Sensors **2011**, *11*, 6575-6592; doi:10.3390/s110706575

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Polynomial Method for PLL Controller Optimization [†]

Ta-Chung Wang [1,⋆], Sanjay Lall [2] and Tsung-Yu Chiou [3]

[1] Institute of Civil Aviation, National Cheng-Kung University, No.1 University Road, Tainan 701, Taiwan

[2] Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA; E-Mail: lall@stanford.edu

[3] MediaTek Inc., No.1 Dusing Road. 1, Hsinchu Science Park, Hsinchu 30078, Taiwan; E-Mail: tsung-yu.chiou@mediatek.com

[⋆] Author to whom correspondence should be addressed; E-Mail: tachung@mail.ncku.edu.tw; Tel.: +886-6-275-7575 ext. 63-620; Fax: +886-6-238-9940.

[†] Based on “Nonlinear Phase-Locked Loop Design using Semidefinite Programming”, by Ta-Chung Wang, Tsung-Yu Chiou and Sanjay Lall, which was presented at the 16th IEEE Mediterranean Conference on Control and Automation. ©2008 IEEE.

*Received: 12 May 2011; in revised form: 15 June 2011 / Accepted: 22 June 2011 /
Published: 27 June 2011*

Abstract: The *Phase-Locked Loop* (PLL) is a key component of modern electronic communication and control systems. PLL is designed to extract signals from transmission channels. It plays an important role in systems where it is required to estimate the phase of a received signal, such as carrier tracking from *global positioning system* satellites. In order to robustly provide centimeter-level accuracy, it is crucial for the PLL to estimate the instantaneous phase of an incoming signal which is usually buried in random noise or some type of interference. This paper presents an approach that utilizes the recent development in the semi-definite programming and sum-of-squares field. A Lyapunov function will be searched as the certificate of the pull-in range of the PLL system. Moreover, a polynomial design procedure is proposed to further refine the controller parameters for system response away from the equilibrium point. Several simulation results as well as an experiment result are provided to show the effectiveness of this approach.

Keywords: non-linear systems; phase-locked loop; optimization

1. Introduction

For nonlinear control systems, one would often like to know the domain of attraction (DoA) of an equilibrium point. Often, this domain is difficult to both find and represent computationally. The usual mathematical tool used for analyzing of the region of attraction is Lyapunov's method. This gives us a sufficient condition for local stability, although it is often difficult to find a Lyapunov function that can be used as a certificate for the whole DoA. Several prior approaches have used quadratic functions, for example [1–3]. In particular, the approach of [3] makes use of semidefinite programming to find a quadratic function whose sublevel-set is a good inner approximation to the region of attraction. For system in which the DoA is complicated, an ellipsoid may not provide a good approximation, and the above methods leave a large unexplored region within the DoA.

With recent developments in algebra and sum-of-squares techniques, it is now possible to solve for a Lyapunov function with a more general polynomial form [4,5]. Positive definiteness properties are replaced by sum-of-squares(SOS) constraints which can be efficiently solved using convex optimization. The SOSTOOLS [6] toolbox for MATLAB has been developed as an easy computational tool to solve problems that utilizes the SOS techniques. This approach has also allowed finding a Lyapunov function within some specified semi-algebraic region [7,8]. While Lyapunov approach provides a method to certify a given inner approximation to the domain of attraction, it does not immediately provide a way to find it. Tan [9] later extended this concept by using unions of SOS polynomials to estimate the domain of attraction but Tan's approach needs to solve a bilinear optimization. The level-set method [10] has been developed to find a semi-algebraic representation of the DoA by semidefinite programming. With these polynomial techniques, it is possible to precisely estimate the DoA of a nonlinear polynomial system and to find a suitable Lyapunov function as the stability certificate.

A phase-locked loop (PLL) system is a nonlinear system with limited domain of attraction. Due to its importance in communication systems, analyzing and designing a PLL system has attracted many attentions in this field [11–17]. The current approach for designing a controller for a PLL system is still largely based on the linear model [17]. Hence, the performance of the resultant system cannot be guaranteed at system states far away from the designed equilibrium point.

In this paper, we utilize the current SOS techniques to analyze the domain-of-attraction of a PLL system. A local Lyapunov function can then be found as the certificate of the DoA using the proposed approach. The Lyapunov function will be further used to improve the stability region and performance of the PLL system. Using this approach, it is possible to design a PLL with a predefined form of controller that has larger domain-of-attraction than the linear design approach. Moreover, since we are designing the system in the nonlinear region, system dynamics outside the linear region can be further refined. Examples of second order PLL systems are used later in this paper to show the effectiveness of this design approach. In the provided examples, we demonstrated that the system designed by the proposed method has 20% larger domain-of-attraction and less overshoot with faster convergent rate than the linear design approach.

This paper is organized as follows. Section 2 contains some preliminary knowledge about SOS techniques. The advection algorithms as well as the SOS methods for finding a local Lyapunov function are stated in Section 3. A brief discussion of the configuration of a PLL can be found in Section 4 and the proposed method for analyzing and designing a PLL controller are presented in Section 5. Then the paper is summarized in Section 6.

2. Preliminaries

The following are some definitions that will be used frequently in this paper. $\mathbb{R}[x]$ is used to represent the ring of polynomials in x with real coefficients. A polynomial $f \in \mathbb{R}[x]$ is called positive semidefinite (PSD) if $f(x) \geq 0$, for all $x \in \mathbb{R}^n$. A polynomial f is called SOS if there exist polynomials $g_1, ..., g_s \in \mathbb{R}[x]$ such that $f = g_1^2 + g_2^2 + \cdots + g_s^2$. Clearly if f is SOS then f is PSD. It is also well-known that the converse is not true. Σ denotes the set of all SOS polynomials in $\mathbb{R}[x]$. $\mathbb{R}_+$ is used to represent the set of nonnegative real numbers. $\mathcal{B}_r$ is used to represent the open ball with radius r centered at the origin.

Suppose $g : \mathbb{R}^n \to \mathbb{R}$ is C^1. Define the 0-*sub-level set* of g to be $sub(g) \subset \mathbb{R}^n$ given by $sub(g) = \{\, x \in \mathbb{R}^n \mid g(x) \leq 0 \,\}$. Further define the boundary of $sub(g)$ as $\partial\, sub(g)$.

One feature of the proposed advection algorithm is that the advection problem can be converted into a semidefinite program. The following is a standard form of a semidefinite program.

$$\begin{array}{ll} \min\limits_{X} & \text{trace}(CX) \\ \text{s. t.} & \text{trace}(A_i X) = b_i \quad \text{for } i = 1, \dots, m \\ & X \succeq 0, \end{array}$$

where $X \in \mathbb{R}^{n \times n}$ is symmetric. $X \succeq 0$ means that $z^T X z$ is positive semidefinite for all $z \in \mathbb{R}^n$.

The condition of one semi-algebraic set containing another semi-algebraic set is one of the key constraints used in this paper. The following lemma shows that this kind of relationship can be converted to constraints on the coefficients of the polynomials. The proof can be found in [4] or [7].

Lemma 1. *Given $p,\ q \in \mathbb{R}[x]$, suppose there exist $s_0, s_1 \in \Sigma$ such that*

$$s_0 - s_1 q + p = 0 \qquad \text{for all } x \in \mathbb{R}^n \tag{1}$$

Then $sub(q) \subset sub(p)$. Further, given q and the degree bound of p, s_0, and s_1, the set of coefficients of p, s_0 and s_1 satisfying (1) is the feasible set of a semidefinite program.

Proof. See, for example, [4] or [7].

The representation shown in Lemma 1 is one of the simplest cases of Schmüdgen's Theorem [18]. Schmüdgen's Theorem states that if $p \in \mathbb{R}[x]$ is strictly positive inside a compact semi-algebraic set S generated by $p_1, \dots, p_m$ as $S = \{x \in \mathbb{R}^n \mid p_i \geq 0,\ i = 1, 2, \dots, m\}$, then

$$p = \Sigma_v p_1^{v_1} \dots p_m^{v_m} s_v$$

where $v = (v_1, \dots, v_m) \in \{0, 1\}^m$ and $s_v \in \Sigma$. Putinar [19] later showed that under some additional constraints on p_i, p has a simpler representation as

$$p = s_0 + s_1 p_1 + \dots + s_m p_m$$

The gap between Schmüdgen's and Putinar's representation is later investigated by Jacobi and Prestel [20]. In the simple case shown in Lemma 1, if $sub(q) \subset sub(p)$ and $sub(q)$ is compact, the representation of p by (1) is always possible.

The following result is similar. Given $q \in \mathbb{R}[x]$, if there exists $s_0,\ s_1 \in \Sigma$ and $\epsilon > 0$ such that

$$s_0 + s_1 q - p + \epsilon = 0$$

then $sub(p) \subset sub(q)$.

Usually q is a given polynomial and p is the solution to find such that $sub(p)$ and $sub(q)$ approximately represent the same set with some other constraints on p, such as having lower degree or passing through several pre-specified points. The above results are used to construct such constraints.

3. Acquiring the Local Lyapunov Function

Finding a local Lyapunov function is coupled with finding the DoA. Without a clear knowledge of the actual shape of the DoA, it is hard to find a Lyapunov function that can be used to represent the entire DoA [4,8]. To deal with this difficulty, we utilize the current development in set advection [10].

3.1. Set Advection

In this paper, we will consider the following autonomous system

$$\dot{x}(t) = f(x) \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz. From the basic local existence and uniqueness theorem [21], given an open subset $U \in \mathbb{R}^n$, there exist $c \in \mathbb{R}_+$ such that the autonomous system (2) has a unique solution for any $z \in U$ in the compact time interval $[-c, c]$.

We define the *flow map* $\phi_t : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ to be the local unique solution of

$$\begin{aligned} \frac{\partial \phi_t(z)}{\partial t} &= f(\phi_t(z)) \quad \text{for } t \in [-c, c],\ c(z) \in \mathbb{R}_+,\ z \in \mathbb{R}^n \\ \phi_0(z) &= z \end{aligned}$$

For any $t \in \mathbb{R}$ such that $\phi_t(x)$ exists, the map $\phi_t : \mathbb{R}^n \to \mathbb{R}^n$ is continuous, invertible and has a continuous inverse, *i.e.*, it is a topological homeomorphism on $\mathbb{R}^n$ [22].

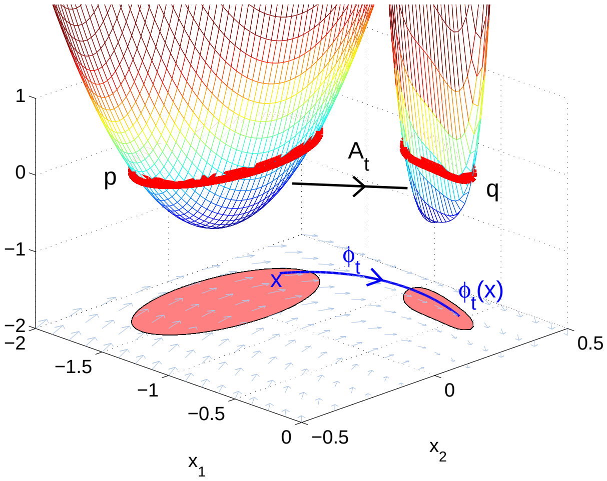
Given $t \in \mathbb{R}$, we define the time t advection operator $A_t : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R})$ by

$$q = A_t p \quad \text{if} \qquad q(x) = p\left(\phi_{-t}(x)\right) \text{ for all } x \in \mathbb{R}^n$$

where $C(X, Y)$ is the set of functions mapping from X to Y. The map A_t is also called the Liouville operator associated with f. A very important property is that it is linear. Figure 1 shows the concept of the advection operator. Given polynomial p, A_t maps the coefficients of p to another polynomial q such that $sub(q) = \phi_t\, sub(p)$. We relate the advection operator to the advection of sets in the following remark.

Remark 1. *Suppose $g_1,\ g_2$ are functions mapping $\mathbb{R}^n$ to $\mathbb{R}$. If $g_2 = A_t g_1$ then $sub(g_2) = \phi_t\left(sub(g_1)\right)$.*

Figure 1. The advection operator A_t.



3.2. Time-Stepping

Since we are performing advection, we must use an approximation to the flow map ϕ_h with time step h. Many such approximations are possible, and are provided by the theory of numerical integration. The first-order Taylor approximation to $q = A_h p$ is the map $B_h : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R})$ given by

$$q = B_h p \qquad \text{if } q(x) = p(x) - hDp(x)f(x)$$

where the derivative $Dp(x)$ is a linear map $Dp(x) : \mathbb{R}^n \to \mathbb{R}^n$ at each point x.

Based on the required accuracy of the advection, we could also choose to use higher order Taylor approximation. However, depending on the system dynamics, this usually will lead to the requirement of using higher degree polynomials in the sum-of-squares constraints. The relationship between the accuracy and the degree of polynomials will be further investigated in future work.

3.3. Domain-of-Attraction Estimation

The set advection concept is used to estimate the DoA of a system. We use the following definition of the DoA in this paper.

Definition 1. *Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ is analytic with the flow map, ϕ, and the origin is asymptotically stable. Define the* domain-of-attraction *(also called the basin/region of attraction) of f to be $R \subset \mathbb{R}^n$ such that for any $x \in R$, $\phi_t(x)$ is defined for all $t \geq 0$ and $\lim_{t\to\infty} \phi_t(x) = 0$.*

The following properties can be easily derived. The detailed proofs can be found in [10].

Lemma 2. *Suppose f is analytic and the origin is asymptotically stable and $R \neq \emptyset$. Suppose also $S_1 \subset R$ and $0 \in S_1$, and S_1 is a connected closed positively invariant set. Let $h > 0$ be a positive constant, and define the* backward advection *of S_1 to be S_2, given by*

$$S_2 = \phi_{-h} S_1$$

Then $S_1 \subset S_2 \subset R$, and S_2 is also connected, closed and positively invariant. Furthermore, $\partial S_2 = \phi_{-h} \partial S_1$.

Theorem 1. *Suppose f is analytic and the origin is asymptotically stable and $h > 0$. Also suppose $0 \in S_0$ and $S_0 \subset R$ is a closed connected positively invariant set, such that there exists $\epsilon > 0$ such that $\mathcal{B}_\epsilon \subset S_0$.*

Define the sequence of sets S_0, S_1, $S_2, \ldots$ by

$$S_{k+1} = \phi_{-h} S_k \qquad \textit{for } k = 0, 1, 2, \ldots$$

Then this sequence converges to R in the following sense:

(i) $S_k \subset R$ *for all* $k \in \mathbb{N}$.

(ii) $S_k \subset S_{k+1}$ *for all* $k \in \mathbb{N}$.

(iii) For every $x \in R$, there exists n such that $x \in S_n$

3.4. Star-Shaped Constraint

For the case of estimating the DoA, we introduce the concept of star-shaped sets. The star-shaped sets have many important properties and can be easily implemented as a semidefinite program. We now start with the first property. The detailed information about the star-shaped set can be found in [10].

Definition 2. *A set $S \in \mathbb{R}^n$ is called* star-shaped *if for all $x \in S$*

$$\lambda x \in S \qquad \textit{for all } \lambda \in [0, 1]$$

The set S is called strictly star-shaped *if for all $x \in S$*

$$\lambda x \in int(S) \qquad \textit{for all } \lambda \in [0, 1)$$

Note that a star shaped set S is connected. We now give a simple sufficient condition that ensures a sub-level set is star-shaped. We make the following definition.

Definition 3. *Suppose $g : \mathbb{R}^n \to \mathbb{R}$. We call g* strictly star-shaped *if g is C^1 and further satisfies $g(0) < 0$ and*

$$Dg(x)x > 0 \qquad \textit{for all } x \neq 0$$

The following lemma shows the connection between strictly star-shaped functions and star-shaped sets.

Lemma 3. *Suppose $g : \mathbb{R}^n \to \mathbb{R}$ is strictly star-shaped. Then $sub(g)$ is strictly star-shaped.*

Proof. Suppose $x \in sub(g)$. Let $y : \mathbb{R}_+ \to \mathbb{R}^n$ be the function

$$y(t) = e^{-t} x$$

The trajectory of $y(t)$ follows the straight line connecting x and the origin. We would like to show that $y(t) \in sub(g)$ for all $t \geq 0$. We have

$$\begin{aligned} \frac{d}{dt} g(y(t)) &= -Dg\left(y(t)\right) y(t) \\ &< 0 \end{aligned}$$

for all $t \geq 0$. Also

$$g\left(y(t)\right) - g\left(y(0)\right) = \int_0^t \frac{d}{dt} g\left(y(t)\right)\, dt$$

and since $y(0) = x$ we have $g\left(y(t)\right) < 0$ for all $t \geq 0$ as desired.

Now we need to show that if $y \in \partial\, sub(g)$ then there does not exist $\lambda \in [0, 1)$ such that $\lambda y \in \partial\, sub(g)$. Suppose for the sake of a contradiction that there does exist such a y and λ. We know that there exists such $\lambda > 0$, since $g(0) < 0$. Define the function $h : [0, 1] \to \mathbb{R}$ by

$$h(\theta) = g(\theta y) \qquad \text{for } \theta \in [0, 1]$$

Then the derivative of h is

$$\begin{aligned} h'(\theta) &= \frac{1}{\theta} Dg(\theta y)(\theta y) \\ &> 0 \end{aligned}$$

for all $\theta \in (0, 1)$. From the assumptions we know $h(\lambda) = 0$ and $h(1) = 0$. Since h is C^1 on $[\lambda, 1]$, by the mean-value theorem there must exist $\theta \in (\lambda, 1)$ such that $h'(\theta) = 0$, which is a contradiction.

For the purposes of this paper, we would like to construct a convex set of functions whose sub-level sets are connected. Although the convex set of all convex functions on $\mathbb{R}^n$ will suffice, using it would unnecessarily restrict the class of sets describable to be convex. One cannot simply use the set of all functions whose 0-sub-level set is connected, since this set of functions is not convex. We therefore choose the set of *strictly star-shaped* functions, which is a convex set. We will use strictly star-shaped polynomials to represent sets. This is significantly more general than existing approaches using quadratic functions [1–3]. Also, it has been shown in Lemma 3 that if g is strictly star-shaped, then $sub(g)$ is strictly star-shaped. By using this property, we can easily pose the star-shaped constraints on g to make $sub(g)$ a connected set.

3.5. An Algorithm for Backward Advection

Here we will state the result of the backward advection algorithm. Given a strictly star-shaped polynomial g_{i-1} such that $sub(g_{i-1}) \subset R$, and $sub(g_{i-1})$ is bounded and positively invariant, we compute a polynomial g_i such that $sub(A_h g_i) \approx sub(g_{i-1})$ as follows.

Pick $\alpha > 0$ and positive integer d. Solve, using semidefinite programming, the following feasibility problem for $g_i \in \mathbb{R}[x]$, s_1, s_2, s_3, $s_4 \in \Sigma$.

$$\begin{aligned} g_i(0) &= -1 \\ Dg_i(x)x &> 0 \\ s_3 - s_4 g_{i-1} + B_{(h-\alpha)} g_i &= 0 \\ s_1 + s_2 g_{i-1} - B_h g_i &= 0 \\ \deg(g_i) &\leq d \end{aligned}$$

Here we introduced an important parameter, α, which we think of as follows. The above algorithm finds a degree d polynomial g_i such that g_i is strictly star shaped, $\phi_h\, sub(g_i) \subset sub(g_{i-1})$, and

$\phi_{h-\alpha}\, sub(g_i) \supset sub(g_{i-1})$. Hence the parameter α may be thought of as a tolerance that allows for the constraint that g_i is required to have degree d or less. Then from the result of Theorem 1, $\lim_{i\to\infty} sub(g_i)$ converges to the domain-of-attraction. It should be noted that this technique only works in the case that the advected set is positively/negatively invariant.

3.6. Stopping Conditions

By using the proposed level-set method, one can successfully propagate the system states backward in time. However, a stopping criterion is still needed to terminate the iterations. To detect the convergence of the advected sets, the closeness of two semi-algebraic sets is analyzed. The following result shows that the closeness of two semi-algebraic sets can be estimated by using scaled sets. The detailed proof can be found in [10].

Theorem 2. *Suppose g_1 and g_2 are strictly star-shaped functions, $sub(g_1) \subset sub(g_2)$, and $sub(g_2)$ is bounded. Suppose x_1, $x_2 \in \mathbb{R}^n$ are two points such that*

$$x_1 \in \partial\, sub(g_1) \qquad \text{and} \qquad x_2 \in \partial\, sub(g_2),$$

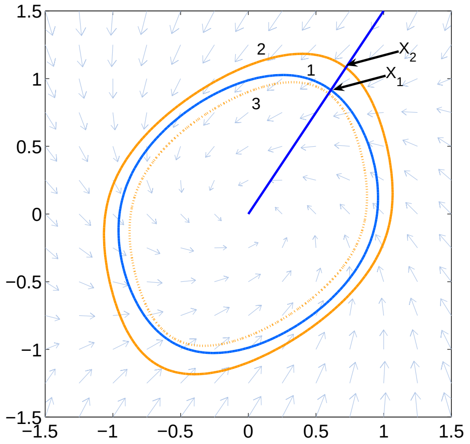
and $x_1 = \alpha x_2$ for some $\alpha \geq 0$. Define the function $q : \mathbb{R}^n \to \mathbb{R}$ by

$$q(x) = g_2(\lambda x) \qquad \text{for all } x \in \mathbb{R}^n$$

where $\lambda > 1$. Then if $sub(q) \subset sub(g_1)$,

$$\frac{\|x_2 - x_1\|}{\|x_2\|} \leq 1 - \lambda^{-1}$$

Figure 2. Example of stopping conditions. Curve 1 is the inner set and curve 2 is the outer set. Curve 3 is the shrunk set of the outer set.



To determine when the algorithm should terminate, one formulates an optimization problem using Lemma 1 to determine the smallest $\lambda > 1$ such that $sub(q) \subset sub(g_2)$, where $q(x) = g_2(\lambda x)$. Again, this may be evaluated using semidefinite programming. In practice, one picks a $\lambda > 1$ in advance, and checks this condition after each iteration. Figure 2 shows an example. Here Curve 1 is $\partial\, sub(g_1)$, Curve 2 is $\partial\, sub(g_2)$, and Curve 3 is $\partial\, sub(q)$. The largest radial deviation between Curves 1 and 2 is less than 0.3.

3.7. The Local Lyapunov Function

We find a local Lyapunov function in order to construct an initial star-shaped positively invariant set. The following result is standard.

Proposition 1. *Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ is analytic and the origin is a stable equilibrium point. Also suppose $V : \mathbb{R}^n \to \mathbb{R}$ is a C^1 function, $\gamma > 0$, and the set*

$$D_\gamma = \{\, x \in \mathbb{R}^n \mid V(x) \le \gamma \,\}$$

is compact. Further suppose

$$\begin{aligned} V(x) &> 0 && \textit{for all } x \neq 0 \\ V(0) &= 0 \\ DV(x)f(x) &< 0 && \textit{for all } x \neq 0, x \in D_\gamma \end{aligned}$$

Let $g_0(x) = V(x) - \gamma$. Then $sub(g_0)$ is positively invariant, and $sub(g_0) \subset R$.

One simple approach to finding an initial sub-level set is to find a quadratic Lyapunov function for the linear model of the system, and use a small sub-level set of this quadratic polynomial as the initial set.

An alternative method which often gives a much larger initial set is as follows. Choose a polynomial $p \in \mathbb{R}[x]$ such that $sub(p) \subset R$. We then solve the following convex feasibility problem. Find $V \in \mathbb{R}[x]$ and $s_0, s_1 \in \Sigma$ such that

$$\begin{aligned} DV(x)x &> 0 && \text{for all } x \neq 0 \\ V(x) &> 0 && \text{for all } x \neq 0 \\ V(0) &= 0 \\ DV(x)f(x) + s_0 - s_1 p &= 0 && \text{for all } x \neq 0 \end{aligned}$$

Similar methods for finding local Lyapunov functions along with details on the construction of the associated semidefinite program may be found in [5,8]. Here we have added the first constraint to ensure that $V - \gamma$ is strictly star-shaped for $\gamma > 0$. Note that these constraints imply that all sub-level sets of V are compact. Given V, we then solve the convex program

$$\begin{aligned} &\text{maximize} && \gamma \\ &\text{subject to} && V - \gamma - s_0 - s_1 p - \epsilon = 0 && \text{for all } x \\ &&& s_0, s_1 \in \Sigma \end{aligned}$$

where $\epsilon > 0$ is small. The optimal γ satisfies $sub(V-\gamma) \subset sub(p)$. Then V and γ satisfy the assumptions of Proposition 1 and so we may use $g_0 = V - \gamma$ as the function defining our initial level-set.

After we have found an estimate of the DoA, we can then use Proposition 1 to find a Lyapunov function that can be used to describe the behavior of the system within the DoA. To adequately describe the system behavior, this approach requires that we have a good estimate of the DoA. The following examples show that the level-set algorithm can precisely estimate the DoA.

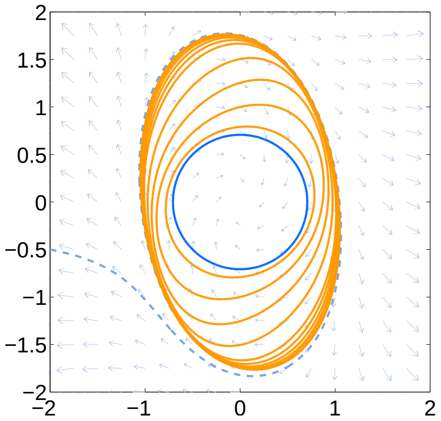
3.8. Examples of DoA Estimation

Example 1. *Consider the following dynamical system*

$$\dot{x} = 0.5y - x(1 - x^2 - 0.25y^2)$$
$$\dot{y} = -x - 0.5y(1 - x^2 - 0.25y^2)$$

The origin is a locally stable equilibrium point. Here we start with the initial polynomial $g_0 = 2x^2 + 2y^2 - 1$. The results of the level-set method are shown in Figure 3, using time step $h = 0.2$. It can be seen that the successive iterates approach the true DoA.

Figure 3. Successive iterates of the level-set algorithm.



Example 2. *Consider the Van der Pol oscillator with inverted time:*

$$\dot{x} = -y$$
$$\dot{y} = x - y(1 - x^2)$$

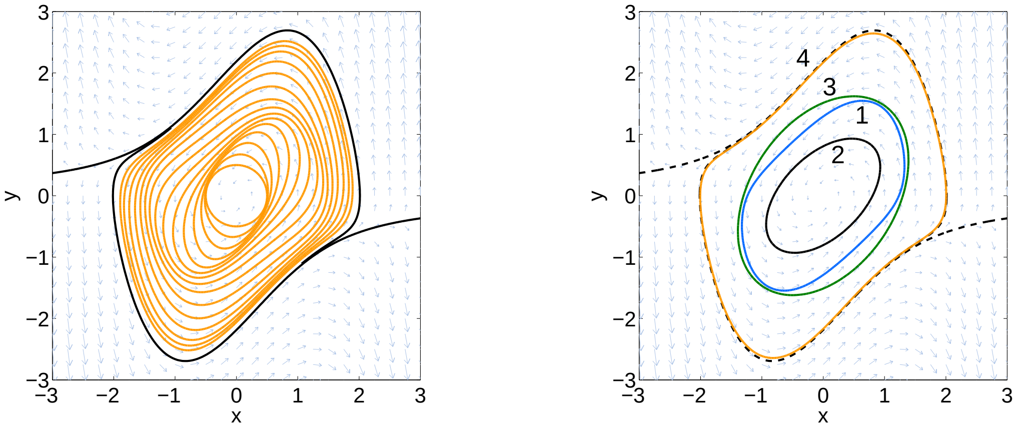
This system is locally stable around the origin. An initial sub-level set given by the quadratic polynomial $g_0 = 4x^2 + 4y^2 - 1$ is used which can be verified to be positively invariant. A time step of $h = 0.2$ is used. The time tolerance parameter α is 0.02. The even-numbered iterates g_0, g_2, $g_4, \ldots$ are shown in Figure 4. For reference, some of the iterates are listed below and normalized to allow integer coefficients.

$$\begin{aligned}
p_2 &= -1,000 + 2,252\,y^2 - 88\,y^4 + 11\,y^6 - 907\,xy - 56\,xy^3 - 4\,xy^5 + 3,883\,x^2 \\
&\quad + 360\,x^2y^2 - 57\,x^2y^4 + 660\,x^3y - x^3y^3 - 417\,x^4 + 21\,x^4y^2 + 81\,x^5y + 260\,x^6 \\
p_4 &= -1,000 + 1,614\,y^2 - 137\,y^4 + 16\,y^6 - 1,654\,xy - 170\,xy^3 + 14\,xy^5 + 3,162\,x^2 \\
&\quad + 480\,x^2y^2 - 43\,x^2y^4 + 94\,x^3y - 35\,x^3y^3 + 144\,x^4 - 2\,x^4y^2 + 192\,x^5y + 335\,x^6 \\
p_{28} &= -10,000 + 2510\,y^2 - 56\,y^4 + 2\,y^6 - 4,306\,xy + 42\,xy^3 + 4\,xy^5 + 4,099\,x^2 \\
&\quad + 25\,x^2y^2 + 2\,x^2y^4 + 1,103\,x^3y - 27\,x^3y^3 - 687\,x^4 - x^4y^2 + 2\,x^5y + 84\,x^6
\end{aligned}$$

It can be seen that the iterates gradually approach the exact boundary of the DoA. After 30 iterations, the solution covers most of the stable region.

After 40 *iterations, the stopping criteria allowing an absolute radial change of* 0.01 *has been met. The final result is shown in Figure 4 as Curve* 4*. Curve* 1 *is* $\partial\mathcal{C}(g_0)$ *if we were using the semidefinite-programming based procedure in Section 3.7. For comparison, Curve* 2 *is the result of [2] and Curve* 3 *is the result of [1].*

Figure 4. Van der Pol oscillator. The left figure shows the sequence of iterations. The right figure shows the final iterate as Curve 4 along with some other results. Curve 1 is $\partial\, sub(g_0)$, when g_0 is obtained through the semidefinite programming based procedure in Section 3.7. For comparison, Curve 2 is the result of [2] and Curve 3 is the result of [1].

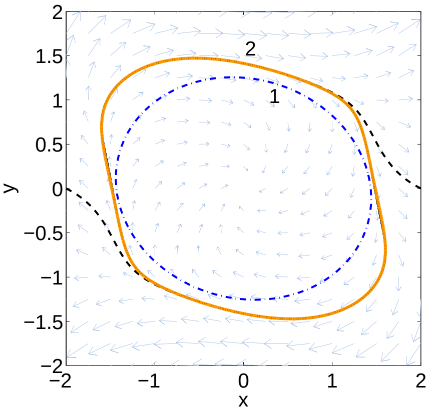


Example 3. *The following dynamic system is the Example S4 taken from [3].*

$$\dot{x} = -2x + y + x^3 + y^5$$
$$\dot{y} = -x - y + x^2y^3$$

Here the initial set is a small circle around the origin. After 20 iterations, the estimated boundary of the DoA has reached the pre-specified bound. The final iterate is shown in Figure 5. The dashed curves show the two system trajectories which are used to represent the true boundary of the domain of attraction. Curve 2 represents the result of the final iterate. Curve 1 is the result from [3].

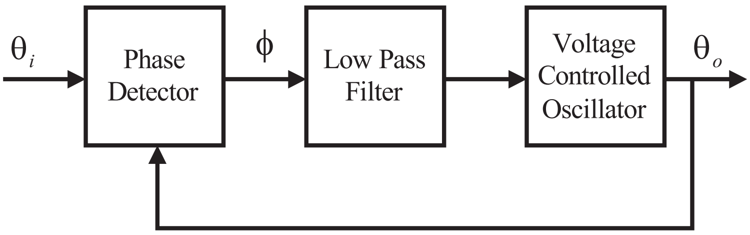
Figure 5. Sub-level sets of g_0 and g_{20}. The dashed-dot curve is the result from [3] and the solid curve is the result of the advection algorithm.



4. Configuration of a PLL

Figure 6 shows the basic configuration of a PLL. It has three components; a phase detector, a loop filter, and a *voltage controlled oscillator*(VCO). The VCO generates an output signal whose phase, $\theta_0(t)$, depends on the phase, $\theta_i(t)$, of the input signal. The PLL is phase locked when the phase error $\phi(t) = \theta_i(t) - \theta_0(t)$ is a constant value and the loop is in stable equilibrium state. Usually, it is desired that the phase error, $\phi(t)$, is maintained at zero.

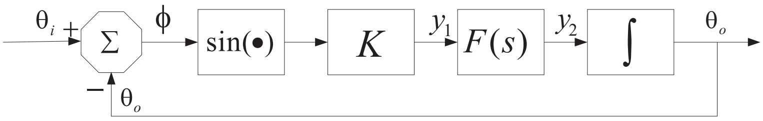
Figure 6. Basic Configuration of a PLL.



Of interest is the behavior of the phase error $\phi(t)$. Because of its sinusoidal nonlinearity in the PLL, the phenomenon of chaos is believed to exist [11,12] and its inherent chaotic behavior for broadening the pull-in range of PLL has also been realized [13,14]. A nonlinear controller can drive PLL from chaotic state into periodic state or vice versa [15]. For higher-order PLL, it is not possible to determine whether the loop will or will not slip cycles using the initial frequency alone. In this case, one might define the pull-in range as the separatrix ordinate at $\phi = 0$ [17]. Analyzing the DoA of the PLL system provides a better description of the region in which a PLL locks up without slipping. The Lyapunov method has been used for stability analysis in control systems. Here the advection algorithm will be used to find the guaranteed stability boundary of the PLL system and the associated local Lyapunov function is then used to further refine the controller parameters. In [16], a Lyapunov styled analysis for PLL system up to third order is presented. The method shown in this section provides a way to analyze the DoA for a more general system. Also, the form of the Lyapunov function used here is much more flexible.

Figure 7 shows the nonlinear model of the PLL. The sine function here represents the phase detector of the system. K in Figure 7 stands for the loop gain of the system. $F(s)$ is equivalent to the low pass filter shown in Figure 6 and it corresponds to the controller of the PLL. Finally, the integrator in Figure 7 is the voltage or numerically controlled oscillator. The key idea of a PLL system is to use the command, y_2, from $F(s)$ to steer the oscillator such that $\theta_0(t)$ tracks $\theta_i(t)$ as closely and quickly as possible.

Figure 7. Model of the Phase-Locked Loop.



4.1. Second Order PLL

To use the nonlinear design approach, start with a reference design. The reference design used in this paper is the linear model of a PLL system. A Proportional-Integrator (PI) controller is chosen to be the filter, $F(s)$, as

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s} \tag{3}$$

Using the model shown in Figure 7, it is routine to check that the resulting dynamic equation of the system is

$$\frac{d^2\phi}{dt^2} + K\frac{\tau_2}{\tau_1}\cos(\phi)\frac{d\phi}{dt} + \frac{1}{\tau_1}K\sin(\phi) = \frac{d^2\theta_i}{dt^2} \tag{4}$$

Assume that the received signal frequency is varying linearly with time and has zero radial acceleration and let $x_1 = \phi,\ x_2 = \dot{\phi}$. The PLL system can be rewritten as the following state space model

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= -K\frac{\tau_2}{\tau_1}\cos(x_1)x_2 - \frac{1}{\tau_1}K\sin(x_1) \\ &= k_1\cos(x_1)x_2 + k_2\sin(x_1) \end{aligned} \tag{5}$$

Equation (5) is the final nonlinear model of the second order PLL. A linearized model can then be derived as

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= k_1 x_2 + k_2 x_1 \end{aligned}$$

The filter F can then be designed using existing linear design approach [17]. One typical choice is to let $\omega_n = 15, \zeta = 0.707, K = 1$, where ω_n is the natural frequency, ζ is the damping ratio, and K is the overall gain. The two coefficients of the filter are then given as $\tau_1 = \frac{K}{\omega_n^2}, \tau_2 = \frac{2\zeta}{\omega_n}$

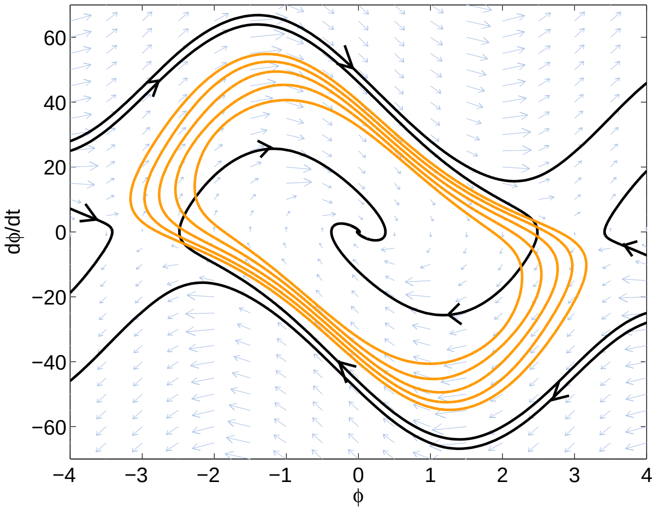
5. Phase-Locked Loop Analysis and Design

In this section, the second order PLL controller will be used to demonstrate the nonlinear design approach. The same approach can also be applied to the third order controller design.

5.1. Pull-In Range of the Traditional PLL System

Now the advection algorithm can be applied to the PLL nonlinear system. To reduce the required number of iterations, a local Lyapunov function is used as the initial set. After a few iterations of the algorithm, it gives us the estimated domain-of-attraction of the system. The result is shown in Figure 8. Note that the estimated region is based on the Taylor series expansion of the sine and cosine functions. From *Ston-Weierstrass* Theorem, we can approximate the sine and cosine functions to the desired accuracy within a bounded interval. In this example, two degree-10 polynomials are used to approximate sine and cosine functions and the estimated region is only valid between $-\pi$ and π. More terms of the Taylor series could also be used to improve the accuracy.

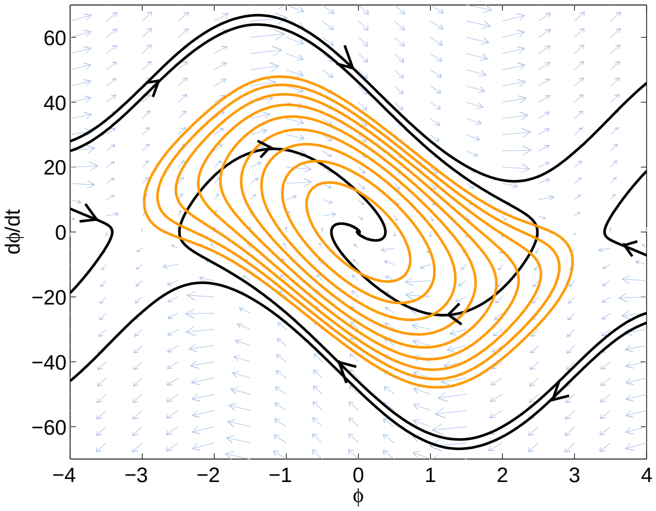
Figure 8. Result of the original PLL system.



5.2. PLL System Controller Design

After getting a good estimated DoA, a local Lyapunov function that describes the system behavior can be easily computed using Proposition 1. This local Lyapunov function can also be used to describe the trajectories of the system in the DoA. Figure 9 shows several Lyapunov level-sets obtained from the SOS approach.

Figure 9. Local Lyapunov level sets of the original PLL system.



The SOS techniques can be applied to the design of a controller. For the PLL system, it is desired to design a system which has a larger DoA or faster converging speed. Here, the objective is to find possible system parameters such that the same Lyapunov function is still valid and the system converges faster. This can be done by solving a semidefinite program.

Suppose V is a local Lyapunov function for the PLL system. Using the SOS technique, solve the following optimization problem:

$$\begin{aligned} &\max \alpha \\ &\text{s.t.} -(DV)f = s_1 + s_2(a - V) \\ &\quad -(DV)f - \alpha q = s_3 + s_4(b - V) \\ &\quad p(k) \leq 0 \end{aligned} \tag{6}$$

where $a,\ b \in \mathbb{R}_+$ specify the domain of the constraints and q is a positive definite performance polynomial specified by the user. $p(k)$ is a linear constraint of controller parameter k. As before, s_1, s_2, s_3, s_4 are SOS polynomials.

Since V is now a given function, the above constraints are linear in the controller parameters. The first constraint shows that V is a valid Lyapunov function in $sub(V-a)$. This constraint is used to specify the desired DoA to maintain. The second constraint along with the objective function will put an upper bound on the derivative of the Lyapunov function in $sub(V-b)$. Faster decreasing rate implies faster converging speed. This specifies the performance requirement of our system. The user could also put different performance requirements in different sub-level sets of V.

Besides dynamic performance constraints, noise bandwidth constraints will be applied as well. The noise bandwidth for this PLL system with PI controller has the following form [17]

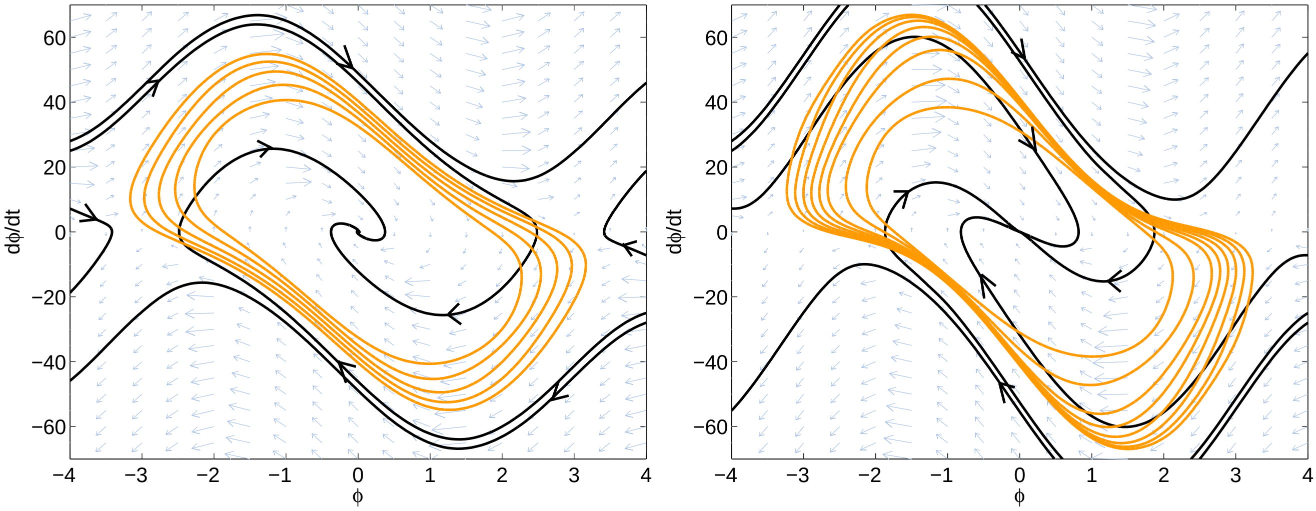
$$B_L = \frac{\omega_n}{2}\left(\zeta + \frac{1}{4\zeta}\right)$$

Assume $\zeta \geq 1,\ \omega_n \geq 1$. Then

$$B_L = \frac{\zeta\omega_n}{2} + \frac{\omega_n}{8\zeta} \leq -\frac{k_1}{4} - \frac{k_2}{8}$$

This linear upper bound will be used to find a set of controller parameters that have better dynamic performance while maintaining the same noise bandwidth.

Figure 10. Comparison of the domain-of-attraction. **Left**: linear design. **Right**: nonlinear design.



The system phase portrait as well as the estimated stable region are shown in Figure 10. The nonlinear design has $K = 1$, $\omega_n = 10.813$, and $\zeta = 1.3303$. The noise bandwidth is 8.1082 Hz, which is slightly higher than the noise bandwidth of the linear design, 7.9546 Hz. From the phase portrait, the nonlinear design has approximately 20% larger guaranteed domain-of-attraction. It can also be observed from the phase portrait that the nonlinear design has less overshoot than the linear design. This shows that the proposed method increases the system performance while not sacrificing too much of the noise rejection capability.

A Simulink model is used to compare the nonlinear designed PLL controller with the linear design. In this Simulink model, the sinusoidal input is collapsed by measurement noise and clock noise with zero mean and variances 0.1 and 0.0001, respectively. Figure 11 is the phase error of the two designs. It is clear that the nonlinear designed controller has a much faster convergence rate than the original design.

Figure 11. Phase error of Simulink simulation. **Left**: linear design. **Right**: nonlinear design.

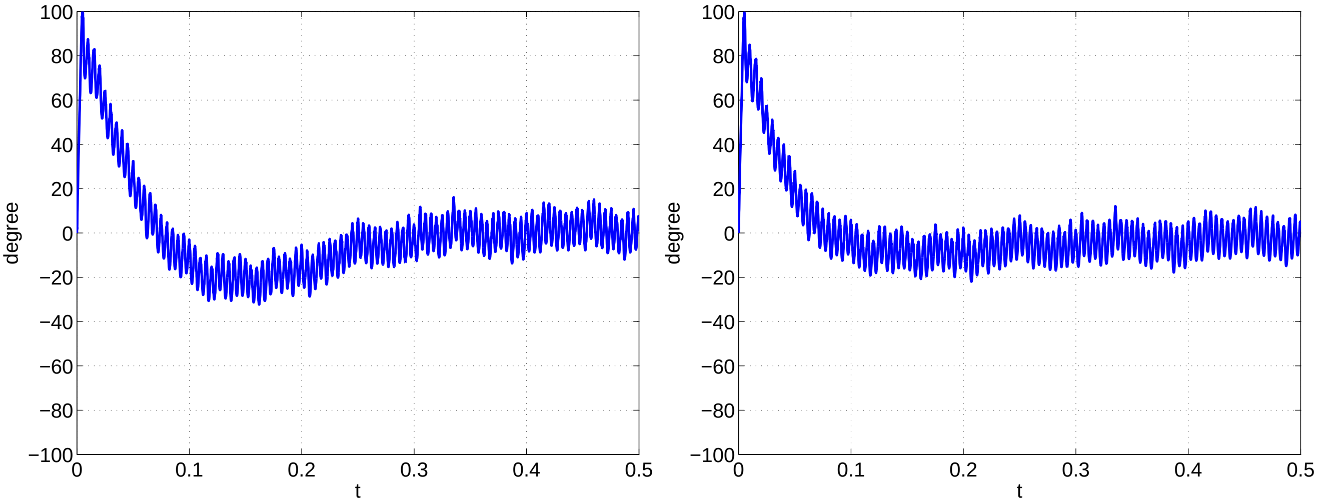
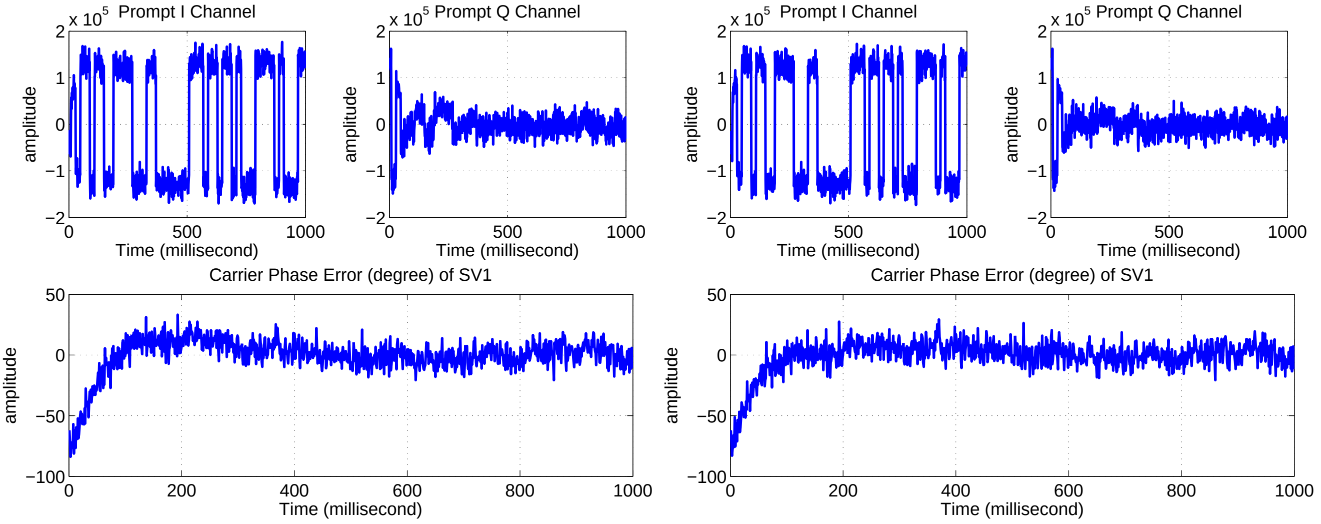


Figure 12. Real GPS experimental results. **Left**: linear design. **Right**: nonlinear design.



Both systems are tested on a NORDNAV R25 software GPS receiver. This receiver collects, down-converts and samples the GPS data by the front end, so that the collected GPS data can be post-processed repeatedly using different tracking-loop filter orders. The results are shown in Figure 12.

It can be seen that the original system has some overshoot and converges around 300 ms. The nonlinear design has much less overshoot and converges about five times faster than the original design.

6. Summary

In this paper, we presented a method of designing a PI controller of a PLL system. This design approach is based on the polynomial nonlinear model of the PLL system. This approach starts with the linear design of the controller and then estimates the DoA of the linear designed system to get the suitable local Lyapunov function for the system. The Lyapunov function is then used as the performance constraints to further refine the performance of the system outside the linear region. The DoA of the initial design can also be extended to get a better pull-in region of the PLL system. This approach gives us a way to design a fixed form controller for a nonlinear system.

Acknowledgements

A preliminary version of this paper was presented in 16th IEEE Mediterranean Conference on Control and Automation (MED'08) [23]. The first two authors were partially supported by by the Stanford URI *Architectures for Secure and Robust Distributed Infrastructures*, AFOSR DoD award number 49620-01-1-0365. The first author was also supported by National Science Council, Taiwan, under grant: NSC 96-2218-E-006-283-MY3.

References

1. Davison, E.J.; Kurak, E.M. A computational method for determining quadratic Lyapunov functions for non-linear systems. *Automatica* **1971**, *7*, 627–636.
2. Levin, A. An analytical method of estimating the domain of attraction for polynomial differential equations. *IEEE Trans. Autom. Control* **1994**, *39*, 2471–2475.
3. Chesi, G.; Tesi, A.; Vicino, A. On Optimal Quadratic Lyapunov Functions for Polynomial Systems. In *Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2002)*, South Bend, IN, USA, 12–16 August 2002.
4. Parrilo, P.A.; Lall, S. Semidefinite programming relaxations and algebraic optimization in control. *Eur. J. Control* **2003**, *9*, 307–321.
5. Papachristodoulou, A.; Prajna, S. On the Construction of Lyapunov Functions Using the sum of Squares Decomposition. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, USA, December 2002.
6. Prajna, S.; Papachristodoulou, A.; Parrilo, P. Introducing SOSTOOLS: A General Purpose Sum of Squares Programming Solver. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, USA, December 2002.
7. Parrilo, P. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 2000.

8. Jarvis-Wloszek, Z.W. Lyapunov Based Analysis and Controllers Synthesis for Polynomial Systems Using Sum-of-Squares Optimization. Ph.D. Thesis, University of California, Berkeley, CA, USA, 2003.
9. Tan, W.; Packard, A. Stability region analysis using polynomial and composite polynomial lyapunov functions and sum-of-squares programming. *IEEE Trans. Autom. Control* **2008**, *53*, 565–571.
10. Wang, T.; Lall, S.; West, M. Polynomial Level-Set Methods for Nonlinear Dynamical Systems Analysis. In *Proceedings of the 43rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, 28–30 September 2005.
11. Endo, T.; Chua, L.O. Chaos from phase-locked loops. *IEEE Trans. Circuits Syst.* **1988**, *35*, 987–1003.
12. Endo, T.; Chua, L.O. Chaos from phase-locked loops—Part II: Non-Hamiltonian case. *IEEE Trans. Circuits Syst.* **1988**, *36*, 255–633.
13. Endo, T.; Chua, L.O. Synchronization of chaos in phase-locked loops. *IEEE Trans. Circuits Syst.* **1991**, *38*, 1580–1588.
14. Bradley, E.; Straub, D.E. Using chaos to broaden the capture range of a phase-locked loop. *IEEE Trans. Circuits Syst.* **1993**, *40*, 808–818.
15. Liu, Z.F.; Man, K.F.; Tang, K.S.; Zhong, G.Q. Controlling Chaos in Phase-Locked Loop. In *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, IECON'98*, Aachen, Germany, 31 August–4 Sepember 1998; Volume 3, pp. 1560–1563.
16. Abramovitch, D.Y. Analysis and Design of a Third order Phase-Lock Loop. In *Proceedings of the IEEE Military Communications Conference, MILCOM 98*, Boston, MA, USA, 18–21 October 1998; Volume 2, pp. 455–459.
17. Gardner, F.M. *Phaselock Techniques*, 3rd ed.; Wiley-Interscience: Malden, MA, USA, 2005.
18. Schmüdgen, K. The *K*-moment problem for compact semi-algebraic sets. *Math. Ann.* **1991**, *289*, 203–206.
19. Putinar, M. Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.* **1993**, *42*, 969–984.
20. Jacobi, T.; Prestel, A. Distinguished representations of strictly positive polynomials. *J. Reine Angew. Math.* **2001**, *532*, 223–235.
21. Guckenheimer, J.; Holmes, P. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*; Springer-Verlag: Berlin, Germany, 2002.
22. Khalil, H. *Nonlinear Systems*, 3rd ed.; Prentice-Hall Inc.: New Jersey, NJ, USA, 2000.
23. Wang, T.C.; Lall, S.; Chiou, T.Y. Nonlinear Phase-Locked Loop Design using Semidefinite Programming. In *Proceedings of the 16th IEEE Mediterranean Conference on Control and Automation*, Ajaccio-Corsica, France, 25–27 June 2008; pp. 1640–1645.