GPS Receiver Tracking Loop Design using H_∞ control Theory

***Gyu-In Jee, Jin Seok Hong, Whi Kim, Sang Hyun Kim**
GPS System Lab., Konkuk Univ.
(*Dept. of Geomatics Engineering, University of Calgary)

Chan Gook Park
GNC Lab. Kwangwoon Univ.

Biography

Gyu-In Jee is an Associate Professor in the Department of Electronics Engineering at Konkuk University in Seoul, Korea. He received his Ph.D. in Systems Engineering from Case Western Reserve University. For last 7 years, his research has been focused on GPS and navigation system. He has worked on several research projects; GPS receiver software design, GPS/INS integration for land vehicle, DGPS system development, GPS engine design using the Mitel chip sets, wireless location in CDMA network, etc. He is currently on Sabbatical leaving at the university of Calgary.

Jin Seok Hong is in the doctoral course at the Dept. of Electronics Eng. in Konkuk University. His research interests are software GPS receiver, GPS receiver navigation algorithm, GPS attitude system and GPS/INS integrated system.

Whi Kim is in the master's course at Dept. of Electronic Engineering in Konkuk University. His research interests include Internet based DGPS system and Software GPS receiver.

Sang Hyun Kim is a graduate research assistant at GPS system lab. He received the B.Sc. degrees in Department of Electronics Engineering at Konkuk University, Korea in 2000.

Chan Gook Park has worked as an associate professor in department of control and instrumentation Engineering at Kwangwoon University, Seoul, Korea, since 1994. His research interests include Kalman filtering, inertial navigation systems, and GPS/INS integration.

Abstract

The heart of GPS receiver is tracking loop and navigation algorithm. The delay lock loop is a well-known technique to track the pseudo-noise codes for spread spectrum system. This paper presents a noncoherent square law DLL and provides a linear model of it. And the loop filter is designed using H infinity control theory. And in order to analyze the performance of designed loop filter, the nonlinear Monte Carlo simulation of the GPS receiver under the combined dynamic and signal-to-noise ratio (SNR) conditions is performed.

Introduction

The Navstar GPS(Global Positing System) is a satellite-based, world wide, all-weather navigation and timing system. The heart of GPS receivers is tracking loop and navigation algorithm. The tracking algorithm enables continually generating replica code that occurs maximum code correlation in the receiver to decode navigation message from GPS satellite and to generate pseudorange measurement. In order to work navigation algorithm properly, the measurements that depend on the tracking loop are needed. In the case that tacking loop can't track incoming GPS satellite signal the measurement cannot be generated [1,2,3].

The tracking loop is accomplished using PLL (Phase Lock Loop) techniques and most tracking loop filter is designed based on the linear model of PLL and classic control theory. The tracking loop exhibits the characteristic that, if the tracking error exceeds a certain boundary, the tracking loop will no longer be able to track the signal and will lose lock. This is because the characteristics of code and carrier tracking loops discriminators are nonlinear, especially near the threshold regions.

This paper presents a noncoherent square law DLL and provides a linear model of it. And the loop filter is designed using H infinity control theory. Sometimes GPS receiver is in trouble with bad RF environment such as high foliage and high jamming. In this case, we need the algorithm that put up with the bad circumstances. The H-infinity controller makes the system have robustness in the worst-case situation.

Noncoherent DLL Tracking Loop

Generally the code tracking loop of GPS receiver is composed as figure 1. The code tracking loop can be divided into coherent tracking loop and noncoherent tracking loop. In order to use coherent tracking loop the receiver must generate exact carrier signal that matches with input signal. Since GPS receiver typically operate at very low signal-to-noise ratios the generation of this coherent reference at low signal-to-noise ratios is difficult. Another difficulty of coherent demodulation is that the carrier signal is modulated by data. Therefore general GPS receiver uses noncoherent delay lock loop.

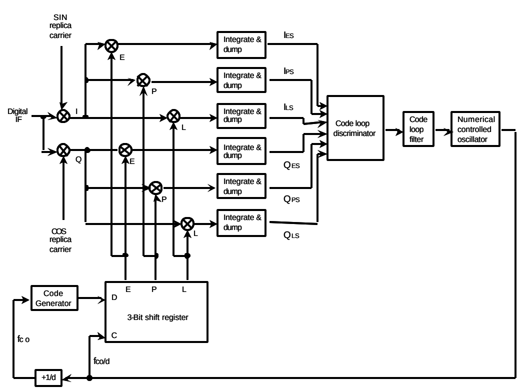

Fig. 1. Structure of Code Tracking Loop

I & Q Model

The received signal is modeled as the sum of three signals : the propagation delayed signal transmitted by the GPS satellite, the jamming signals, and receiver thermal noise effects.

$$s(t) = D(t-t)C(t-t)\sqrt{2P(t-t)}\sin[w_c t + q_d(t)] + J_n(t)\cos[w_c t + q_n(t)] + n_T(t) \quad (1)$$

where,

$D(t-t)$: data modulation
$C(t-t)$: code modulation
$P(t-t)$: signal power
w_c : carrier frequency
t : code phase delay
$q_d(t)$: Doppler phase shift
$J_n(t)$: jamming amplitude
$q_n(t)$: jamming phase
$n_T(t)$: thermal noise

The jamming amplitude $J_n(t)$ is modeled as a zero mean first order Gauss-Markov process with autocorrelation given by

$$\Omega_n(t) = 2Je^{-\frac{|t|}{a}} \quad (2)$$

The jamming phase $q_n(t)$ is a random constant that is uniformly distributed between $-\pi/2$ and $\pi/2$. The jamming to signal power ratio, expressed in dB, is defined by

$$J/S = 10\log_{10}\frac{J}{P} \quad (3)$$

The thermal noise is treated as white Gaussian noise with spectral intensity of

$$2N_0 = k_b T_{eq} \quad (4)$$

where $k_b = 1.37\times10^{-23} J/K$ is Boltzmann's constant, and T_{eq} is the receiver equivalent temperature. Sampling model of input signal to derive I & Q model, excluding noise, is

$$\sqrt{2P}C_k D_k \sin((w_{IF} + \Delta w)t_k + f_0) \quad (5)$$

The internal replica signal model about Inphase and Quadrature Phase is as follows :

$$AC_{r,k}\sin(w_{IF} + f_r) \quad (6)$$

$$AC_{r,k}\cos(w_{IF}t_k + f_r) \quad (7)$$

Therefore correlation output of Inphase is

$$\begin{aligned} &\sqrt{2P}AC_kC_{r,k}D_k\sin((w_{IF}+\Delta w)t_k + f_0)\sin(w_{IF}t_k + f_r) \\ &= \frac{\sqrt{2P}}{2}AC_kC_{k,r}D_k\cos(\Delta w t_k + \Delta f) \end{aligned} \quad (8)$$

And correlation output of Quadrature Phase is

$$\begin{aligned} &\sqrt{2P}AC_kC_{r,k}D_k\sin((w_{IF}+\Delta w)t_k + f_0)\cos(w_{IF}t_k + f_r) \\ &= \frac{\sqrt{2P}}{2}AC_kC_{k,r}D_k\sin(\Delta w t_k + \Delta f) \end{aligned} \quad (9)$$

If we set A=2, the average of accumulator output for I and

Q channel early, prompt and late can be computed as

$$I_P = \sqrt{2P} M_E R(t) D_k \sin(p\Delta fT + \Delta f)\mathrm{sinc}(\Delta fT) \quad (10)$$

$$Q_P = \sqrt{2P} M_E R(t) D_k \cos(p\Delta fT + \Delta f)\mathrm{sinc}(\Delta fT) \quad (11)$$

$$I_E = \sqrt{2P} M_E R(t - \frac{d}{2}) D_k \sin(p\Delta fT + \Delta f)\mathrm{sinc}(\Delta fT) \quad (12)$$

$$Q_E = \sqrt{2P} M_E R(t - \frac{d}{2}) D_k \cos(p\Delta fT + \Delta f)\mathrm{sinc}(\Delta fT) \quad (13)$$

$$I_L = \sqrt{2P} M_E R(t + \frac{d}{2}) D_k \sin(p\Delta fT + \Delta f)\mathrm{sinc}(\Delta fT) \quad (14)$$

$$Q_L = \sqrt{2P} M_E R(t + \frac{d}{2}) D_k \cos(p\Delta fT + \Delta f)\mathrm{sinc}(\Delta fT) \quad (15)$$

where

$R(t)$: autocorrelation function of PN code

M_E : number of sample during accumulation

Noncoherent delay lock loop discriminator

The structure of noncoherent Square-Law DLL is shown in Figure 2. The received signal was made low appropriate IF through the mixing process after pass the RF band-pass filter $H_1(s)$. IF signal was divided into Inphase and Quadrature Phase, pass correlator after sampling, then process discriminator for DLL. Discriminator square signal and detect delay error using envelope after excluding data modulation and carrier phase error.

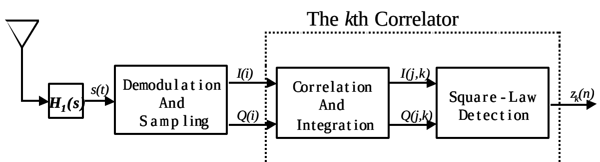

Fig. 2. Noncoherent Square-Law Discriminator

The derive of model coefficient in case of use of noncoherent DLL discriminator using the power difference between Early code and Late code is

$$\begin{aligned}\Delta_{DLL}(t) &= \left[(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)\right] \\ &= K_0 D_{DLL}(t)\end{aligned} \quad (16)$$

If the frequency error is small, discriminator gain K_0 is as follows :

$$\begin{aligned}K_{DLL} &= 4PM_E^2 \,\mathrm{sinc}^2(p\Delta fT) \\ &\cong 4PM_E^2\end{aligned} \quad (17)$$

In this case, distance between the early code and late code, d, is one chip, and $D_\Delta(t)$ means output of the discriminator that slopes is one at zero point. Noise signal variance of the discriminator output composed of output average is given by

$$s_{DLL}^2 = \frac{8N_{\mathrm{var}}^2 + 4PN_{\mathrm{var}} M_E^2}{N_{EML}} \quad (18)$$

Noncoherent DLL Model

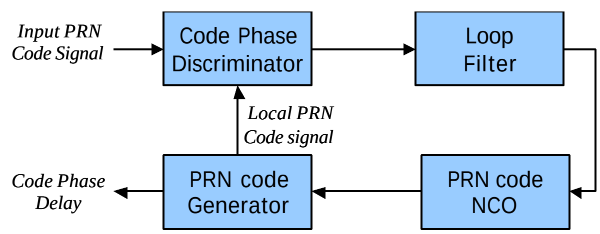

Fig. 3. DLL Block Diagram

After signal acquisition process, accurate code lock is accomplished by DLL. Fig. 3 shows the block diagrams of DLL. The code phase discriminator detects the code phase error between incoming signal and replica signal. Because the output of code phase discriminator has the noise component, it passes through the loop filter and is used for input to code generator. The objective of DLL tracking loop filter design is to reduce noise in order to make small RMS jitter. The performance of tracking loop is estimated by the variance of DCO input signal. The Signal that passed the discriminator takes on the following form,

$$e_e(k) = K_0 D_\Delta(e) + N_D(k) \quad (19)$$

The general nonlinear noncoherent DLL model represented as following

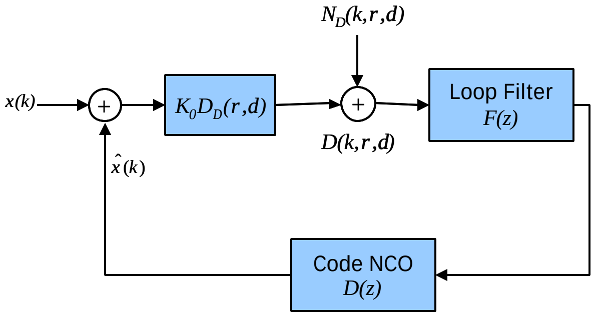

Fig. 4. Nonlinear equivalent circuit for nonlinear noncoherent DLL model

Assuming the discriminator operates on linear space, linear model is derived as follow

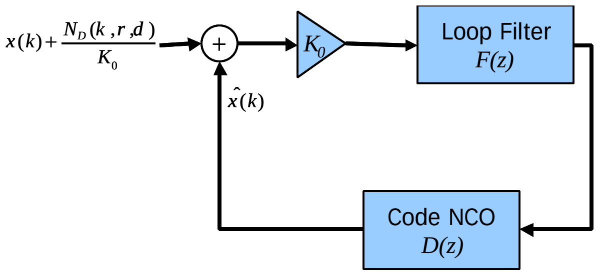

Fig. 5. Linear DLL Model

Where the DCO transfer function is as follows

$$D(z) = K_D \frac{z}{z-1} \tag{20}$$

Therefore, closed loop transfer function is as follow

$$H_x(z) = \frac{\hat{x}(z)}{x(z)} = \frac{PF(z)K_D \frac{z}{z-1}}{1+PF(z)K_D \frac{z}{z-1}} = \frac{PK_D F(z)}{z-1+PK_D F(z)} \tag{21}$$

$$H_e(z) = \frac{e(z)}{x(z)} = \frac{1}{1+PF(z)K_D \frac{z}{z-1}} = \frac{z-1}{z-1+PK_D F(z)} \tag{22}$$

A RMS Tracking Jitter is given as follow,

$$\sigma_x^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} N_{xx}(e^{j\omega})\left|H(e^{j\omega})\right|^2 d\omega \tag{23}$$

Tracking loop design using H_∞ controller

The H_∞ control problem is formulated as follows : consider the two-port diagram in Fig. 6, and find an internally stabilizing controller, $K(s)$, for the plant $P(s)$, such that the ∞-norm of closed loop transfer function is below a given level γ (a positive scalar). This problem is called the standard control H_∞ problem. The optimal H_∞ control problem is to minimize the ∞-norm of some transfer function.

From the derived linear DLL model two port model is induced in order to use H_∞ controller as GPS receiver tracking loop, which shown in Fig. 7.

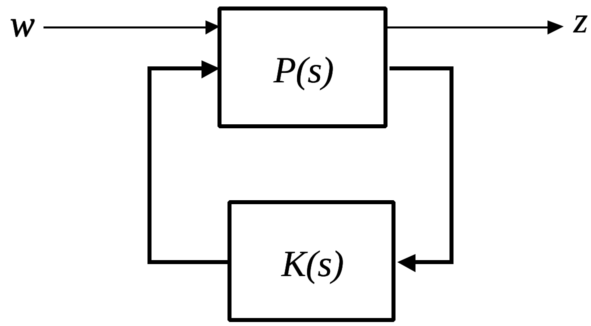

Fig. 6. The two-port block diagram for H_∞ control.

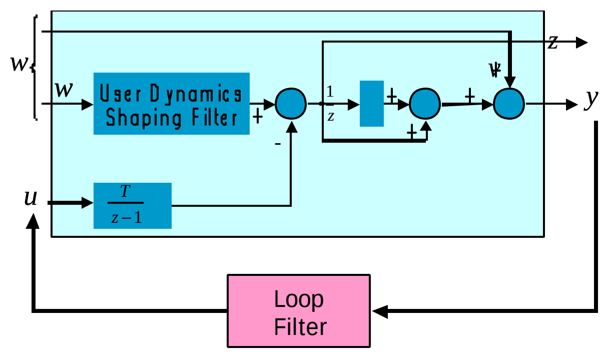

Fig. 7. Tracking loop model for H_∞ controller

System State Equation

For the tracking loop design, two-input, two-output model equation is required. The user dynamics is modeled using shaping filter that the input is unit white noise[1],

$$\begin{bmatrix} p_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} p_n \\ v_n \end{bmatrix} + q\begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \tag{24}$$

$$t_n = \begin{bmatrix} \frac{1}{cT_c} & 0 \end{bmatrix}\begin{bmatrix} p_n \\ v_n \end{bmatrix} \tag{25}$$

Where, c is speed of light and T_c is the code chip length. Multiplying $1/cT_c$ converts the output into code phase unit. The integration of the output of the loop filter to produce the estimate of the code phase delay is realized via the following equation.

$$t_{n+1} = \hat{t}_n + \hat{t}_n \Delta t \qquad (26)$$

The code phase estimation error is the difference between the code phase and the estimate of the code phase

$$e_n = t_n - \hat{t}_n \qquad (27)$$

The equation for the propagation of the code phase estimation error is obtained by combining the previous results.

$$e_{n+1} = e_n + \frac{v_n}{cT_c}\Delta t + \frac{qw_n}{2cT_c}\Delta t^2 - \hat{t}\Delta t \qquad (28)$$

For the improvement of stability of system model, discriminator output is modeled by the mean of current and previous output.

$$y_n = 2\left(\frac{e_{n-1} + e_n}{2}\right) + v_n \qquad (29)$$

Let h_n represent the output of the unit delay. The state space representation of the delay is simply,

$$h_{n+1} = e_n = t_n - \hat{t}_n \qquad (30)$$

The linear model for measurement is

$$y_n = e_n + h_n + s_v v_n \qquad (31)$$

Where s_v^2 is the variance of the post correlation noise v_n.

For the open loop two-input, two-output model the state space can be written as

$$x_n = [v_n \quad e_n \quad h_n]^T \qquad (32)$$

A minimal realization for open loop system model is as following

$$x_{n+1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{\Delta t}{cT_c} & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} x_n + \begin{bmatrix} q\Delta t & 0 \\ \frac{q\Delta t^2}{2cT_c} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} w_n \\ v_n \end{bmatrix} + \begin{bmatrix} 0 \\ -\Delta t \\ 0 \end{bmatrix} u_n \qquad (33)$$

$$e_n = [0 \quad 1 \quad 0]x_n + [0 \quad 0]\begin{bmatrix} w_n \\ v_n \end{bmatrix} + [0]\hat{t}_n \qquad (34)$$

$$y_n = [0 \quad 1 \quad 1]x_n + [0 \quad s_v]\begin{bmatrix} w_n \\ v_n \end{bmatrix} + [0]\hat{t}_n \qquad (35)$$

Nonlinear Simulation Result

The performances of designed tracking loop are analyzed using nonlinear I/Q model simulation. Though the nonlinear simulation could be performed using sampled IF signal model, in this case the needed computation power becomes too excessive to spend cost and time. So I/Q model with 1KHz output rate is used for the simulation. And second order active PI type filter that has 1Hz the bandwidth is used for comparison[2]. The nonlinear simulation flow is shown in Fig. 8.

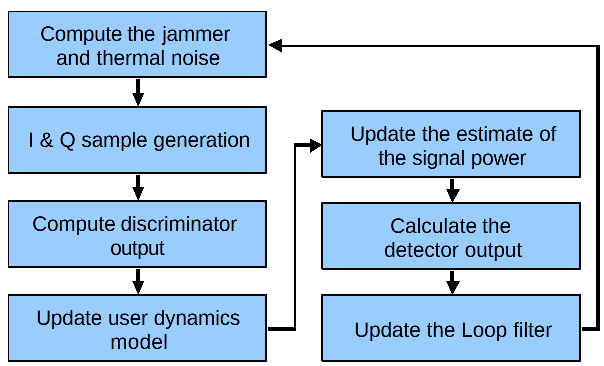

Fig. 8 Nonlinear Simulation Flow

An example code phase trajectory which used in simulation for user dynamics is given in Fig. 9 and Fig. 10, where the unit is converted to range and range rate by multiplying code wavelength to code phase and frequency. The code phase change is the integral of frequency change.

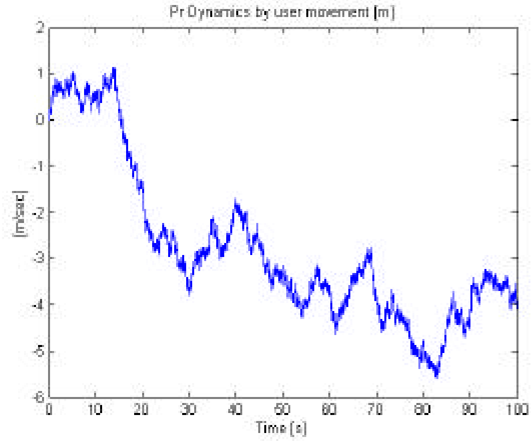

Fig. 9 Example of velocity change by user by user dynamics

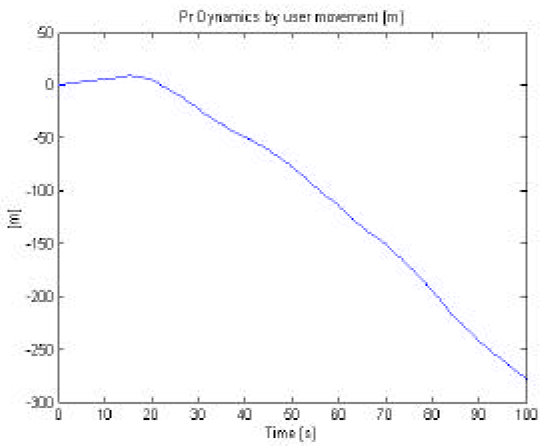

Fig. 10 Example of pseudorange change by user by user dynamics

Characteristics of the Loop Filter

The singular value plots of the filter transfer functions are shown in Fig. 11 and the singular value plots of the closed-loop transfer functions are shown in Fig. 12. In these plots, the PI controller has higher gain than H_∞ controller at overall frequency area but the H_∞ controller has a roll off at high frequency area that regarded as disturbances and unexpected motion of user movement. This characteristic is also shown in singular value plot of closed-loop transfer function.

The transient responses of designed loop filters are shown in Fig. 13 and Fig. 14 for PI controller and H_∞ controller respectively. In this simulation assumes that the user movement is static and the satellite has the constant velocity dynamics. As illustrated in these figures, the H_∞ controller reaches steady state quickly than active PI controller but the tracking error after steady state has larger standard deviation than it of active PI controller.

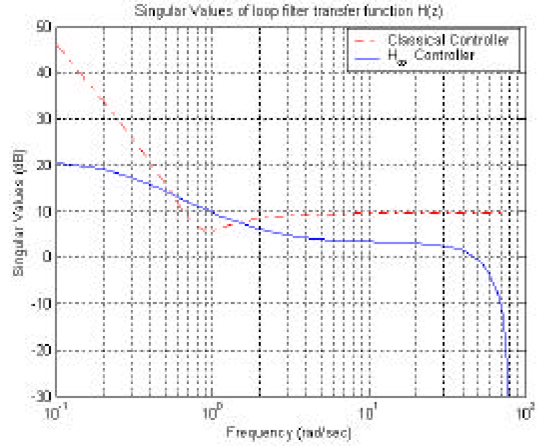

Fig. 11 Singular value plot of loop filter transfer functions

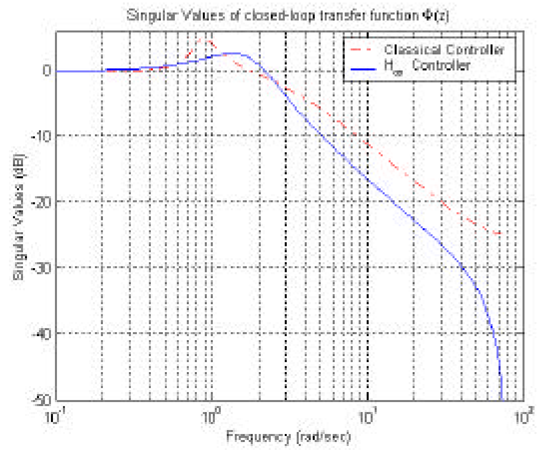

Fig. 12 Singular value plot of closed-loop transfer functions

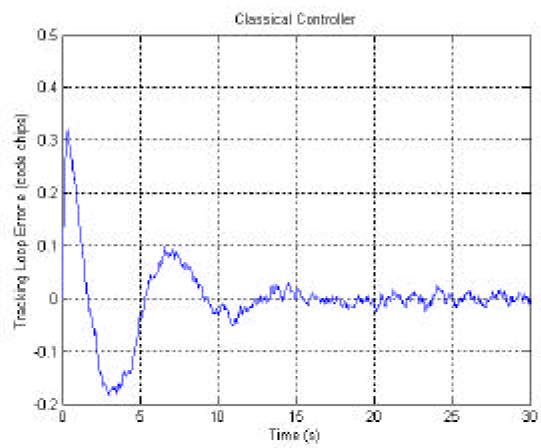

Fig. 13 Code Tracking Loop Error when active PI controller is used.

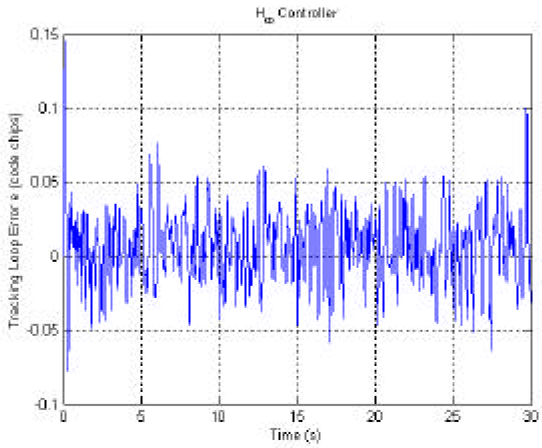

Fig. 14 Code Tracking Loop Error when H∞ controller is used.

Performance Analysis of Tracking Loop

Since the discriminator output influences the overall performance of DLL directly, the noise characteristic of discriminator is important. The analytical output variance of DLL is given by

$$s_{DLL}^{2} = \frac{8N_{var}^{2} + 4PN_{var}M_{E}^{2}}{N_{EML}} \tag{36}$$

In this equation the higher Signal-to-Noise ratio has the higher noise variance but the correspond slope of discriminator is steep the relative noise level is reduced. Examples of simulation results that normalized for 0.5 chip error are as shown in Fig. 15 and Fig. 16. The simulation parameters of Fig. 15 are 20dB jamming-to-signal ratio and 34dB carrier-to-noise ratio and it of Fig. 16 are J/S=10dB and C/N_0=44dB. Generally the lower CNR and the higher JSR make wider envelop of noise variance and finally the tacking loop become break down.

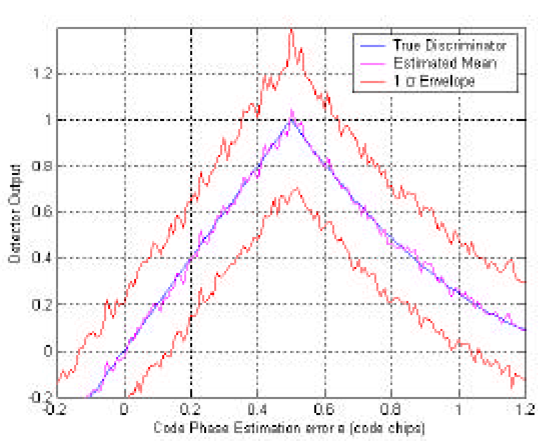

Fig. 15 Discriminator Characteristics: J/S=20dB , C/N_0=34dB

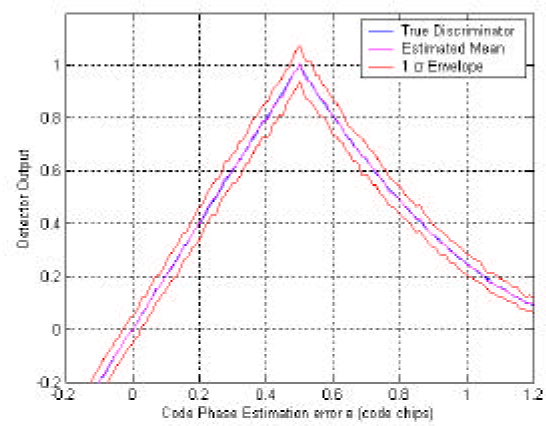

Fig. 16 Discriminator Characteristics: J/S=10dB , C/N_0=44dB

In order to use for benchmark, when the tacking loop performance of dynamic user will be analyzing, the static case simulation is performed with varying CNR. The simulation results are shown as Fig. 17. The active PI controller shows better tracking loop performances above 33dB. But the tracking performance is worse below 34dB and the tracking looses lock at 32dB. The H∞ controller has the 3dB better performance than active PI in point of tracking view.

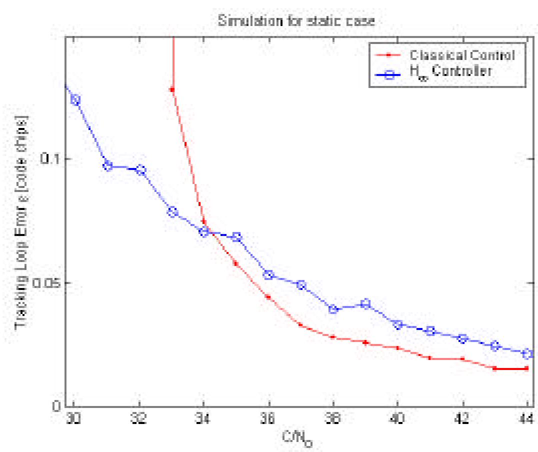

Fig. 17 Tracking loop performances: static user

The simulation results under the user dynamics with varying CNR and varying dynamic gain are shown in Fig. 18 and Fig. 19. Fig. 20 shows the difference of tracking loop performance between 1g and 10g dynamic gain. Although the user dynamic is added to input code phase, the entire performances are maintained and the loosing lock point is the same as static case. This is because the above second-order loop tracks a frequency-modulated signal and returns the phase discriminator to the null point.

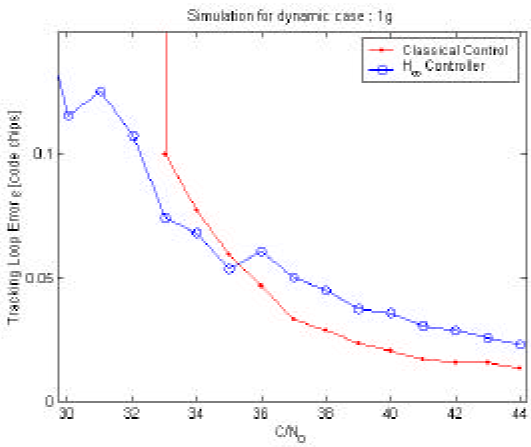

Fig. 18 Tracking loop performances: 1g dynamic gain

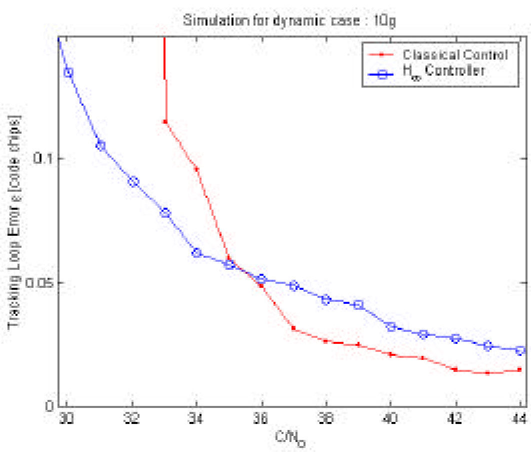

Fig. 19 Tracking loop performances: 10g dynamic gain

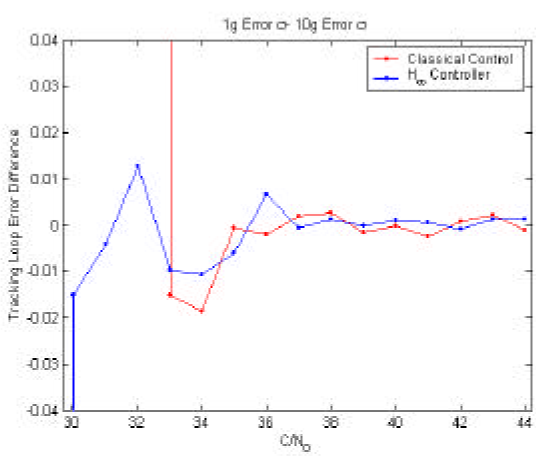

Fig. 20 Tracking Loop Error Difference between 1g and 10g dynamic gain

Fig. 21 shows the tracking loop performances between active PI controller and H_∞ controller with fixed CNR and varying JSR. The JSR changes from 0dB to 40dB. The simulation was performed classifying static and dynamic user and the results of simulation are illustrated in Fig. 21 and Fig. 22 respectively. The H_∞ controller has 3dB better performance than active PI controller. Since the H_∞ controller designed for worst case situation, when unexpected error is added, it shows superior performance.

Conclusion

In this paper, linear model for noncoherent delay lock loop of GPS receiver is represented and the loop filter is designed based on this linear model using classical and modern control theory. The used modern control theory is H-infinity control technique where the two-port model for loop filter design is induced from linear DLL model. And the second-order active PI controller is chosen for comparative classical controller. The performances of designed loop filters are investigated using Monte Carlo simulation with varying CNR, JSR

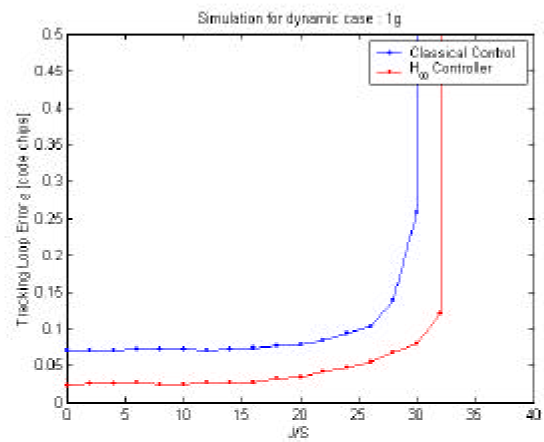

Fig. 21 Tracking Loop performance with varying JSR: Static

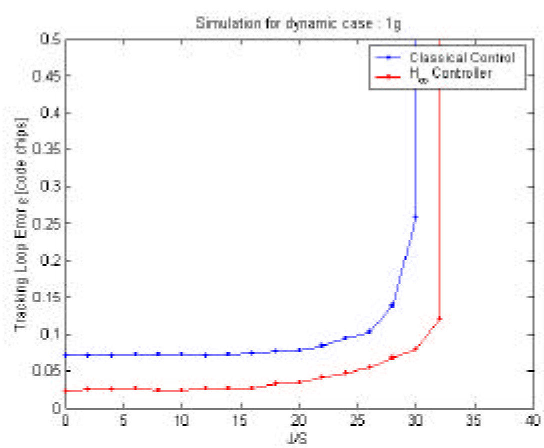

Fig. 22 Tracking Loop performance with varying JSR: Dynamic

and user dynamic conditions. The simulation results show that in normal condition the active PI controller has good performance but in the worst case the H-infinity controller could have better performances. Thus some specific environments where high foliage or high jamming exists, using H-infinity controller for code tracking loop filter is preferred.

References

[1] M. P. Fikes,*GPS Receiver Tracking Loop Optimization Using l_1 Theory*, MS Thesis, MIT, Cambridge, Massachusetts, 1994.

[2] E. D. Kaplan Eds.*Understanding GPS Principles and Application*, Artech House, 1996.

[3] B. Parkinson and J. Spilker, Jr., eds. *Global Positioning System: Theory and Applications I, Progress in Astronautics and Aeronautics*, vol. 163, pp. 129-132, AIAA, Washington, D.C., 1996.