A Convex Optimization Approach to GPS Receiver Tracking Loop Design

HE-SHENG WANG
Department of Communications and Guidance Engineering
National Taiwan Ocean University
Keelung 202, TAIWAN
TEL: +886-2-24622192ext7202, FAX: +886-2-24633492
E-mail: hswang@mail.ntou.edu.tw

Abstract - In this paper, the application of convex optimization theory to GPS (Global Positioning System) receiver tracking loops is investigated. We design an $\mathcal{H}_\infty$ controller for the receiver tracking loop based on an LMI (Linear Matrix Inequality) approach. The $\mathcal{H}_\infty$ controller is particularly attractive because it is a robust design in the sense that small disturbances lead to small tracking errors. Furthermore, it easily accommodates the inclusion of plant uncertainties as part of the plant model. By adding unstructured or structured perturbations to the plant model, it is possible to design controller that ensure stability robustness and performance robustness of the closed-loop system. In order to apply the $\mathcal{H}_\infty$ optimal design, we first rewrite the GPS receiver tracking loop into a two-input two-output generalized plant model, and then the $\mathcal{H}_\infty$ controller is design by using an LMI approach. The LMI is a convex optimization problem in which a local solution is guaranteed to be a global minimizer. By using a software-based GPS L5 signal generator, various levels of disturbances are derived as test inputs to evaluate the performance of the resulting tracking loop.

Key-Words: GPS; Linear Matrix Inequality; Convex Optimization; $\mathcal{H}_\infty$ Control; Delayed-Lock Loop

1. Introduction

A generic GPS receiver can be roughly split into two main functional parts: 1) the signal processing part, and 2) the navigation processing part [1][4][6][7][9][12]. The job of the signal processing is, broadly speaking, to track certain characteristics of the incoming GPS signal in order to produce measurements of the pseudorange and the Doppler shift. The purpose of the navigation processor is to assemble the pseudoranges measured by different channels in order to determine the receiver's inertial position with respect to the WGS84 coordinate system. In addition, the navigation processor assembles the Doppler shift measurements in each of the channels to provide an estimate of the receiver's three dimensional velocities. Usually, the signal processing part runs at high rate, while the navigation processor runs at a much slower rate.

There are three pieces of information typically estimated by the receiver's signal processor: code phase, Doppler shift, and signal power. The estimation of these quantities may or may not be done in an independent manner. Traditionally, code phase estimation is usually accomplished via a delay-locked loop (DLL). DLLs have the characteristic that if the code tracking error exceeds a given threshold, the DLL will lose lock

on the signal and no longer be able to track the signal. Since $\mathcal{H}_\infty$ theory produces closed-loop systems that minimize the worst case amplification from the input to output, it seems natural to apply $\mathcal{H}_\infty$ theory to the code tracking loop problem, with the goal of bounding the tracking loop error and preventing loss-of-lock under adverse dynamic and jamming scenarios.

In the early eighties, Zames [10][11] formulated the $\mathcal{H}_\infty$ problem. In this scenario, the objective is to design a feedback compensator that minimized the maximum closed-loop system gain when mapping from arbitrary square-integrable input sequences to square-integrable output sequences. $\mathcal{H}_\infty$ theory is called a worst case theory because it minimizes the worst case system performance over the class of all allowable input signals. The maximum system gain is the $\mathcal{H}_\infty$ norm of the system which is the peak of the system's singular value plot.

$\mathcal{H}_\infty$ theory is particularly attractive because it easily accommodates the inclusion of plant uncertainties as part of the plant model. By adding unstructured or structured perturbations to the plant model it is possible to design compensators that ensure stability robustness and performance robustness of the closed-loop design.

In this paper, the application of $\mathcal{H}_\infty$ optimal control theory to GPS (Global Positioning System) receiver tracking loops is investigated. We design an $\mathcal{H}_\infty$ controller for the receiver tracking loop. In order to apply the $\mathcal{H}_\infty$ optimal design, we first rewrite the GPS receiver tracking loop into a two-input two-output generalized plant model. By using a software-based GPS L5 signal generator, various levels of disturbances are derived as test inputs to evaluate the performance of the resulting tracking loop.

2. Overview of $\mathcal{H}_\infty$ Control Theory

The standard $\mathcal{H}_\infty$ control problem is conceptually organized as depicted in Figure 1 [2][3]. The general plant is represented by *G*, and *K* represents the feedback controller. Let the plant *G* be described by the following state-space equations:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 w(k) + B_2 u(k), \\ z(k) &= C_1 x(k) + D_{11} w(k) + D_{12} u(k), \\ y(k) &= C_2 x(k) + D_{21} w(k) + D_{22} u(k), \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, and $w \in \mathbb{R}^r$ represents a set of exogenous inputs which includes disturbances to be rejected and/or reference commands to be tracked. A, B_1, B_2, C_1, C_2, D_{11}, D_{12}, D_{21}, and D_{22} are constant matrices with compatible dimensions. The second equation defines a penalty variable $z \in \mathbb{R}^s$, which may include a tracking error, as well as a cost of the inputs *u* and *w* needed to achieve the prescribed control goal. The third equation defines a set of measured variables $y \in \mathbb{R}^p$, which are functions of the state plant *x*, the control input *u*, and the exogenous input *w*.

The control action to (1) is to be provided by a controller which processes the measured variable *y* and generates the appropriate control input *u*, and is modeled by equations of the form

$$\begin{aligned} \xi(k+1) &= A_C \xi(k) + B_C y(k), \\ u(k) &= C_C \xi(k) + D_C y(k) \end{aligned} \tag{2}$$

in which $\xi \in \mathbb{R}^{n_C}$ is the state variables of the controller. A_C, B_C, C_C, and D_C are the controller's parameters to be determined. The purpose of the control is twofold: to achieve closed-loop stability and to attenuate the

influence of the exogenous input w on the penalty variable z. If the plant transfer function matrix $G(z)$ is

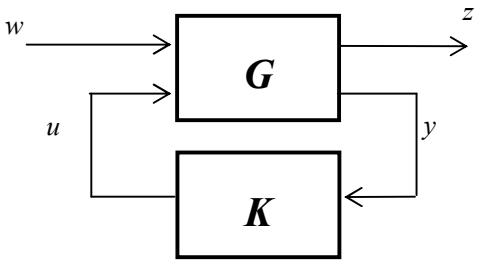

Figure 1. Standard Block Diagram

partitioned into four transfer functions such that

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \tag{3}$$

then the closed-loop map between w and z is given by the following linear fractional transformation:

$$\mathcal{LFT}(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}. \tag{4}$$

The suboptimal $\mathcal{H}_\infty$ control problem of parameter γ consists of finding a controller $K(z)$ such that[1]:

- the closed-loop system is internally stable,
- the $\mathcal{H}_\infty$ norm of $\mathcal{LFT}(G, K)$ (the maximum gain from w to z) is strictly less than γ.

Solutions of this problem (if any) will be called γ-suboptimal controllers. In this section, we introduce the LMI (Linear Matrix Inequality) approach to solve the suboptimal $\mathcal{H}_\infty$ control problem. In this approach, we shall make the following two assumptions:

(A1) (A, B_2, C_2) is stabilizable and detectable,

(A2) $D_{22} = 0$.

The first assumption is necessary and sufficient to allow for plant stabilization by dynamic output feedback. As for **(A2)**, it incurs no loss of generality while considerably simplifying calculations.

Assuming **(A2)** and given any proper real-rational controller $K(z)$ of realization

$$K(z) = D_C + C_C(zI - A_C)^{-1}B_C, \;\; A_C \in \mathbb{R}^{n_C \times n_C} \tag{5}$$

a realization of the closed-loop transfer function from w to z is obtained as:

$$\mathcal{LFT}(G, K) = D_{cl} + C_{cl}(zI - A_{cl})^{-1}B_{cl},$$

where

$$\begin{aligned} A_{cl} &= \begin{bmatrix} A + B_2D_CC_2 & B_2C_C \\ B_CC_2 & A_C \end{bmatrix}, \\ B_{cl} &= \begin{bmatrix} B_1 + B_2D_CD_{21} \\ B_CD_{21} \end{bmatrix}, \\ C_{cl} &= \begin{bmatrix} C_1 + D_{12}D_CC_2 & D_{12}C_C \end{bmatrix}, \\ D_{cl} &= D_{11} + D_{12}D_CD_{21}. \end{aligned} \tag{6}$$

Gathering all controller parameters into single variable

$$\Theta := \begin{bmatrix} A_C & B_C \\ C_C & D_C \end{bmatrix}$$

and introducing the shorthand notations:

$$A_0 = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, B_0 = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, C_0 = \begin{bmatrix} C_1 & 0 \end{bmatrix},$$
$$\mathcal{B} = \begin{bmatrix} 0 & B_2 \\ I & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & I \\ C_2 & 0 \end{bmatrix}, \mathcal{D}_{12} = \begin{bmatrix} 0 & D_{12} \end{bmatrix}, \quad (7)$$
$$\mathcal{D}_{21} = \begin{bmatrix} 0 \\ D_{21} \end{bmatrix}$$

The closed-loop matrices A_{cl}, B_{cl}, C_{cl}, D_{cl} can be written as:

$$A_{cl} = A_0 + \mathcal{B}\Theta C;\ B_{cl} = B_0 + \mathcal{B}\Theta\mathcal{D}_{21};$$
$$C_{cl} = C_0 + \mathcal{D}_{12}\Theta C;\ D_{cl} = D_{11} + \mathcal{D}_{12}\Theta\mathcal{D}_{21}. \quad (8)$$

Note that (7) involves only plant data and that A_{cl}, B_{cl}, C_{cl}, D_{cl} depend affinely on the controller data Θ.

The following lemma plays a central role in the subsequent development.

Lemma 1. *Given a symmetric matrix* $\Psi \in \Re^{m\times m}$ *and two matrices P, Q of column dimension m, consider the problem of finding some matrix* Θ *of compatible dimensions such that*

$$\Psi + P^T\Theta^T Q + Q^T\Theta P < 0. \quad (9)$$

Denote by W_P, W_Q *any matrices whose columns form bases of the null spaces of P and Q, respectively. Then* (9) *is solvable for* Θ *if and only if*

$$\begin{cases} W_P^T\Psi W_P < 0, \\ W_Q^T\Psi W_Q < 0. \end{cases} \quad (10)$$

Proof. For a proof, see [2].

In the following, we recall the Bounded Real Lemma for discrete-time systems. This lemma helps turning the $\mathcal{H}_\infty$ suboptimal constraints into an LMI.

Lemma 2. *Consider a discrete-time transfer function* $T(z)$ *of realization* $P(z) = D + C(zI - A)^{-1}B$. *The following statements are equivalent*:

(*i*) $\left\| D + C(zI - A)^{-1}B \right\|_\infty < 1$ *and A is stable in the discrete-time sense*;

(*ii*) $\inf_{T \text{ invertible}} \sigma_{\max} \begin{bmatrix} TAT^{-1} & TB \\ CT^{-1} & D \end{bmatrix} < 1,$

(*iii*) *there exists* $X = X^T > 0$ *such that*

$$\begin{bmatrix} A^T XA - X & A^T XB & C^T \\ B^T XA & B^T XB - I & D^T \\ C & D & -I \end{bmatrix} < 0,$$

(*iv*) *there exists* $X = X^T > 0$ *such that*

$$\begin{bmatrix} -X^{-1} & A & B & 0 \\ A^T & -X & 0 & C^T \\ B^T & 0 & -I & D^T \\ 0 & C & D & -I \end{bmatrix} < 0. \quad (11)$$

Proof. See, e.g., [2].

Applying the lemma to the realization (6) of the closed-loop system, the controller

$$K(z) = D_C + C_C(zI - A_C)^{-1}B_C, \ A_C \in \mathbb{R}^{n_C\times n_C}$$

is γ-suboptimal if and only if the LMI

$$\begin{bmatrix} -X_{cl}^{-1} & A_{cl} & B_{cl} & 0 \\ A_{cl}^T & -X_{cl} & 0 & C_{cl}^T \\ B_{cl} & 0 & -\gamma I & D_{cl}^T \\ 0 & C_{cl} & D_{cl} & -\gamma I \end{bmatrix} < 0$$

holds for some $X_{cl} > 0$ in $\mathbb{R}^{(n+n_C)\times(n+n_C)}$. With the decomposition (8) and Θ in hand, (11) can be rewritten as follows:

$$\Psi_{X_{cl}} + \mathcal{Q}^T \Theta^T \mathcal{P} + \mathcal{P}^T \Theta \mathcal{Q} < 0, \tag{12}$$

where

$$\Psi_{X_{cl}} = \begin{bmatrix} -X_{cl}^{-1} & A_0 & B_0 & 0 \\ A_0^T & -X_{cl} & 0 & C_0^T \\ B_0^T & 0 & -\gamma I & D_{11}^T \\ 0 & C_0 & D_{11} & -\gamma I \end{bmatrix}, \tag{13}$$

$$\mathcal{P} = \begin{bmatrix} \mathcal{B}^T & 0 & 0 & \mathcal{D}_{12}^T \end{bmatrix} = \left[\begin{array}{cc|cc|cc} 0 & I & 0 & 0 & 0 & 0 \\ B_2^T & 0 & 0 & 0 & 0 & D_{12}^T \end{array}\right], \tag{14}$$

$$\mathcal{Q} = \begin{bmatrix} 0 & C & \mathcal{D}_{21} & 0 \end{bmatrix} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & C_2 & 0 & D_{21} & 0 \end{array}\right], \tag{15}$$

From **Lemma 1**, (12) is feasible in Θ if and only if

$$W_{\mathcal{P}}^T \Psi_{X_{cl}} W_{\mathcal{P}} < 0;\ W_{\mathcal{Q}}^T \Psi_{X_{cl}} W_{\mathcal{Q}} < 0, \tag{16}$$

where $W_{\mathcal{P}}$ and $W_{\mathcal{Q}}$ denotes bases of Ker$\mathcal{P}$ and Ker$\mathcal{Q}$, respectively. The following notations are introduced to simplify formulas and calculations.

$$\begin{aligned} &\hat{B}_2 := B_2 D_{12}^+; \hat{A} := A - \hat{B}_2 C_1; \hat{B}_1 := B_1 - \hat{B}_2 D_{11}; \\ &\hat{C}_1 := (I - D_{12} D_{12}^+) C_1; \hat{D}_{11} := (I - D_{12} D_{12}^+) D_{11}, \end{aligned} \tag{17}$$

and

$$\begin{aligned} &\tilde{C}_2 := D_{21}^+ C_2; \tilde{A} := A - B_1 \tilde{C}_2; \tilde{C}_1 := C_1 - D_{11} \tilde{C}_2; \\ &\tilde{B}_1 := B_1 (I - D_{21}^+ D_{21}); \tilde{D}_{11} := D_{11} (I - D_{21}^+ D_{21}). \end{aligned} \tag{18}$$

The following result gives a necessary and sufficient condition for the solvability of the γ-suboptimal $\mathcal{H}_\infty$ control problem[2].

Theorem 3. *Consider a proper discrete-time plant G(z) of order n and minimal realization* (1) *and assume* **(A1)-(A2)**. *Let W_{12} and W_{21} denote bases of the null spaces of $(I - D_{12}^+ D_{12}) B_2^T$ and $(I - D_{21} D_{21}^+) C_2$, respectively. With the notation* (17)-(18), *the γ-suboptimal $\mathcal{H}_\infty$ control problem is solvable if and only if*

(i) $\gamma > \max\left(\sigma_{\max}(\hat{D}_{11}), \sigma_{\max}(\tilde{D}_{11})\right)$,

(ii) *there exist pairs of symmetric matrices* (*R*, *S*) in $\mathbb{R}^{n \times n}$ *such that*

$$\hat{C}_1 R \hat{C}_1^T + \gamma^{-1} \hat{D}_{11} \hat{D}_{11}^T < \gamma I; \quad \tilde{B}_1^T S \tilde{B}_1 < \gamma I \tag{19}$$

$$\begin{aligned} &W_{12}^T \{ \hat{A} R \hat{A}^T - R - \gamma \hat{B}_2 \hat{B}_2^T \\ &+ \begin{bmatrix} \hat{C}_1 R \hat{A}^T \\ \hat{B}_1^T \end{bmatrix}^T \begin{bmatrix} \gamma I - \hat{C}_1 R \hat{C}_1^T & -\hat{D}_{11} \\ -\hat{D}_{11}^T & \gamma I \end{bmatrix}^{-1} \begin{bmatrix} \hat{C}_1 R \hat{A}^T \\ \hat{B}_1^T \end{bmatrix} \} W_{12} < 0 \end{aligned} \tag{20}$$

$$\begin{aligned} &W_{21}^T \{ \hat{A}^T S \hat{A} - S - \gamma \tilde{C}_2^T \tilde{C}_2 \\ &+ \begin{bmatrix} \tilde{B}_1^T S \tilde{A} \\ \tilde{C}_1 \end{bmatrix}^T \begin{bmatrix} \gamma I - \tilde{B}_1^T S \tilde{B}_1 & -\tilde{D}_{11}^T \\ -\tilde{D}_{11} & \gamma I \end{bmatrix}^{-1} \begin{bmatrix} \tilde{B}_1^T S \tilde{A} \\ \tilde{C}_1 \end{bmatrix} \} W_{21} < 0 \end{aligned} \tag{21}$$

$$R > 0; \quad S > 0; \quad \lambda_{\min}(RS) \geq 1. \tag{22}$$

Moreover, the set of γ-suboptimal controllers of order r is non empty if and only if (ii) *holds for some R, S which further satisfy the rank constraint*:

Rank $(I - RS) \leq r$.

3. Linear Model of the Non-coherent DLL and Loop Filter Design

In order to apply the optimization $\mathcal{H}_\infty$ theory to the GPS receiver tracking loop problem, a linear model of the tracking loop must be developed. Since, in the previous section, the $\mathcal{H}_\infty$ control theory is phrased in discrete-time, the linear model and associated loop filter will be developed in the discrete-time domain. An in

depth treatise on GPS receiver tracking loops can be found in [3][4].

The complete noncoherent DLL considered in this paper has a bank of 11 correlators: the early, prompt, and the late correlators, and eight dedicated signal power estimation correlators. The outputs of these 11 correlators are combined to form the measurement y_n at 10 Hz, which is input to the loop filter. The loop filter operates on the incoming measurement stream to produce an optimal estimate of the code

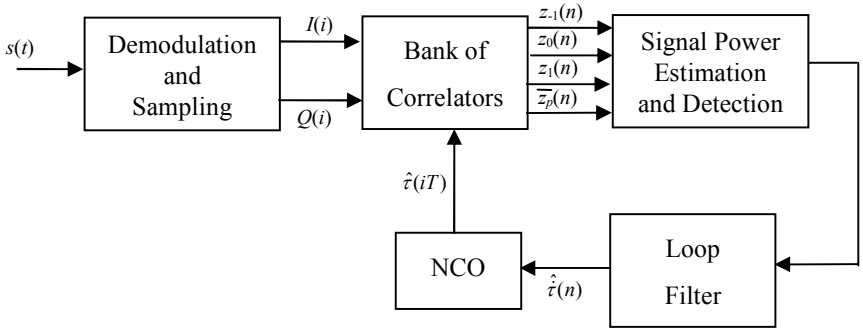

Figure 2. High Level Abstraction of the Noncoherent Delay-Locked Loop

phase velocity $\hat{\dot{\tau}}_n$. The code phase velocity is integrated to form the estimate of the code phase delay $\hat{\tau}(\cdot)$, which is fed back to the bank of correlators. This is depicted in Figure 2.

This model can be further simplified and put into the form shown in Figure 3, which will be referred to as Model 1. This is a linear model of the code tracking loop, whose fundamental input is the code phase delay τ_n. The code phase delay estimate $\hat{\tau}_n$ is subtracted from the code phase delay τ_n to produce the error signal e_n. This error signal is detected in the presence of additive noise v_n. The process of correlation and detection causes a difference in the time at which an estimate $\hat{\tau}_n$ is produced, and the time at which the error in that estimate is detected through y_n. The delay is assumed to be 0.1 s and is represented in the loop as a discrete unit delay ($1/z$). The square law detection process is nonlinear. Therefore, the detector shape is included as a nonlinear block. For small errors of less than half a code chip, the detection process is linear, with $y_n = 2e_{n-1} + v_n$. The output of the loop filter is the estimated code phase velocity $\hat{\dot{\tau}}_n$, which is integrated to produce the estimated code phase $\hat{\tau}_n$. The integration is represented by the z-transform function $0.1/(z - 1)$. In order to design the loop filter, Model 1 is augmented with a shaping filter that accepts white Gaussian noise as its input and produces the LOS code phase delay $\tau(t)$ (measured in second) as its output. The continuous-time state space equations for this shaping filter are

$$\begin{aligned}\begin{bmatrix}\dot{p}(t)\\ \dot{v}(t)\end{bmatrix} &= \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}p(t)\\ v(t)\end{bmatrix} + \begin{bmatrix}0\\ g(t)\end{bmatrix}w(t),\\ \tau(t) &= \begin{bmatrix}\frac{1}{c} & 0\end{bmatrix}\begin{bmatrix}p(t)\\ v(t)\end{bmatrix} + [0]w(t),\end{aligned} \tag{23}$$

where $p(t)$ and $v(t)$ represent platform LOS position and velocity, respectively; c is the speed of light, $w \sim N(0, 1)$ is white Gaussian noise, and $g(t)$ is the gain factor specifying the intensity of the code phase dynamics. When Model 1 is used to produce $\mathcal{H}_\infty$ design, the resulting closed-loop system does not maintain lock in the nonlinear simulation, and in general exhibits problem with stability. It is thought that this is due to differences between the linear model and the truth represented by the nonlinear model. To overcome this difficulty, we adopt a refined linear model shown in Figure 4, which will be referred to as Model 2.

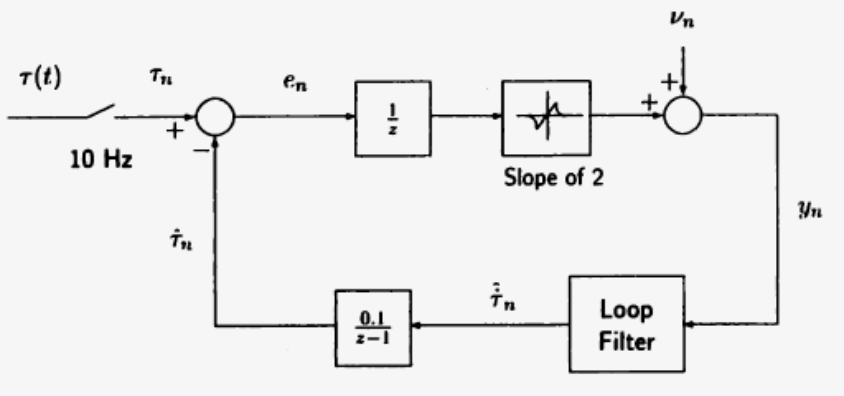

Figure 3. Model 1: Linear Model of the Noncoherent DLL

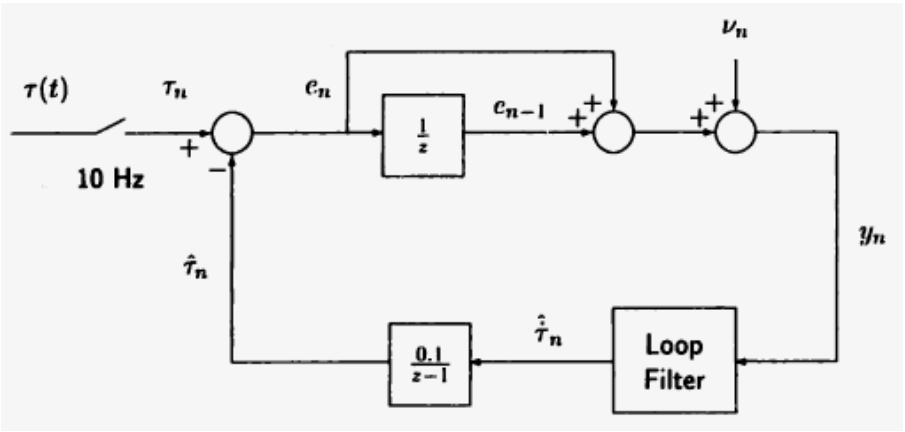

Figure 4. Model 2: Refined Linear Model of the Noncoherent DLL

It is thought that a problem exists in Model 1 with the assumption that the error signal is constant over the code loop interval of 0.1s. The estimate $\hat{\tau}_n$ generally varies much more quickly than the true code phase delay $\tau(n)$, i.e., the error signal $e(n)$ is generally not constant from one code loop iteration to the next. In Model 1, it is assumed that the error detected at time index n is the actual error at time index $n - 1$, multiplied by 2, and corrupted by the additive post correlation noise v_n. In Model 2, the error detected at time

index n is the average of the errors at time indices n and $n-1$, multiplied by 2, and corrupted by the post correlation noise ν_n, i.e.:

$$\text{Model 1: } y_n = 2e_{n-1} + \nu_n,$$
$$\text{Model 2: } y_n = 2\left(\frac{e_{n-1}+e_n}{2}\right) + \nu_n$$

Model 2 better represents the inter-sample nature of the correlation process. For example, as in Figure 5, both Model 1 and 2 should perform equally well for cases when the error signal remains fairly constant between time intervals $n-1$ and n. On the other hand, the error detected in the time interval between n and

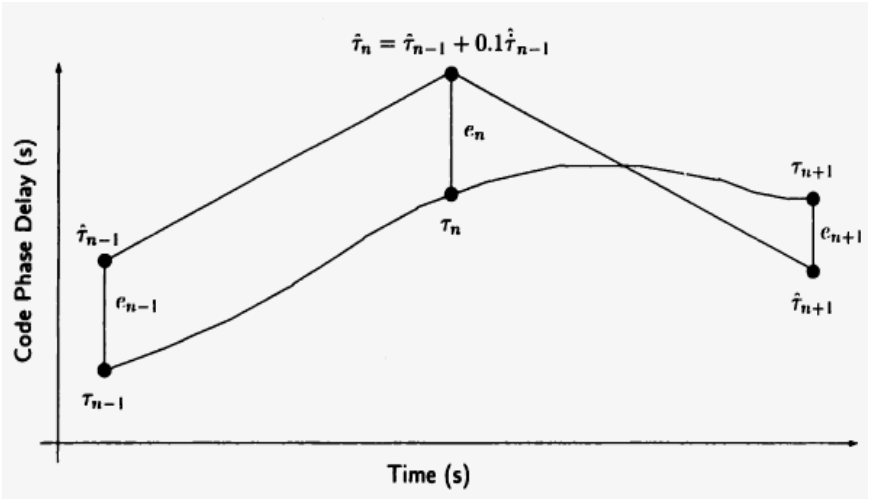

Figure 5. Sample Time Histories of the Code Phase Delay τ_n and its estimate $\hat{\tau}_n$

$n+1$ will be approximately zero, and this is accurately reflected in Model 2, whereas with model 1, the detected error is assumed to approximately e_n.

In order to design the loop filter, the linear model needs to be put into the two-input, two-output generic form. This model is depicted in Figure 6. The generic plant is represented by $G(z)$ and the loop filter is represented by $F(z)$. The first input is a 2×1 vector of disturbances. The first element of the disturbance vector w_n represents noise that drives the LOS dynamics, and the second element of the disturbance vector ν_n represents normalized post-correlation noise. The second input represents the output of the loop filter, or the estimated code phase velocity $\dot{\hat{\tau}}_n$. The first output of the two-input, two-output model is e_n, the code phase estimation error. The second output is the input to the loop filter, or y_n. In the $\mathcal{H}_\infty$ framework, the goal is to minimize the l_2-induced system norm of the closed-loop mapping from the disturbance vector to the error signal. For the remainder of this section, we shall derive the state space model for the generic plant.

The LOS dynamics model is formulated as the discrete-time equivalent to the continuous-time position and velocity model given in equation (23). The resulting 10 Hz model is:

$$\begin{bmatrix} p_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{10} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_n \\ v_n \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{10}\right)^2 \frac{1}{2} \\ \frac{1}{10} \end{bmatrix} g_n w_n. \qquad (24)$$

The code phase delay (measured in code chips) is obtained by scaling the LOS position by the speed of the

light c and the code chip length T_c:

$$\tau_n = \begin{bmatrix} \frac{1}{cT_c} & 0 \end{bmatrix} \begin{bmatrix} p_n \\ v_n \end{bmatrix}. \quad (25)$$

The reason that the code phase delay is expressed in code chips instead of seconds is that the $\mathcal{H}_\infty$ optimization is sensitive to the numerical conditioning of the problem, *i.e.*, the $\mathcal{H}_\infty$ software has many tolerances which are set appropriately for signals that are near unity. Using code chips allows the software to be used with the default tolerances. When the resulting loop filter is utilized in the nonlinear simulation, appropriate scaling is necessary to maintain compatibility.

The integration of the output of the loop filter $\hat{\dot{\tau}}_n$ to produce the estimate of the code phase delay $\hat{\tau}_n$ is realized via the following equation:

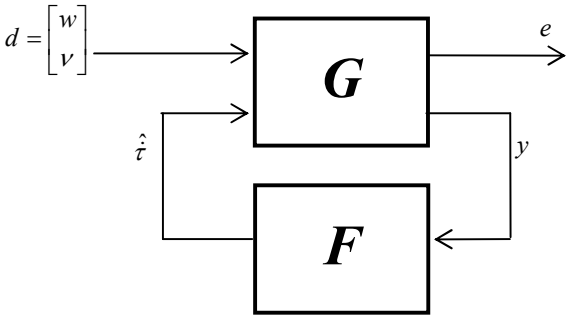

Figure 6. Two-Input, Two-Output Generic Model

$$\hat{\tau}_{n+1} = \hat{\tau}_n + \frac{1}{10}\hat{\dot{\tau}}_n. \quad (26)$$

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

$$e_n = \tau_n - \hat{\tau}_n. \quad (27)$$

The equation for the propagation of the code phase estimation error is obtained by combining the results in equations (24)-(27):

$$e_{n+1} = e_n + \frac{1}{10cT_c}v_n + \frac{g}{200cT_c}w_n - \frac{1}{10}\hat{\dot{\tau}}_n. \quad (28)$$

Let η_n represent the output of the unit delay in Figure 4. The stat space representation of the delay is simply

$$\eta_{n+1} = e_n = \tau_n - \hat{\tau}_n.$$

Assuming that the code phase estimation error is less than one half of a code chip, the linear model for the measurement is

$$y_n = e_n + \eta_n + \sigma_v v_n,$$

where σ_v is the standard deviation of the post correlation noise v_n.

Assembling all of the equations, we can write the state space representation for the open loop two-input, two-output model. The state will consist of a stacked vectorstate = $x_n = [v_n\ e_n\ \eta_n]^T$.

The first input is $[w_n\ v_n]^T$ and the second input is $\hat{\dot{\tau}}$. The first output is e_n and the second output is y_n. A minimal realization for $G(z)$ is given by

$$\begin{bmatrix} v_{n+1} \\ e_{n+1} \\ \eta_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \dfrac{1}{10cT_c} & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_n \\ e_n \\ \eta_n \end{bmatrix} + \begin{bmatrix} \dfrac{g}{10} & 0 \\ \dfrac{g}{200cT_c} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_n \\ \nu_n \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{-1}{10} \\ 0 \end{bmatrix} \hat{\tilde{\tau}}_n,$$

$$e_n = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x_n + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} w_n \\ \nu_n \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \hat{\tilde{\tau}}_n,$$

$$y_n = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} x_n + \begin{bmatrix} 0 & \sigma_\nu \end{bmatrix} \begin{bmatrix} w_n \\ \nu_n \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \hat{\tilde{\tau}}_n.$$

The system $G(z)$ may be written in the following compact form

$$G(z) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right], \tag{29}$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ \dfrac{1}{10cT_c} & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, B_1 = \begin{bmatrix} \dfrac{g}{10} & 0 \\ \dfrac{g}{200cT_c} & 0 \\ 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ \dfrac{-1}{10} \\ 0 \end{bmatrix},$$

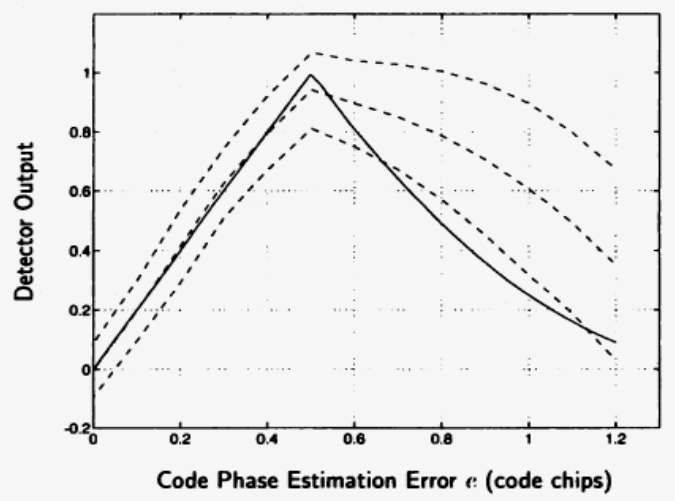

Figure 7. Statistics of Detector Output

$$C_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, D_{11} = \begin{bmatrix} 0 & 0 \end{bmatrix}, D_{12} = \begin{bmatrix} 0 \end{bmatrix},$$
$$C_2 = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}, D_{21} = \begin{bmatrix} 0 & \sigma_\nu \end{bmatrix}, D_{22} = \begin{bmatrix} 0 \end{bmatrix}.$$

4. Simulation Results

In order to produce a $\mathcal{H}_\infty$ suboptimal loop filter, the variance of the post correlation noise v_n and the magnitude of the noise gain g_n in the dynamics model must be known. These two quantities appear in the matrices B_1 and D_{21}, respectively, in equation (29). This is analogous to the requirement of having to known the process and measurement noise intensities in order to derive a conventional Kalman filter.

Let's first consider the post correlation noise v_n. For our example design, assume that the signal power is 9×10^{-16} W, and that the jamming to signal power ratio (J/S) is 40 dB. At this power levels, the empirical output of the detector is as shown in Figure 7. The parameters used in this simulation are given in Table 1. The simulation was run for 30 seconds in order to generate each point in the plot. The solid curve

represents the analytically derived detector output, $R_c^2(e-T_c/2)-R_c^2(e+T_c/2)$. The dashed line that closely follows the solid line is the mean of the detector output. The dashed lines that are above and below the mean represent the empirically obtained standard deviation of the simulated detector output. The effective post correlation noise standard deviation σ_v is a function of the tracking error e. This phenomenon is due to the signal power estimation method. For code phase estimation errors less than 0.5 code chips, the standard deviation is approximately equal to 0.1. Therefore σ_v will be set to 0.1 in the $\mathcal{H}_\infty$ designs. For the code phase dynamics, usually it is strongly dependent on the receiver platform characteristics, the mission, and whether or not IMU (Inertial Measurement Unit) is available, etc. For our simulation, we simply choose $g(t)$ = 10 m/s^2. Observe that the A-matrix of the generalized plant is not stable because it has eigenvalues lying on the unit circle. If the system is not stabilisable, it is generally not possible to find an internally stabilizing controller for the unstable generalized plant. To circumvent this difficulty, we shall slightly perturb the A-matrix into the following form:

Table 1. Detector Output Simulation Parameters

Code Phase Est. Error	e	(Fixed)
Signal Power	P	9×10^{-16} W
Jamming Power	J	9×10^{-12} W
Jamming to Signal Ration	J/S	40 dB
Jamming Time Constant	α	1×10^{-3} s
Boltzmann's Constant	k_b	1.37×10^{-23} J/K
Receiver Equiv. Temperature	T_{eq}	290 K
LOS Dynamics Gain	g	N/A
Code Loop Time-Step	dT	0.1 s
Correlator Integration Time	T_{50}	0.02 s
A/D Integration Time	T	9.7752×10^{-7} s
Chip Length (L5 PRN Code)	T_c	9.7752×10^{-7} s
Correlator Spacing	D	4.8876×10^{-7} s
Simulation Time-Step	dT	0.02 s
Signal Power Est. Cutoff	$\hat{P}_{\min}$	5×10^{-37} W
Signal Power Est. Filter Gain	α	0.01
Signal Power Est. Factor	β	0.9375
Signal Power Est. Factor	γ	0.6391
Dedicated Sig. Pow. Correlators	κ	8

$$A=\begin{bmatrix} 0.99 & 0 & 0 \\ \dfrac{1}{10cT_c} & 0.99 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

In this case, the LOS dynamics model is modified accordingly as:

$$\begin{bmatrix} p_{n+1} \\ v_{n+1} \end{bmatrix}=\begin{bmatrix} 0.99 & \dfrac{1}{10} \\ 0 & 0.99 \end{bmatrix}\begin{bmatrix} p_n \\ v_n \end{bmatrix}+\begin{bmatrix} \left(\dfrac{1}{10}\right)^2\dfrac{1}{2} \\ \dfrac{1}{10} \end{bmatrix} g_n w_n. \quad (30)$$

Using this model, an example trajectory is given in Figure 8 and Figure 9.

Table 2. Filter Design Parameters

Post Correlation Noise Std. Dev.	σ_v	0.1 $2T_c$
LOS Dynamics Gain	g	10 m/s^2
Code Loop Time-step	dT	0.1 s

Using the design parameters given in Table 2, an optimal $\mathcal{H}_\infty$ filter is generated as follows:

$$\xi_{n+1} = A_{H_\infty}\xi_n + B_{H_\infty}y_n,$$
$$\hat{\tau}_n = C_{H_\infty}\xi_n + D_{H_\infty}y_n,$$

where

$$A_{H_\infty} = \begin{bmatrix} 0.3487 & -1.0320 \\ -0.5288 & 0.1419 \end{bmatrix},$$
$$B_{H_\infty} = \begin{bmatrix} 0.9691 \\ -0.0645 \end{bmatrix},$$
$$C_{H_\infty} = \begin{bmatrix} -2.8031 & -4.7640 \end{bmatrix},$$
$$D_{H_\infty} = \begin{bmatrix} 5.1045 \end{bmatrix}.$$

Transfer function of the controller can be derived as:

$$K(z) = \frac{5.104z^2 - 4.913z + 0.000033}{z^2 - 0.4906z - 0.4962}$$

Its pole-zero map is shown in Figure 10. The overall closed-loop system can be obtained by utilizing equation (4). A Bode diagram of the system is shown in Figure 11.

Tracking performances of the proposed design for various level of disturbances are shown in Figures 12-14. In order to justify the performance of the $\mathcal{H}_\infty$ filter, we also derive an $\mathcal{H}_2$ filter (or Kalman filter) for the noncoherent DLL. The $\mathcal{H}_2$ controller is given as follows:

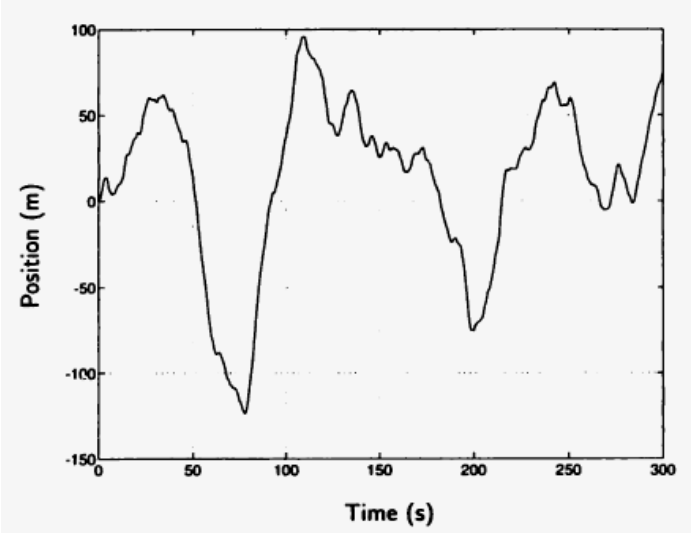

Figure 8. Sample Position Trajectory for Model in Equation (30)

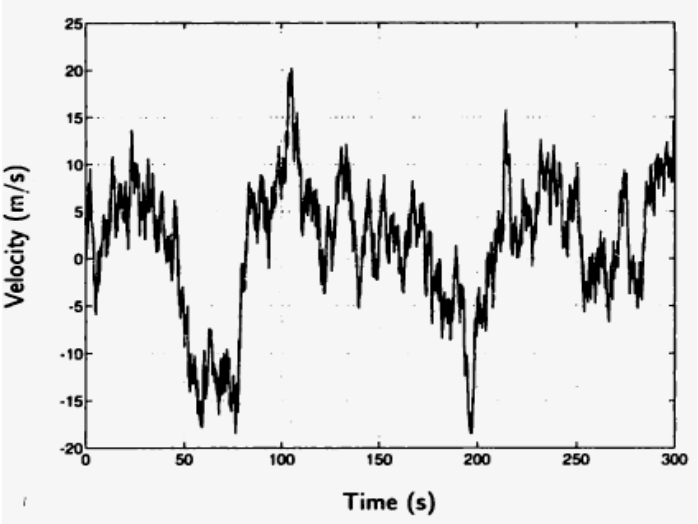

Figure 9. Sample Velocity Trajectory for Model in Equation (30)

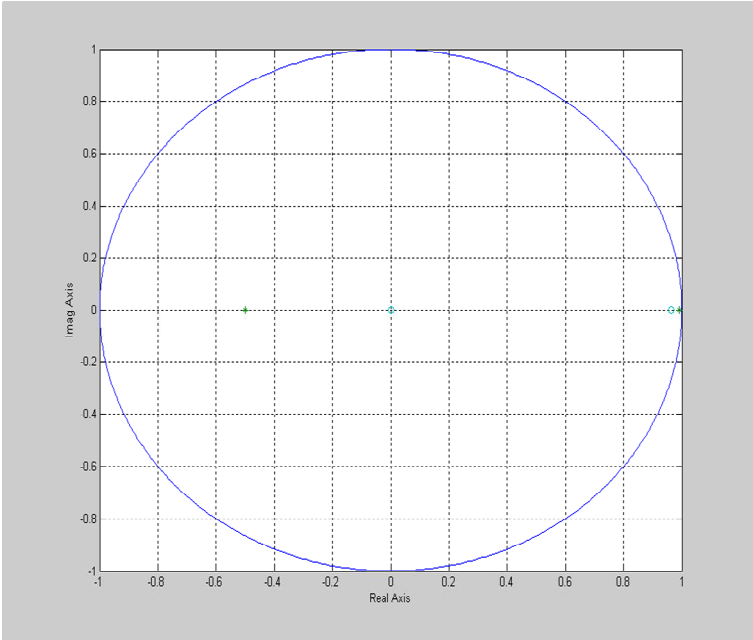

Figure 10. Poles (x) and Zeros (o) of the $\mathcal{H}_\infty$ Optimal Filter

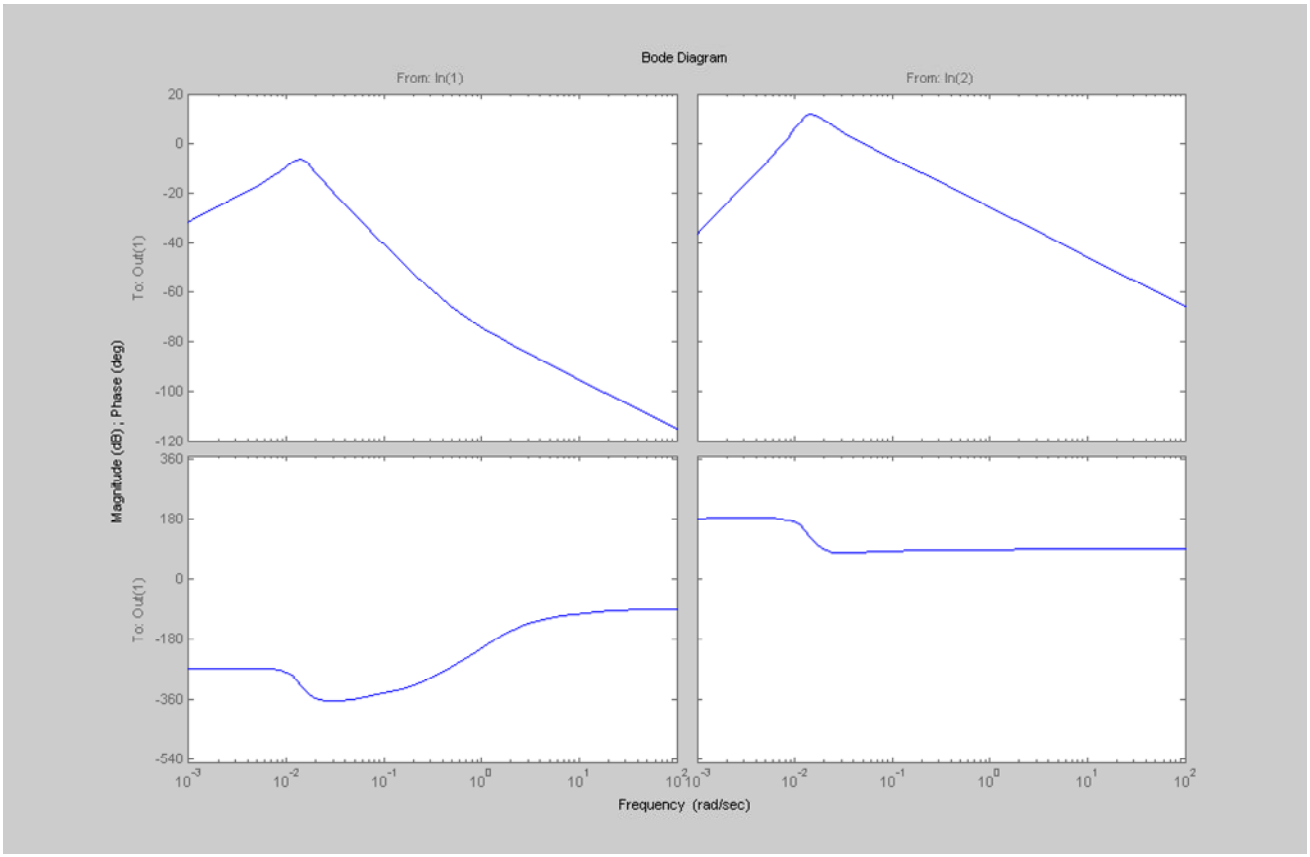

Figure 11. Bode Diagram for the Closed-Loop System

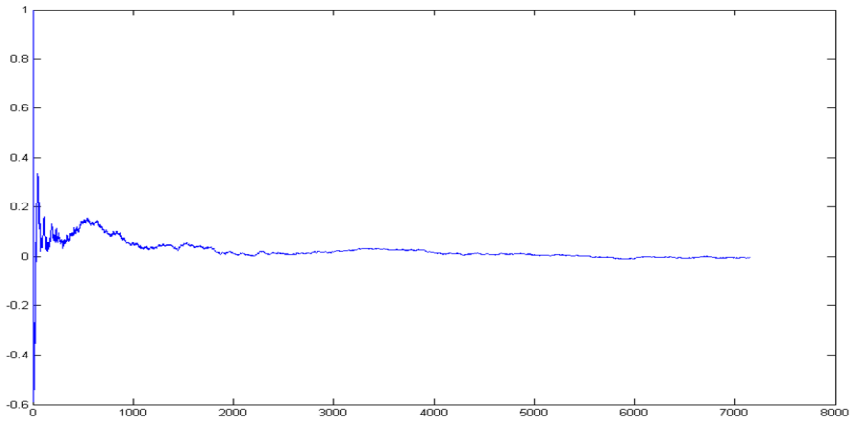

Figure 12. Tracking Performance without Input Disturbance

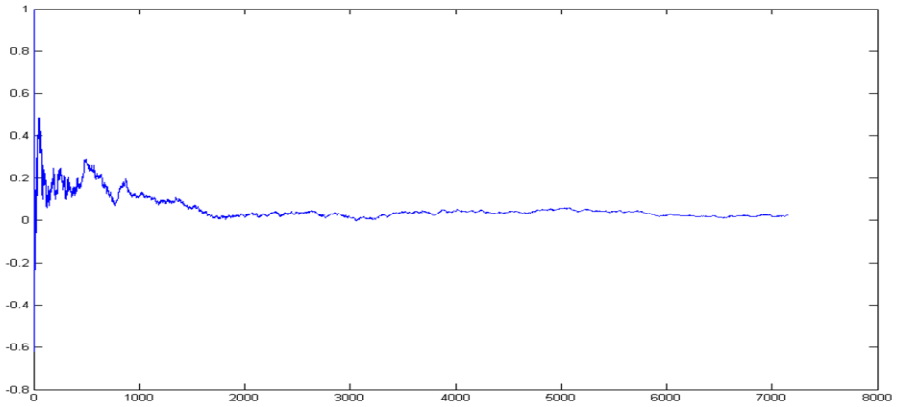

Figure 13. Tracking Performance with Input Disturbance (Random Noise with Mean:1, Variance: 3)

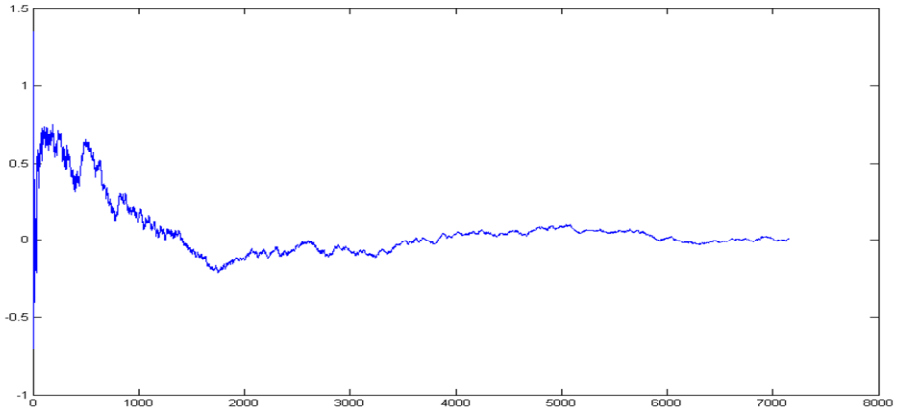

Figure 14. Tracking Performance with Input Disturbance (Random Noise with Mean:5, Variance: 7)

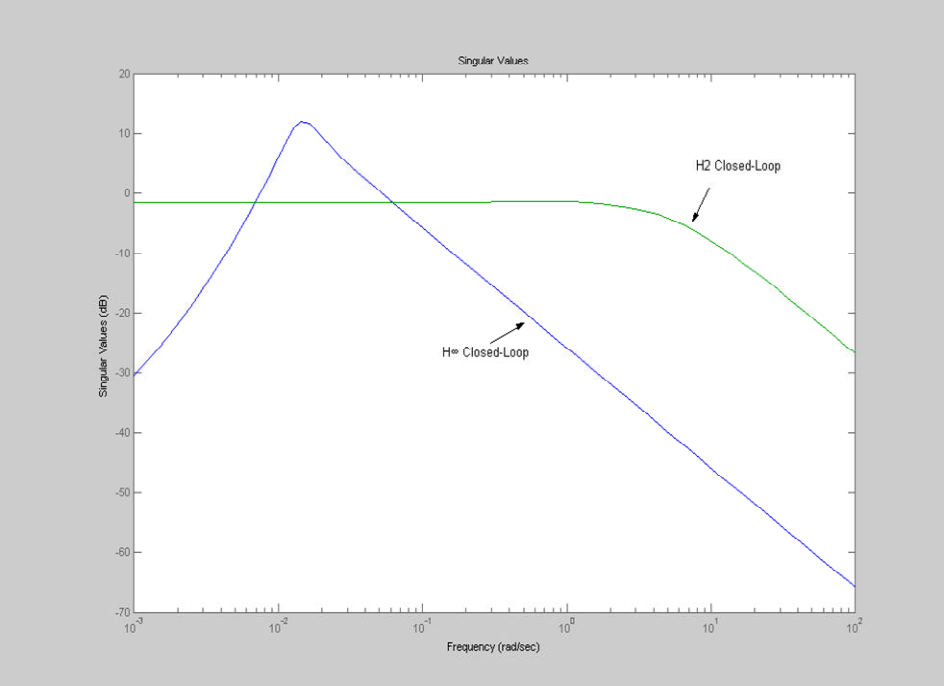

Figure 15. Singular Value Plots of the Closed-Loop Transfer Functions for the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ Designs

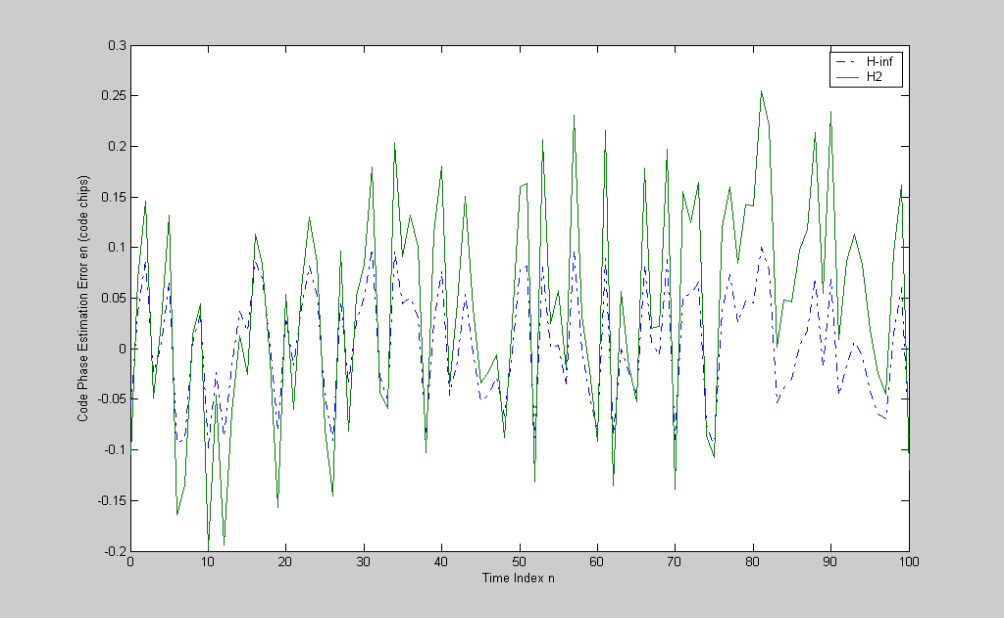

Figure 16. Code Phase Estimation Errors of the Closed-Loop Systems

$$\xi_{n+1} = A_{H_2}\xi_n + B_{H_2}y_n,$$
$$\hat{\tau}_n = C_{H_2}\xi_n + D_{H_2}y_n,$$

where

$$A_{H_2} = \begin{bmatrix} 0.9900 & -7.9512 \\ 0 & -0.1607 \end{bmatrix},$$
$$B_{H_2} = \begin{bmatrix} 7.9512 \\ 0.1607 \end{bmatrix},$$
$$C_{H_2} = \begin{bmatrix} 0.0338 & -1.8622 \end{bmatrix},$$
$$D_{H_2} = \begin{bmatrix} 1.8622 \end{bmatrix}.$$

The singular value plots of the closed-loop transfer functions are shown in Figure 15. For both of the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ designs, code phase estimation errors are shown in Figure 16. It is evident from the figure that the code phase estimation error for the $\mathcal{H}_\infty$ design is relatively small compared to the $\mathcal{H}_2$ design.

5. Conclusion

In this paper, we have derived an $\mathcal{H}_\infty$ control for the GPS receiver tracking loop. The simulation results show that the resulting loop filter can track the PRN code phase in a satisfactory way (see Figure 12-14). Various levels of disturbances are derived as test inputs to evaluate the performance of the resulting tracking loop. In order to justify the performance of the $\mathcal{H}_\infty$ filter, we also derive an $\mathcal{H}_2$ filter for the non-coherent DLL. It is shown in the simulation that the code phase estimation error for the $\mathcal{H}_\infty$ design is relatively small in compared with the $\mathcal{H}_2$ design.

Acknowledgement

The work was supported by National Science Council, Taiwan, R.O.C. under Grant NSC 98-2221-E-019-020-MY3.

References:

[1] Ph. Dondon, T. Tsing, and F. Sandoval, A Didactical Electronic Project for Graduated Students: Initiation to GPS Localisation and Navigation using a Small-Scale Model Electric Car, *WSEAS Trans. on Advances in Engineering Education*, Vol. 6, No. 4, 2009.

[2] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis, State Space Solutions to Standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control Problems, *IEEE Tran. Automatic Control*, Vol. 34, 1989, pp. 831-846.

[3] P. Gahinet and P. Apkarian, A Linear Matrix Inequality Approach to $\mathcal{H}_\infty$ Control, *Int. J. Robust and Nonlinear Control*, Vol. 4, 1997, pp. 421-448.

[4] L. Baroni and H.K. Kuga, Evaluation of Two Integer Ambiguity Resolution Methods for Real Time GPS Positioning, *WSEAS Trans. on Systems*, Vol. 8, No. 3, 2009.

[5] J.J. Spilker, Jr., *Digital Communications by Satellite*, Prentice Hall, 1977, Englewood Cliffs, N.J.

[6] J.J. Spilker, Jr., GPS Signal Structure and Performance Characteristics, *Navigation*, Vol. 25, 1980, pp. 29-54.

[7] J.B.Y. Tsui, *Fundamentals of Global Positioning System Receivers*: *a Software Approach*, John Wiley & Sons, Inc., 2000.

[8] A.J. Van Dierendonck, Satellite Radio Navigation, *Avionics Navigation Systems*, 2nd Edition, M. Kayton and W. R. Fried, Eds., John Wiley & Sons, 1997, pp. 178-282.

[9] H.S. Wang, A Behavioral Approach to GNSS Positioning and DOP Determination, *WSEAS Trans. On Systems*, Vol. 7, No. 9, 2008.

[10] G. Zames, On the Input-Output Stability of Nonlinear Time-Varying Feedback Systems, Part I, *IEEE Transaction on Automatic Control*, Vol. 11, 1966, pp. 228-238.

[11] G. Zames, On the Input-Output Stability of Nonlinear Time-Varying Feedback Systems, Part II, *IEEE Transaction on Automatic Control*, Vol. 11, 1966, pp. 465-47.

[12] S.S.N. Zulkifli, M. Abdullah, A. Zaharim, and M. Ismail, Ionospheric Effects on GPS Range Finding Using 3D Ray-Tracing and Nelder-Mead Optimisation Algorithm, *WSEAS Trans. on Mathematics*, Vol. 8, No. 1, 2009.