Q signal are represented below.

$$I_P = \sqrt{2P}M_E R(\tau) D_k \sin(\pi\Delta f T + \Delta\phi)\,\mathrm{sinc}(\Delta f T)$$
$$Q_P = \sqrt{2P}M_E R(\tau) D_k \cos(\pi\Delta f T + \Delta\phi)\,\mathrm{sinc}(\Delta f T) \quad (1)$$

where

P : Power of input signal
M_E : Number of I & Q samples in 1 ms
$R(\tau)$: Auto-correlation function
D_k : Discriminator output
Δf : Carrier frequency changes
$\Delta\phi$: Carrier phase changes

The values are summed and averaged over 20 code periods or 20ms. The correlation values are then fed into the discriminator algorithm that gives the early/late relationship to the code loop filter, which determines if there should be a phase change in the local prompt PRN code.

From the viewpoint of positioning, the process of GPS receiver can be divided into two steps: first is to estimate each of the delay time of satellite signal, second is to estimate the user position with the estimated delay time, shown in Fig. 2.

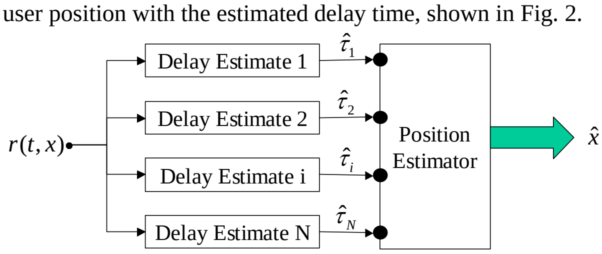


Fig. 2 Conventional GPS position estimate process

That is, a generalized form of GPS position estimate processors has independent parallel estimators for each τ_i.

2.2 Conventional Delay-lock loop

The function of the DLL is to track the CA code component of the GPS signal. In addition to generating the prompt code needed for tracking the carrier in the PLL, the code phase from the code generator is also used to determine the pseudoranges for position determination.

Delay-lock loop consist of a discriminator, loop filter, and code NCO (or DCO). Fig. 3 shows the non-linear model of delay lock loop.

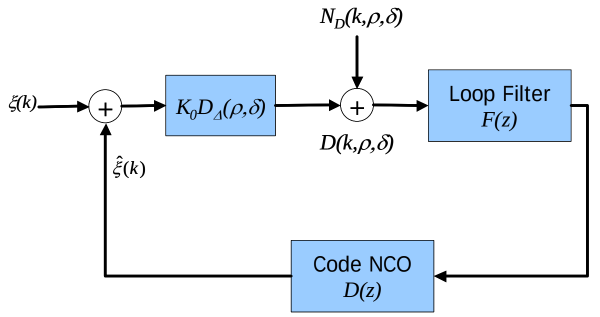


Fig. 3 Conventional DLL block diagram

DLL Discriminator

There are several types of discriminator: early minus late power, dot-product power, cross-product power. Below equation represents the non-coherent early minus late type. In equation, early and late is usually 1/2 chip.

$$\begin{aligned}\Delta_{DLL}(\tau) &= \left[(I_E^2+Q_E^2)-(I_L^2+Q_L^2)\right] \\ &= PM_E^2\,\mathrm{sinc}^2[\pi\Delta f T]\left[R^2(\tau-\frac{d}{2})-R^2(\tau+\frac{d}{2})\right] \\ &= K_{DLL}D_{DLL}(\tau)\end{aligned} \quad (2)$$

where

I_E, Q_E : Early I and Q signal
I_L, Q_L : Late I and Q signal
K_{DLL} : Gain of Discriminator
$D_{DLL}(\tau)$: Discriminator characteristics
d : Chip space between early and late signal

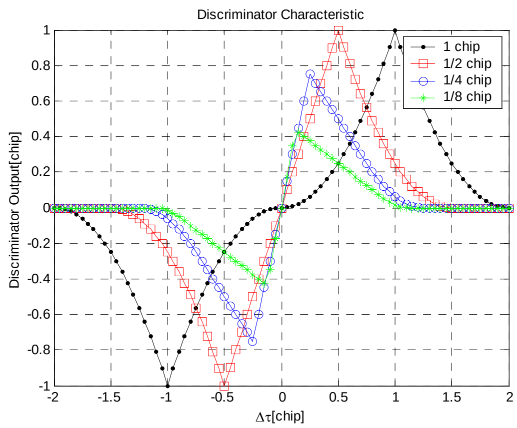


Fig. 4 Discriminator Characteristics

DLL Loop Filter

The transfer function F(z) represents the loop filter of a tracking loop. The loop filter is usually a low-pass filter used to suppress the noise and high-frequency signal components from the phase-detector/delay discriminator and provide a DC-controlled signal for NCO. A 1st order tracking loop is obtained when F(z) = 1. There are usually three options for selecting the loop filters in a 2nd order loop: a lag filter, an active filter or a passive filter. A 2nd order loop with lag filter is usually referred to as a modified 1st order loop rather than a genuine 2nd order loop. A 3rd orders are insensitive to acceleration, and optimal for constant jerk (rate of change of acceleration) input. 4th order loops are insensitive to jerk. It is rare that a loop is constructed with an order higher than third.

The loop order is sensitive to the same order of dynamics. The 1st order loop is suitable for a user position that varies in a random walk manner (white noise velocity); the 2nd order loop is suitable for a user velocity that varies in a random walk manner (white noise acceleration); the 3rd order loop is suitable for a user acceleration that varies in a random walk manner (white noise jerk).

2nd order closed-loop filter is below. The open loop transfer function F(s) in the 's' domain:

$$\frac{y(s)}{x(s)} = F(s) = \frac{(T_2 s+1)}{T_1 s+\alpha} \qquad \alpha = \begin{cases} 1, & passive \\ 0, & active \end{cases} \qquad (3)$$
$$= \frac{a\omega_n s + \omega_n^2}{Ks}$$

And the 2nd order closed-loop transfer function in the 's' domain and the discrete time domain is below.

$$H(s) = \frac{a\omega_n s + \omega_n^2}{s^2 + a\omega_n s + \omega_n^2}$$
$$y_k = a_0 x_k + a_1 x_{k-1} - b_1 y_{k-1} \qquad (4)$$

where $a_0 = \dfrac{T_s + 2T_2}{\alpha T_s + 2T_1},\ a_1 = \dfrac{T_s - 2T_2}{\alpha T_s + 2T_1},\ b_1 = \dfrac{\alpha T_s - 2T_1}{\alpha T_s + 2T_1},$

$\omega_n = \sqrt{K_{DLL} K_{DCO} / T_1}$: Natural frequency

$\zeta = (T_2/2)\omega_n$: Damping ratio

$B_L = \omega_n (1+4\zeta^2)/(8\zeta)$: Closed loop bandwidth

Performance of DLL

The rule of thumb for evaluating the performance of DLL is to see the dynamics stress error and thermal noise error in addition to the tracking error in range.

Generally, thermal noise error can be calculated with the equation:

$$\sigma_{tDLL} = T_C \sqrt{\frac{B_L}{2C/N_0}\left(1 + \frac{2}{T_C C/N_0}\right)} \quad (meters) \qquad (5)$$

where

T_C : Time interval of a chip

C/No : Carrier-to noise ratio in dB

B_L : Loop bandwidth

The necessary condition for better performance is to have lower bandwidth and higher C/No.

Another factor of performance of tracking loop is to evaluate the dynamic stress error. The dynamic stress error is related with the loop filter.

If we used the 2nd order loop filter, the equation of dynamic stress error is below.

$$R(t) = \frac{1}{\omega_L^2}\tau^{(2)} \qquad (6)$$

$\tau^{(2)}$ represents the second derivatives of delay time and ω_L^2 is the square of the loop frequency.

Then the rule of thumb is that the sum of the thermal noise error and the dynamic stress error must be less than the given chip space.

$$3\sigma_{DLL} = 3\sigma_{tDLL} + R_e \le d \quad (chips)$$

3. VDLL PROCESSING TECHNIQUE

3.1 Concept of Vector Delay-Lock Loop

There are two forms of generalized GPS position estimate processors. The first form first estimates the delay using independent parallel estimators for each τ_i and then estimates position as a separate process. This form has the conventional DLL with independent delay estimation processors. The second form of processor estimates the position directly without an independent intermediate delay estimate. This estimator may also produce an estimate of delay, because with GPS a delay estimate is needed for recovery of the navigation data.

Assume for the moment that the user position is an unknown constant vector and that the received signal samples at discrete times t_k are $r(t_k, x)$ and are represented over the time interval $t_1, t_2, ..., t_k$ as the following received vector:

$$r(x) = \left[r(t_1, x), r(t_2, x), ...,\ r(t_N, x)\right] \qquad (7)$$

Then the estimation can be represented as following equation and figure:

$$\hat{x}_a = F_a[r(x)] \qquad (8)$$

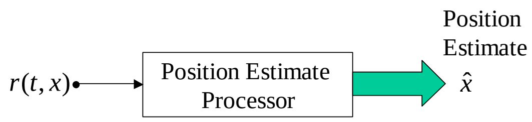


Fig. 5 VDLL Concept

The received baseband signal summed all satellites signals is the function of user position directly.

As we know, not all of their delay measurements are truly independent because the geometry of the satellite-user paths depends on the motion of satellite and user. That is, delay time of each channel is changed simultaneously as user or satellite is moving and is correlated with other channel dynamics.

This means that all of delay times have to be estimated in a single vector-processing loop, the vector DLL. This loop contains the delay time estimation processor for all satellites in view using the estimated position rate vector.

3.2 Vector Delay Lock Loop Processing

The VDLL is a further generalization of the extended Kalman filter (EKF) and is a quasi-optimal extension of the EKF estimation process. The received signal is a single scalar observable, and each satellite-received signal component $s_i\left[t - \tau_i(x(t))\right]$ has two layered non-linearities; namely, $s_i[\tau_i]$ and $\tau_i(x_k)$ and not just a single non-linearity. The scalar observable signal at baseband is as follows:

$$r_k = \sum_{i=1}^{N} a_i s_i\left[k - \tau_i(x_k)\right] + n_k \qquad (9)$$

where the noise samples are independent and Gaussian with zero mean. The objective is to perform a quasi-optimal modified extended Kalman estimate of a vector user position x_k from this scalar observable. To derive the VDLL configuration, it can be written the Taylor's series expansion for the signal from satellite *i* as follows:

$$s_i\left[t-\tau_i(x(t))\right] = s_i(t-\tau_i(\hat{x})) + s_i'(t-\tau_i(\hat{x}))g_i^T(t)(x-\hat{x})+,... \quad (10)$$

where

$$g_i^T \triangleq \left(\frac{\partial\tau_i}{\partial x}, \frac{\partial\tau_i}{\partial y}, \frac{\partial\tau_i}{\partial z}, \frac{\partial\tau_i}{\partial B}\right),$$

$$\frac{\partial s_i}{\partial x} = s_i'\left[t-\tau_i(x)\right]\frac{\partial\tau_i}{\partial x}, \quad s_i'(t-\tau_i) = \frac{\partial s_i}{\partial\tau_i}(t-\tau_i)$$

The vector $g(t)$ has components that vary slowly with time as the satellite-user geometry changes. From the product of the received signal $r(t)$ with each of N differentiated signal reference waveforms $s_i'\left[t-\tau_j(\hat{x})\right]$, to obtain the observation vector.

$$v(t) = \begin{bmatrix} r(t)s_1'(t-\hat{\tau}_1) \\ r(t)s_2'(t-\hat{\tau}_2) \\ \cdots \\ \cdots \\ r(t)s_N'(t-\hat{\tau}_N) \end{bmatrix} = \begin{bmatrix} a_1 D(\Delta\tau_1) \\ a_2 D(\Delta\tau_2) \\ \cdots \\ \cdots \\ a_N D(\Delta\tau_N) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ \cdots \\ \cdots \\ n_N(t) \end{bmatrix} \cong \boldsymbol{A}P_d \begin{bmatrix} \Delta\tau_1 \\ \Delta\tau_2 \\ \cdots \\ \cdots \\ \Delta\tau_N \end{bmatrix} + \boldsymbol{n}$$

where

$$\Delta\tau_i = \tau_i - \hat{\tau}_i = \Delta\tau_i(x)$$
$$D(\Delta\tau_i) = E\left[s(t)s'(t+\Delta\tau_i)\right] = R_s'(\Delta\tau_i) \cong R_s''(0)\Delta\tau \quad for \ \ \Delta\tau \ll T$$

Use the Jacobian geometric matrix $\partial\tau_i/\partial x_j = G$ to obtain an approximation for above equation and the following linearized matrix equation is produced.

$$v(t) = \begin{bmatrix} r(t)s_1'(t-\hat{\tau}_1) \\ r(t)s_2'(t-\hat{\tau}_2) \\ \cdots \\ \cdots \\ r(t)s_N'(t-\hat{\tau}_N) \end{bmatrix} \cong \boldsymbol{A_k}P_d \begin{bmatrix} g_1(t_k) \\ g_2(t_k) \\ \cdots \\ \cdots \\ g_N(t_k) \end{bmatrix} \varepsilon + \boldsymbol{n}+,...,$$

$$\cong \boldsymbol{A_k}P_d \begin{bmatrix} \frac{\partial\tau_1}{\partial x} & \frac{\partial\tau_1}{\partial y} & \frac{\partial\tau_1}{\partial z} & \frac{\partial\tau_1}{\partial b} \\ \frac{\partial\tau_2}{\partial x} & \frac{\partial\tau_2}{\partial y} & \frac{\partial\tau_2}{\partial z} & \frac{\partial\tau_2}{\partial b} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial\tau_N}{\partial x} & \frac{\partial\tau_N}{\partial y} & \frac{\partial\tau_N}{\partial z} & \frac{\partial\tau_N}{\partial b} \end{bmatrix} \begin{bmatrix} x-\hat{x} \\ y-\hat{y} \\ z-\hat{z} \\ b-\hat{b} \end{bmatrix} + n \quad (12)$$

where ε $\cong P_d \boldsymbol{A_k} G(t_k)\varepsilon_k + n_k$ is the user position vector error. And rewrite above equation in discrete time notation as follows;

$$v(t) = \begin{bmatrix} r(t)s_1'(t-\hat{\tau}_1) \\ r(t)s_2'(t-\hat{\tau}_2) \\ \cdots \\ \cdots \\ r(t)s_N'(t-\hat{\tau}_N) \end{bmatrix} \cong P_d \boldsymbol{A_k} G(t_k)[x_k - \hat{x}_{k/k-1}] + n_k \quad (13)$$

where the noise samples are independent in time, and the noise has a covariance matrix $E[n_k^T n_j] = R_k \delta_{jk}$.

This equation is now in the form where we can easily generate the standard Kalman estimator by assuming the following process and measurement equation:

$$\begin{aligned} x_k &= F_k x_{k-1} + w_k \\ v_k &= P_d A_k G_k [x_k - \hat{x}_{k/k-1}] + n \\ &= H_k x_k + n_k - H_k \hat{x}_{k/k-1} \end{aligned} \quad (14)$$

where we define $P_d A_k G_k = H_k$.

The Kalman estimator equation can then be written as follows.

Predictor : $\hat{x}_{k+1/k} = F_k \hat{x}_{k/k}$

Correction update : $\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k\left[v_k - H_k \hat{x}_{k/k-1}\right]$

Kalman gain : $K_k = P_{k/k-1} + H_k^T\left[R_k + H_k P_{k/k-1} H_k^T\right]^{-1}$

The NCOs act as the integrators in the VDLL.

There are several advantages of this VDLL receiver if the number of satellites exceeds the number of dimensions to be estimated. In principle, the use of signals from N>4 satellites in parallel may provide enough total signal power to track successfully and to obtain accurate position estimates using the VDLL receiver under same conditions where the signal strength from each individual satellite is so low that none of the individual conventional (scalar) DLL can remain in lock when operating independently.

There are variations in this vector delay lock receiver depending on the dimensionality of the user state vector, and the dimensionality of the measurements; e.g., pseudoranges or pseudoranges plus Doppler/rate of change measurements on the carrier and other sensor inputs.

4. GPS/INS INTEGRATION ALGORITHM

In this paper, the suggested integration of GPS/INS has two modes: first mode is to integrate in the level of GPS tracking loop (ultra tightly coupled), second mode is to integrate in the level pseudorange rate and velocity estimation (tightly coupled). The tightly coupled integration is used in compensating the INS error using GPS pseudorange rate of satellites and user velocity.

As the tightly coupled is well-known method, we will focus on the ultra tightly coupled method that Doppler frequency changes by user dynamics is estimated from INS and the corrected Doppler estimation in tracking loop is aided by the estimated INS Doppler, and this processes is performed using the vector-processing technique. The algorithm is depicted in following figure.

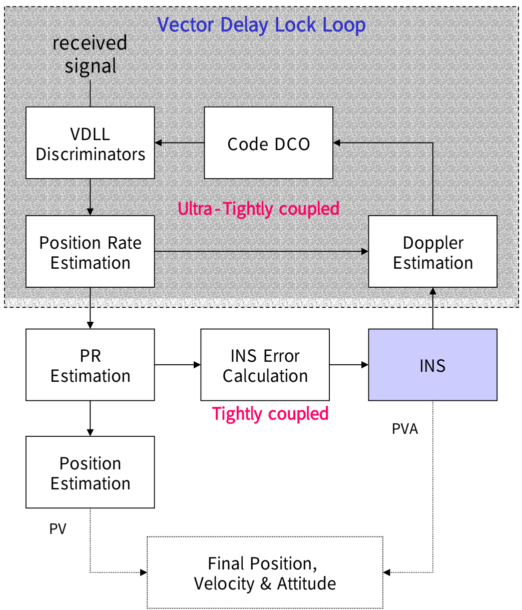


Fig. 6 Proposed Integration Scheme

4.1 Doppler Frequency Estimation from INS

The Doppler frequency of the carrier signal can be expressed simply as the velocity of the receiver relative to the transmitter, projected onto the line-of-sight (LOS) vector.

The relationship is expressed in equation.

$$f_{dopp} = \frac{1}{\lambda}\left(\vec{V}^{RX} - \vec{V}^{S}\right)\vec{1}^{S} \tag{15}$$

where

λ = Wavelength of carrier L1 frequency

$\vec{V}^{RX}$ = Velocity of receiver antenna

$\vec{V}^{S}$ = Velocity of satellite

$\vec{1}^{S}$ = Unit LOS vector from receiver to satellite

The components of this equation can be computed in any Earth-fixed reference frame, such as Earth-Centered Earth-Fixed (ECEF) or local East-North-Up (ENU). The transformation matrix between these two reference frames is easily computed as a function of latitude and longitude.

The changes by Doppler Effect in C/A code can be calculated using the wavelength of C/A code frequency.

Since the inertial sensors provide accelerations in a body-fixed frame, and attitude reference is needed to transform them to an Earth-fixed frame.

4.2 Doppler Frequency Estimation Filter

The accuracy of the Doppler estimates will obviously depend on the quality of the inertial sensors used. Using the equation above, the Doppler measurements can be modeled by following equation. The line-of-sight (LOS) vector from receiver to satellite is made by the linearized transition matrix.

$$\begin{aligned} f_{dopp} &= \frac{1}{\lambda}H^{S}\Delta\left(V_{R-S}^{T}\right) + n = f_{dopp}^{T} + n \\ f_{dopp} - f_{dopp}^{T} &= \frac{1}{\lambda}H^{S}\Delta\left(V_{R-S}\right) - \frac{1}{\lambda}H^{S}\Delta\left(V_{R-S}^{T}\right) \\ &= \frac{1}{\lambda}H^{S}\left[\Delta V_{R-S} - \Delta V_{R-S}^{T}\right] \\ &= \Delta f_{dopp} \end{aligned} \tag{16}$$

The Doppler component from the output of discriminators in VDLL is calculated by this equation:

$$\Delta\tau_{disc.}^{S} = \frac{D_{DLL}(\tau)}{K_{DLL}},\ \Delta f_{Dopp}^{disc.} = F(\Delta\tau_{disc.}^{S}) \tag{17}$$

The difference between two Doppler estimates is the input of measurement equation of Doppler estimation Kalman filter.

4.3 INS-aided GPS Tracking Loop

The designed INS-aided GPS tracking loop is represented in following figure. This model consists in the vector delay estimation block, the inertial navigation algorithm block, the navigation filter, and the Doppler estimation filter. The navigation filter perform the position rate estimation for the vector delay lock loop, convert the position rate estimation into the delta delay time of each satellite, and feed forward to the loop filter to estimate the code DCO Doppler and frequency. The final roll of the navigation filter is to estimate user position using the INS aided information.

The difference between this new designed tracking loop and generic tracking loop is that the navigation cycle is very fast rather than the generic GPS receiver and the code DCO command is generated by the Doppler estimation filter, then the output of discriminator does not put into the code DCO.

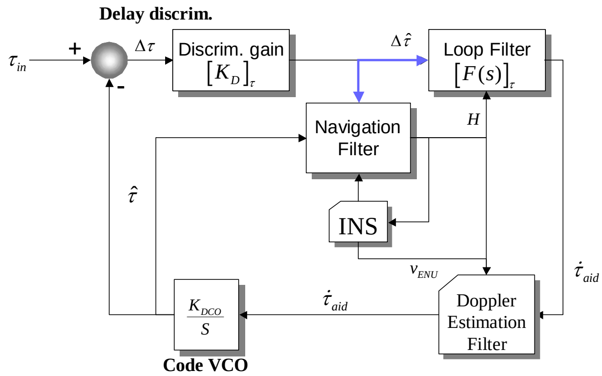


Fig. 7 INS-aided GPS Tracking Loop

The expected advantages of the INS-aided GPS tracking loop is summarized below.

- Can get the increased signal-to-noise ratio
- Can reduce the tracking loop jitter error
- Can track the high-dynamic signal caused by user's fast motion

In this paper, the proposed algorithm to integrate GPS/INS is simulated by ideal GPS simulation signal and INS toolbox.

The following chapter is shared for the representation of simulation results of the algorithm proposed and designed.

5. PERFORMANCE ANALYSIS

5.1 Generic DLL vs. Vector DLL

First of all, the performance of VDLL was evaluated by simulation of weak signal environment. When one of satellite signals is weak, the generic GPS receiver with independent parallel DLLs cannot remain in lock the weak signal. However, the vector DLL can track the weak signal continuously and can estimate the delay time and can use for positioning.

The following figures represented this situation. First case is that the GPS receiver received a weak satellite signal. Second case is that the GPS receiver received two weak satellite signals.

CASE I:

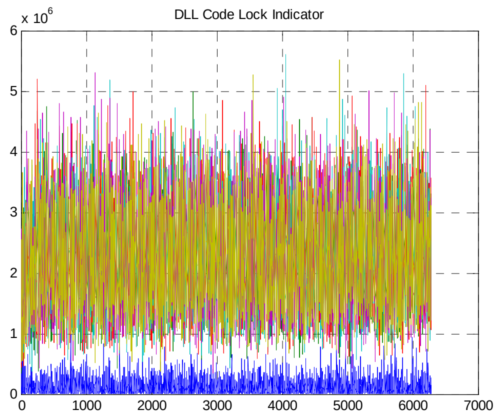
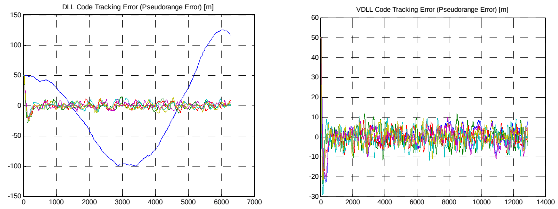


Fig. 8 Case I : Code Lock Indicator with 1 weak signal



(a) Generic DLL (b) Vector DLL

Fig. 9 Code Tracking Error Comparison of Case I: In generic DLL, a weak signal cannot remain the lock. However, VDLL is remaining the lock for all signals.

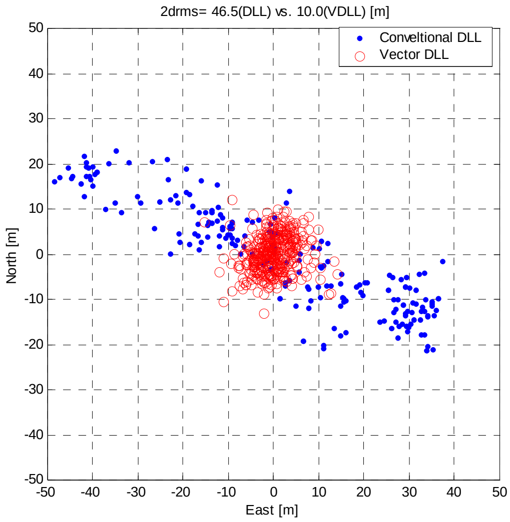


Fig. 10 Positioning Error Comparison of Case I

CASE II:

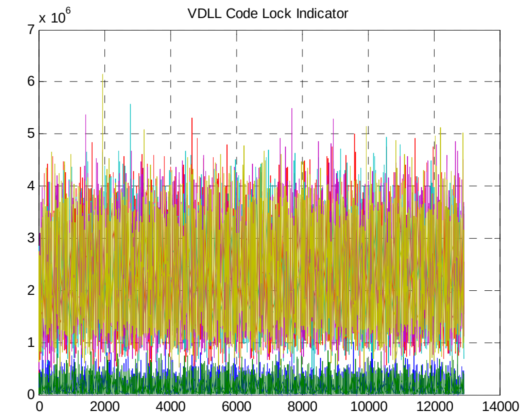
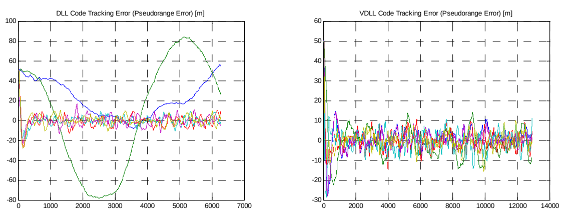


Fig. 11 Case II : Code Lock Indicator with 2 weak signals



(a) Generic DLL (b) Vector DLL

Fig. 12 Code Tracking Error Comparison of Case II

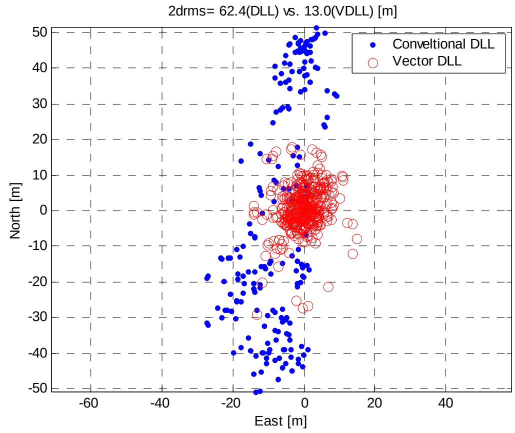


Fig. 13 Positioning Error Comparison of Case II

We can see that the positioning error of VDLL is less than that of generic DLL in weak signal environment. Most important fact is that although the signal-to-noise ratio of weak signals is 30dB less than that of other signals, VDLL can track these weak signals normally.

5.2 GPS/INS Integration

To evaluate the performance of INS-aided GPS tracking loop, we generated the INS information by simulation. If user has the motion of the following figure, then it is possible to estimate the Doppler estimation by multiplying the LOS vector from user to satellites.

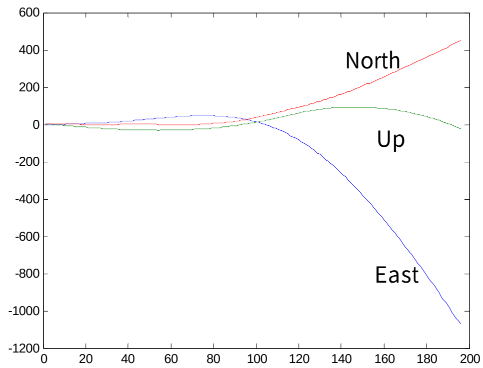


Fig. 14 User Dynamics Information from INS

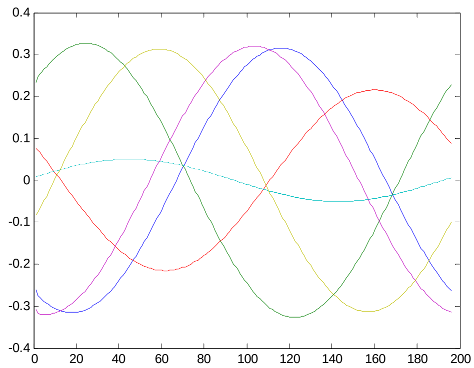
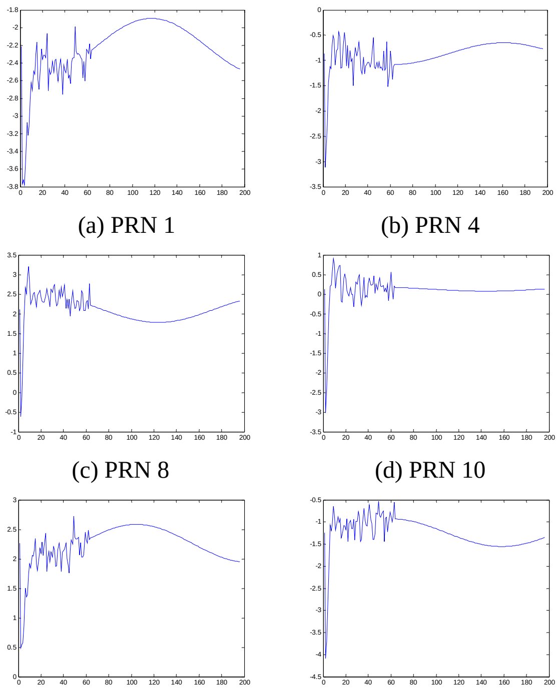


Fig. 15 Estimated Doppler Frequencies for each Satellite

The following figures represented the results of INS-aided effect when INS aiding information is added in the processing of the vector delay lock tracking loop.



(a) PRN 1 (b) PRN 4

(c) PRN 8 (d) PRN 10

(e) PRN 14 (f) PRN 22

Fig. 16 Integrated Estimated Doppler using Kalman Filter

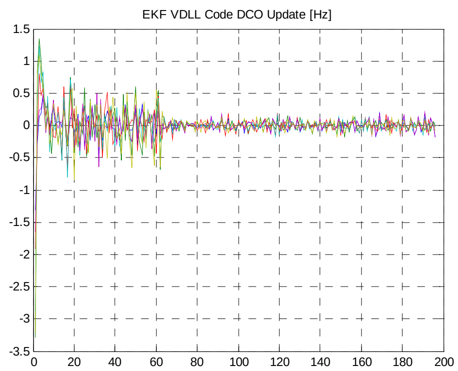


Fig. 17 VDLL Code DCO Update Estimation

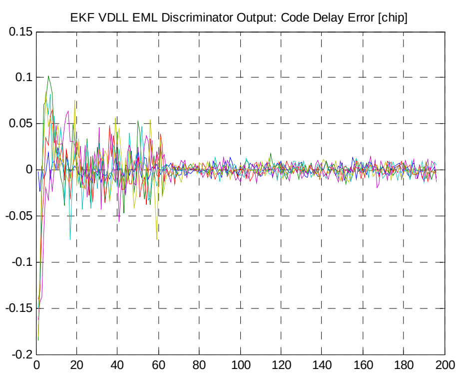


Fig. 18 VDLL Early minus Late Discriminator Output

6. CONCLUSIONS

The vector delay lock loop processing technique has the advantages that it can operate with momentary blockage of one or more satellites, that is, the remaining satellites may be able to give a sufficiently good position estimate that there is never any loss of lock in any of the delay lock loops. However, when user has high dynamic motion, VDLL also stressed in tracking the fast changeable satellite signal caused by user motion. In this paper, we suggested the INS-aided GPS tracking loop contained the VDLL and INS.

The performance of the proposed algorithm was evaluated by simulation. These results of simulation represent that the proposed algorithm can reduce the tracking jitter, positioning error and improve the positioning ability of GPS receiver.

If this algorithm has the complete INS in the GPS receiver, then this GPS receiver can provide user position and velocity information anywhere, anytime, continuously.

Further work is to implement this algorithm in real-time GPS receiver.

ACKNOWLEDGMENTS

This work is supported by KISTEP National Research Laboratory program. The authors wish to thank for their technical and financial supports.

REFERENCES

[1] Bradford W. Parkinson and James J. Spilker Jr., *Global Positioning System: Theory and Applications Volume 1 & II*, AIAA, Washington, 1996.

[2] Elliott D. Kaplan, *Understanding GPS Principles and Applications*, Artech House, London, 1996.

[3] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering 3rd Edit.*, John Wiley & Sons, Boston, 1997.

[4] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, IEE Radar, Sonar, Navigation and Avionics Series 5, 1997.

[5] S. Alban, D. M. Akos S. M. Rock and D. Gebre-Egziabher, "Performance Analysis and Architectures for INS-aided GPS Tracking Loops" *ION-NTM 2003*, pp. 611-622, 2003.

[6] D. –J. Jwo, "Optimisation and sensitivity analysis of GPS receiver tracking loops in dynamic environments," *IEE Proc.-Radar, Sonar Navig.*, pp. 241-250, 2001.

[7] GPS Builder Designer's Guide, GEC Plessey Semiconductors, GPS Group, Wiltshire, United Kingdom, Nov. 1994.