Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005



# Numerical Optimization-based Extremum Seeking Control of LTI Systems

Chunlei Zhang and Raúl Ordóñez

*Abstract*— **An extremum seeking control scheme of LTI systems is introduced in this paper. The extremum seeking problem is treated as a numerical optimization with dynamic system constraints. The convergence and robustness of the extremum seeking scheme is guaranteed by the numerical optimization algorithm, where a detailed analysis based on the line search method is addressed. A simulation example is given to show the effectiveness of the proposed scheme with and without input disturbance.**

## I. INTRODUCTION

Traditional automatic control deals with the problem of *stabilization* of a known reference trajectory or set point, that is, so called "tracking" and "regulation" problem. The reference is often easily determined. However, in some occasions it can be very difficult to find a suitable reference value. For instance, the fuel consumption of a car depends on the ignition angle. It is necessary to change the ignition angle as the condition of the road and the load of the car change to maintain the optimal efficiency. Tracking a varying maximum or minimum of an output (performance) function is called *extremum seeking control* [1], which has two layers of meaning: first we need to seek an extremum of the output function; secondly, we need to be able to control (stabilize) the system and drive the output to that extremum.

Earlier investigations of extremum seeking control systems assume that the system is static, which can be justified if the time between the changes in the optimal reference is sufficiently long. It can be also approached as the plant is a cascade of a nonlinear static map and a linear dynamic system. The first rigorous proof [2] of local stability of perturbation based extremum seeking control scheme uses averaging analysis and singular perturbation, where a high-pass filter and slow perturbation signal are employed to derive the gradient information. The book [3] by Krstić et al. presents a systematic description of the perturbation based extremum seeking control and its applications. The recent progress in semi-global stability appears in [4].

Convergence analysis in gradient estimation based extremum seeking control is performed in [5]. The author of [6] classifies extremum seeking control as an extension of nonlinear programming problem. where readout map is defined as a steady state output function $g(\theta) := \lim_{t\to\infty} y(t)|_{\text{input fixed at }\theta}$. Thus, given experimentally determined waiting time between the measurers, nonlinear programming is successfully applied as an extremum seeking controller. An extremum seeking method with continuous time non-derivative optimizers is proposed in [7]. In order to mimic the static function optimization, a time-scale separation is applied to the dynamic system and the non-derivative optimizer, which is achieved by either accelerating the dynamics of the dynamic system or decelerating the non-derivative optimizer. An improved extremum seeking control based on sliding mode [8] is applied to systems with time delay or slow dynamics and to avoid the problem of excessive oscillation. An extremum seeking control problem is proposed and solved in [9] for a class of nonlinear systems with unknown parameters, where an explicit structure of the performance function is required.

In this paper, we combine numerical optimization algorithms and controllability directly to form a robust extremum seeking control scheme, where an line search method provides candidates of the unknown extremum step by step and a state feedback controller is designed to track the candidates. A problem statement is given in Section II, line search methods and controllability are reviewed in Section III. The analysis of convergence and robustness is performed in Section IV, and a simulation example is shown in Section V. Finally, Section VI concludes the paper.

## II. PROBLEM STATEMENT

Consider a SISO linear time invariant system

$$\dot{x} = Ax + Bu, \tag{1}$$
$$y = J(x), \tag{2}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the performance output, and $J : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. The matrix $A, B$ is given, however, the explicit form of the performance function and the minimum are not known. The following assumptions are made about the LTI system .

*Assumption 2.1:* The performance function $J(x)$ is continuously differentiable, bounded below, convex [1] and generally is unavailable to the designer.

*Assumption 2.2:* The LTI system is controllable and stable[2].

The first assumption guarantees the existence of the minimum, and that numerical optimization algorithms with first order global convergence property will produce a sequence

C. Zhang is with the Department of Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH 45469, USA zhangchz@notes.udayton.edu.

R. Ordóñez is with Faculty of Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH 45469, USA ordonez@ieee.org.

[1]Convexity is assumed here for simplicity. Without convexity, the convergence to global minimum will reduce to first order stationary point.

[2]Given an unstable but controllable system, we can perform pole placement to form a stable closed loop system.

converging to the global minimizer of the performance function. Therefore, the goal of extremum seeking control is to design an input (control law) $u$ based on the output measurements and state measurements to seek the unknown global minimum of the performance function $J(x)$ and drive the state to the vicinity of the minimizer. Moreover, from the point of view of numerical optimization, the extremum seeking control problem can be stated as:

$$\min_{x\in\mathbb{R}^n} \quad J(x) \quad \text{subject to} \quad \dot{x} = Ax + Bu. \tag{3}$$

This is a constrained optimization problem, which is different from the traditional algebraic constraints. Now the state $x$ is feasible if it is a solution of the dynamic system. In the case when $(A,B)$ is controllable, there always exists an input $u$ that transfers $x$ to any where in $\mathbb{R}^n$ in a finite time [10], which justifies the necessity for the second assumption. Although controllable dynamic system constraints do allow $x$ to be anywhere in the state space where the numerical optimizer wants, the way in which $x$ reaches the particular place is determined by the dynamic system.

## III. MATHEMATICAL PRELIMINARIES

### A. Line Search Methods

For unconstrained optimization problem $\min_{x\in\mathbb{R}^n} J(x)$, each iteration of a line search method computes a search direction $p_k \in \mathbb{R}^n$ and then decides how far to move along that direction. The iteration is given by

$$x_{k+1} = x_k + \alpha_k p_k, \tag{4}$$

where the positive scalar $\alpha_k$ is called the *step length* and requires $p_k$ to be a *descent direction* (one for which $p_k^\top \nabla J(x_k) < 0$), because this property guarantees that the function $J$ can be reduced along this direction. The steepest descent direction $p_k = -\nabla J(x_k)$ is the most obvious choice for search direction.

Given a descent direction $p_k$, we face a tradeoff in choosing step length $\alpha_k$ that gives a substantial reduction of $J$ and not spending too much time making the choice. It can be found by approximately solving the following one-dimensional minimization problem:

$$\min_{\alpha>0} \phi(\alpha) = \min_{\alpha>0} J(x_k + \alpha p_k). \tag{5}$$

An *exact* minimization of $\phi(\alpha)$ to find $\alpha$ is expensive and sometimes unnecessary. More practical strategies perform an *inexact* line search to identify a step length that achieves adequate reduction in $J$ at minimal cost. In particular, the *Armijo condition*

$$J(x_k + \alpha_k p_k) \le J(x_k) + c_1 \alpha_k p_k^T \nabla J(x_k) \tag{6}$$

prevents steps that are too long via a sufficient decrease criterion, while the *Wolfe condition*

$$p_k^T \nabla J(x_k + \alpha_k p_k) \ge c_2 p_k^T \nabla J(x_k), \tag{7}$$

prevents steps that are too short via a curvature criterion, with $0 < c_1 < c_2 < 1$. The restriction $c_2 > c_1$ ensures that acceptable points exist. Moreover, in order to avoid a poor choice of descent directions, an *angle condition* is set up to enforce a uniform lower bound on the angle between vector $p_k$ and $-\nabla J(x_k)$, that is

$$\cos\theta_k = \frac{-p_k^\top \nabla J(x_k)}{\|p_k\|\|\nabla J(x_k)\|} \ge c_3 > 0 \tag{8}$$

where $c_3$ is independent of $k$. The following is the first-order global convergence result for line search methods [11], [12].

*Theorem 3.1:* Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$ and be bounded below. And suppose that $\nabla J$ is Lipschitz continuous with constant $L$; that is, $\|\nabla J(y) - \nabla J(x)\| \le L\|y - x\|$ for all $x, y \in \mathbb{R}^n$. If the sequence $\{x_k\}$ satisfies conditions (6), (7) and (8), then

$$\lim_{k\to\infty} \|\nabla J(x_k)\| = 0.$$

The following lemmas will be used in the robustness analysis of the extremum seeking control scheme.

*Lemma 3.2 (Descent Lemma [13]):* Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$. And suppose that $\nabla J$ is Lipschitz continuous with constant $L$. Then for $x, y \in \mathbb{R}^n$,

$$J(x+y) \le J(x) + y^\top \nabla J(x) + \frac{L}{2}\|y\|^2.$$

*Lemma 3.3:* Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$. And suppose that $\nabla J$ is Lipschitz continuous with constant $L$. Let $\alpha_k, p_k$ be the step length and descent direction, then

$$J(x_k + \alpha_k p_k) - J(x_k) \le -\frac{c}{2L}\|\nabla J(x_k)\|^2 \cos^2\theta_k,$$

where $c = 1$ for exact line search, and $c = 2c_1(1 - c_2)$ for inexact line search satisfying conditions (6) and (7), $\theta_k$ represents the angle between vector $p_k$ and $-\nabla J(x_k)$.

*Proof:* First, for exact line search, $\alpha_k$ is the solution of Equation (5). From the Descent Lemma 3.2, we have $J(x_k + \alpha p_k) \le J(x_k) + \alpha p_k^\top \nabla J(x_k) + \frac{\alpha^2}{2} L\|p_k\|^2$ valid for all $\alpha > 0$. Letting $\bar{\alpha} = -\frac{p_k^\top \nabla J(x_k)}{L\|p_k\|^2} > 0$, then

$$\begin{aligned} J(x_k + \alpha_k p_k) - J(x_k) &\le J(x_k + \bar{\alpha} p_k) - J(x_k) \\ &\le \bar{\alpha} p_k^\top \nabla J(x_k) + \frac{\bar{\alpha}^2}{2} L\|p_k\|^2 \\ &= -\frac{(p_k^\top \nabla J(x_k))^2}{L\|p_k\|^2} + \frac{L\|p_k\|^2}{2}\frac{(p_k^\top \nabla J(x_k))^2}{(L\|p_k\|^2)^2} \\ &= -\frac{1}{2L}\|\nabla J(x_k)\|^2 \cos^2\theta_k. \end{aligned}$$

Second, for inexact line search, $\alpha_k$ satisfies conditions (6) and (7). From Lipschitz condition we have $p_k^\top[\nabla J(x_k + \alpha_k p_k) - \nabla J(x_k)] \le \|p_k\|\|\nabla J(x_k + \alpha_k p_k) - \nabla J(x_k)\| \le \alpha_k L\|p_k\|^2$. Then from (7), we have

$$-\alpha_k L\|p_k\|^2 \le p_k^\top[\nabla J(x_k) - \nabla J(x_k + \alpha_k p_k)] \le (1 - c_2) p_k^\top \nabla J(x_k).$$

That is, $-\alpha_k\|p_k\| \le -\frac{1-c_2}{L}\|\nabla J(x_k)\| \cos\theta_k$. Finally from (6),

$$\begin{aligned} J(x_k + \alpha_k p_k) - J(x_k) &\le c_1 \alpha_k p_k^\top \nabla J(x_k) \\ &= -c_1 \alpha_k \|p_k\|\|\nabla J(x_k)\| \cos\theta_k \\ &\le -\frac{c}{2L}\|\nabla J(x_k)\|^2 \cos^2\theta_k \end{aligned}$$

■

Since $0 < c_1 < c_2 < 1$ is required to ensure the feasibility of inexact line search, we will have $c = 2c_1(1-c_2) < 1$. This observation is consistent for the upper bound results in the above two lemmas. That is, we always expect exact line search to have more decrease along the search direction than inexact line search.

### B. Controllability

Following theorem can be found as Theorem 6.1 in [10].

*Theorem 3.4:* For LTI system $\dot{x} = Ax + Bu$, if $(A,B)$ is controllable, then for any initial state $x(t_0) = x_0$ and any final state $x_1$ there exists an input $u$ that transfers $x_0$ to $x_1$ in a finite time. For single input LTI system, the input

$$u(t) = -B^\top e^{A^\top(t_1-t)} W_c^{-1}(t_1)[e^{A(t_1-t_0)}x_0 - x_1] \quad (9)$$

will transfer $x_0$ to $x_1$ at time $t_1$, where $W_c$ is the *controllability Gramian* and can be expressed as

$$W_c(t_1) = \int_{t_0}^{t_1} e^{A(t_1-\tau)}BB^\top e^{A^\top(t_1-\tau)}d\tau = \int_0^{t_1-t_0} e^{A\tau}BB^\top e^{A^\top\tau}d\tau.$$

## IV. EXTREMUM SEEKING CONTROL

### A. An Extremum Seeking Scheme

Now, given a controllable LTI system, we can combine a numerical optimization algorithm and the state tracking controller (9) to form an extremum seeking control scheme that automatically minimizes the convex performance function $y = J(x)$. The basic scheme is outlined as follows.

**Extremum Seeking Scheme**

1) **Step 0**. Given $x_0, t_0 = 0$, set $\varepsilon_0$ and $k := 0$
2) **Step 1**. Use an exact/inexact line search method to produce $x_{k+1} = x_k + \alpha_k p_k$, where the sequence $\{x_k\}$ will converge to the global minimum of the convex performance function. If $\|\nabla J(x_k)\| < \varepsilon_0$, then stop.
3) **Step 2**. Choose $\delta_k$, let $t_{k+1} = t_k + \delta_k$, and the input during $t_k \le t \le t_{k+1}$ is

$$u(t) = -B^\top e^{A^\top(t_{k+1}-t)} W_c^{-1}(t_{k+1})[e^{A\delta_k}x_k - x_{k+1}], \quad (10)$$

where

$$W_c(t_{k+1}) = \int_0^{\delta_k} e^{A\tau}BB^\top e^{A^\top\tau}d\tau \quad (11)$$

4) **Step 3**. Set $k \leftarrow k+1$. Go to step 1.

### B. Convergence Analysis

*Theorem 4.1:* Assume that LTI system (1) is controllable, and the performance function (2) $J : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on $\mathbb{R}^n$, bounded below and convex. Furthermore, suppose that $\nabla J$ is Lipschitz continuous with constant $L$. If the extremum seeking scheme in Section IV-A is applied, the performance function will be globally minimized as $t \longrightarrow \infty$.

*Proof:* From the extremum seeking scheme, any line search method with first order global convergence will produce a descent sequence $\{x_k\}$ that converges to the global minimum of the performance function as $k \to \infty$. And the controller (10) interpolates between the $\{x_k\}$ precisely within finite time $\delta_k$ from $x_k$ to $x_{k+1}$, thus the performance function achieves the global minimum as $t \longrightarrow \infty$. ■

*Corollary 4.2:* In addition to the assumptions in Theorem 4.1, let $x^*$ denote the unknown global minimum of the performance function. If $J$ is strongly convex on $\mathbb{R}^n$, and steepest descent algorithm with exact line search is used in the extremum seeking scheme. Then the state will converge to the $\varepsilon$ neighborhood of $x^*$ at most time $t = \sum_{k=1}^N \delta_k$, where $N = \frac{\log((f(x_0)-f(x^*))/\varepsilon)}{\log(1/h)}$ for some $0 < h < 1$.

*Proof:* If $J$ is strongly convex on $\mathbb{R}^n$, which means there exist constants $Q, q > 0$ such that [14]

$$qI \preceq \nabla^2 f(x) \preceq QI$$

for all $x \in \mathbb{R}^n$. And if a steepest descent algorithm is used in the extremum seeking scheme, letting $h = 1 - q/Q < 1$, we must have $f(x_k) - f(x^*) \le \varepsilon$ after at most

$$N = \frac{\log((f(x_0)-f(x^*))/\varepsilon)}{\log(1/h)}$$

iterations of the steepest descent method with exact line search [14]. Thus, the state $x$ will converge to the $\varepsilon$ neighborhood of $x^*$ at most $t = \sum_{k=1}^N \delta_k$. ■

### C. Robustness Analysis

The main restriction of Theorem 4.1 is that the controller needs to interpolate precisely between the two candidates $x_k$ and $x_{k+1}$. In practical applications, noisy output or state measurements, input disturbance or saturation, nonlinear system dynamics and computational error will be detrimental to this theoretical result. That is, the controller $u$ may be only able to transfer states to the neighborhood of $x_{k+1}$. In fact, the line search method will produce a sequence of $\{\hat{x}_k\}$, where $x_0 = \hat{x}_0$ and

$$x_{k+1} = \hat{x}_k + \alpha_k p_k, \quad \hat{x}_{k+1} = x_{k+1} + e_{k+1}.$$

We will assume that $\|e_k\|$ to be bounded, which generally is the case for input disturbance or computational error given a stable system. For example, let $\hat{u}(t) = u(t) + \Delta u(t)$, where $u(t)$ is given as in (10) to transfer from $\hat{x}_k$ to $x_{k+1} = \hat{x}_k + \alpha_k p_k$, then

$$\begin{aligned}\hat{x}_{k+1} &= e^{A\delta_k}\hat{x}_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}B\hat{u}(\tau)d\tau \\ &= e^{A\delta_k}\hat{x}_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bu(\tau)d\tau \\ &\quad + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}B\Delta u(\tau)d\tau \\ &= x_{k+1} + e_{k+1},\end{aligned}$$

where $e_{k+1} = \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}B\Delta u(\tau)d\tau$ will be bounded if the system is stable and the input disturbance is bounded. Thus, we will hope that a well designed line search method will convey its robustness to the extremum seeking scheme. That is, the new sequence $\{\hat{x}_k\}$ may be able to converge to the neighborhood of the minimum given the error $e_k$ is bounded. Indeed, that is shown in the following theorem.

*Theorem 4.3:* Suppose that LTI system (1) is controllable and stable, and the performance function (2) $J : \mathbb{R}^n \to \mathbb{R}$

is continuously differentiable on $\mathbb{R}^n$, bounded below and convex. Furthermore, suppose that $\nabla J$ is Lipschitz continuous with constant $L$. If the extremum seeking scheme in Section IV-A is applied, where the controller (10) is only able to transfer the current state $\hat{x}_k$ to $\hat{x}_{k+1} = x_{k+1} + e_{k+1}$ and $x_{k+1} = \hat{x}_k + \alpha_k p_k$. The new sequence $\{\hat{x}_k\}$ is a descent sequence, that is

$$J(\hat{x}_{k+1}) < J(\hat{x}_k),$$

given

$$\|e_{k+1}\| \leq \frac{c\|\nabla J(\hat{x}_k)\|^2 \cos^2\theta_k}{L(\sqrt{\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k} + \|\nabla J(x_{k+1})\|)}, \tag{12}$$

where $c = 1$ for exact line search and $c = 2c_1(1-c_2)$ for inexact line search satisfying conditions (6) and (7).

*Proof:* Now, for line search method at step $k+1$ and from Lemmas 3.2 and 3.3 we have

$$\begin{aligned}
&J(\hat{x}_{k+1}) - J(\hat{x}_k) = J(x_{k+1} + e_{k+1}) - J(\hat{x}_k)\\
&\leq J(x_{k+1}) + \nabla J(x_{k+1})^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 - J(\hat{x}_k),\\
&\leq \nabla J(x_{k+1})^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 - \frac{c}{2L}\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k\\
&\leq \frac{L}{2}\Big[\|e_{k+1}\|^2 + 2\frac{\|\nabla J(x_{k+1})\|}{L}\|e_{k+1}\| - \frac{c}{L^2}\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k\Big]\\
&= \frac{L}{2}\Big[\Big(\|e_{k+1}\| + \frac{\|\nabla J(x_{k+1})\|}{L}\Big)^2\\
&\quad - \frac{1}{L^2}\Big(\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k\Big)\Big].
\end{aligned}$$

That is, if we have

$$\begin{aligned}
\|e_{k+1}\| &< \frac{1}{L}(\sqrt{\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k} - \|\nabla J(x_{k+1})\|)\\
&= \frac{c\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k}{L(\sqrt{\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2\cos^2\theta_k} + \|\nabla J(x_{k+1})\|)},
\end{aligned}$$

we can obtain $J(\hat{x}_{k+1}) - J(\hat{x}_k) < 0$. ■

Although the bound (12) is very conservative, it can give us some insights of the robustness of the extremum seeking scheme. First the exact line search allows a larger error bound than the inexact line search. Second, we can see that the bound is an increasing function of $\|\nabla J(\hat{x}_k)\|$. That is, when $\hat{x}_k$ is far away from the minimizer of the performance function, we will expect the gradient to be large and thus the error the scheme can tolerate is also large. This observation implies that the extremum seeking scheme will be very robust until the gradient converges to some invariant set, which is illustrated in the following corollary.

*Corollary 4.4:* In addition to the assumptions in Theorem 4.3, if a steepest descent algorithm is used in the extremum seeking scheme IV-A. And assuming $\|e_k\| \leq e_L$, then we will have the gradient of the sequence $\{\hat{x}_k\}$ converges to the invariant set

$$\|\nabla J(\hat{x}_k)\| \leq \frac{Le_L}{c}[\sqrt{(1+\alpha_k L)^2 + c} + (1+\alpha_k L)], \tag{13}$$

where $c = 1$ for exact line search and $c = 2c_1(1-c_2)$ for inexact line search satisfying conditions (6) and (7).

*Proof:* Now we have $\cos\theta_k = 1$ for steepest descent algorithm, and from Equation (12), as long as

$$\frac{1}{L}(\sqrt{\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2} - \|\nabla J(x_{k+1})\|) > e_L,$$

we will always have $J(\hat{x}_{k+1}) < J(\hat{x}_k)$. So we can find a conservative bound on $\nabla J(\hat{x}_k)$ given the error bound $e_L$.

For steepest descent method, $x_{k+1} = \hat{x}_k - \alpha_k \nabla J(\hat{x}_k)$. And from Lipschitz condition,

$$\begin{aligned}
\|\nabla J(x_{k+1})\| &\leq \|\nabla J(x_{k+1}) - \nabla J(\hat{x}_k)\| + \|\nabla J(\hat{x}_k)\|\\
&\leq (1+\alpha_k L)\|\nabla J(\hat{x}_k)\|.
\end{aligned}$$

Now the bound can be found via

$$\begin{aligned}
&\frac{1}{L}(\sqrt{\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2} - \|\nabla J(x_{k+1})\|) \leq e_L\\
&\Leftrightarrow \sqrt{\|\nabla J(x_{k+1})\|^2 + c\|\nabla J(\hat{x}_k)\|^2} \leq Le_L + \|\nabla J(x_{k+1})\|\\
&\Leftrightarrow c\|\nabla J(\hat{x}_k)\|^2 \leq 2Le_L(1+\alpha_k L)\|\nabla J(\hat{x}_k)\| + L^2 e_L^2\\
&\Leftrightarrow [\sqrt{c}\|\nabla J(\hat{x}_k)\| - \frac{Le_L}{\sqrt{c}}(1+\alpha_k L)]^2 \leq L^2 e_L^2[(1+\alpha_k L)^2/c + 1]\\
&\Leftrightarrow \|\nabla J(\hat{x}_k)\| - \frac{Le_L}{c}(1+\alpha_k L) \leq \frac{Le_L}{c}\sqrt{(1+\alpha_k L)^2 + c}
\end{aligned}$$

Thus, we will have the gradient of the sequence $\{\hat{x}_k\}$ converges to the invariant set

$$\|\nabla J(\hat{x}_k)\| \leq \frac{Le_L}{c}[\sqrt{(1+\alpha_k L)^2 + c} + (1+\alpha_k L)].$$

■

Observed from Equation (13), a diminishing step length $\alpha_k$ is preferred later on to decrease the bound of the invariant set. As $\alpha_k \to 0$, the bound converges to $\big(\frac{1}{c} + \sqrt{\frac{1}{c^2} + \frac{1}{c}}\big)Le_L$. This is again coincident with theory of numerical optimization, where generally a diminishing step length is required for the algorithms to converge to the minimum. And if there is no error between $\hat{x}_k$ and $x_k$, we will see that the gradient converge to zero. Moreover, exact line search can achieve a smaller bound than the inexact line search.

## V. EXAMPLES

Consider a second order stable LTI system in its controllable canonical form. Let $x = [x_1, x_2]^\top$, $x_k = [x_1^k, x_2^k]^\top$

$$\begin{aligned}
\dot{x} &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,\\
y &= J(x) = 5x_1^2 + x_2^2 + 4x_1x_2 - 14x_1 - 6x_2 + 20.
\end{aligned}$$

Here we postulate the explicit form of the performance function is only for simulation purpose. The performance function $J(x)$ has its minimizer at $x^* = (1,1)$ and $J(1,1) = 10$. The explicit form of the function and its minimum are both unknown to the designer. In the extremum seeking control, we only need to access the function value $y$ and its gradient[3] for arbitrary $x \in \mathbb{R}^n$. For steepest descent algorithm with

[3]The gradient information is required since we use steepest descent algorithm in the extremum seeking scheme. We can use derivative free optimization methods to avoid the requirement of gradient information, and the same analysis of the extremum seeking scheme based on steepest descent can be expanded to the derivative free methods.

exact line search [15], we can compute the search direction $p_k = -\nabla J(x_1^k, x_2^k) = [-10x_1^k - 4x_2^k + 14, -2x_2^k - 4x_1^k + 6]^\top = [p_1^k, p_2^k]^\top$. Then in this example, we can derive an explicit expression of the step length $\alpha_k = \text{argmin}_\alpha f(x_k + \alpha p_k) = \frac{(p_1^k)^2 + (p_2^k)^2}{2(5(p_1^k)^2 + (p_2^k)^2 + 4p_1^k p_2^k)}$. Note that generally speaking, exact line search is not possible, however, this example here is only for simplicity, and similar results are expected for inexact line search as well as other optimization algorithms.

Now we apply the extremum seeking scheme in Section IV-A, where steepest descent algorithm of exact line search is used. Given $\delta_k = 2$, $x_0 = [-10, 10]$, the simulation results are shown in Figure 1 with controller (10) interpolating between $x_k$ and $x_{k+1}$ precisely within finite time, the performance function (Figure 1 (a)) approaches to its minimum at $J(1,1) = 10$. The steepest descent algorithm produces a sequence $\{x_k\}$ as a guideline of the controller. The trajectory between $x_k$ and $x_{k+1}$ is shaped by the dynamical system constraints. This can be viewed clearly in Figure 1 (d), where the blue circle represents the $\{x_k\}$ and the red dashed line represents the state trajectory. The choice of $\delta_k$ is rather heuristic in this example, which is actually a very important design factor. We can see that the smaller $\delta_k$ is the larger control force we need to fulfill the transfer. Thus, $\delta_k$ should be chosen appropriately to not exceed the practical limit on the control force. There is always a tradeoff between the extremum seeking time and the control gain. However, the robustness analysis of the extremum seeking scheme provides additional flexibility to the choice of $\delta_k$ since the algorithm can accommodate certain error especially in the beginning. That is, we can allow a fast but loose tuning in the beginning and later on we may need a slow but fine tuning to get even closer to the extremum.

Moreover, a random disturbance uniformly distributed with amplitude 1 is introduced to the input. The simulation results are shown in Figure 2. Now, the controller is not able to reach the desired destination precisely. For example, at the first step, the controller cannot interpolate exactly between the initial position $x_0 = (-10, 10)$ and the desired destination $x_1$, instead it arrives at $\hat{x}_1$ due to the input disturbance. Then the next search destination is generated as $x_2 = \hat{x}_1 - \alpha_k \nabla J(\hat{x}_1)$, while again the state only arrives at $\hat{x}_2$, therefore, eventually we will still have a descent sequence $\{\hat{x}_k\}$ as long as the error $e_k$ satisfies certain bound (12). The comparison of $\{x_k\}$ and $\{\hat{x}_k\}$ can be seen in Figure 2 (d), where the blue circle represents the $\{x_k\}$, magenta square denotes the $\{\hat{x}_k\}$ and the red dashed line is the state trajectory. The performance function again approaches its minimum but with a longer oscillation due to the disturbance, as shown in to Figure 2 (a). This result shows that the robustness analysis in Section IV-C is conservative.

## VI. CONCLUSIONS

In this paper, we successfully incorporate numerical optimization algorithms into an extremum seeking control scheme for controllable LTI systems. The convergence of the proposed extremum seeking scheme is guaranteed given

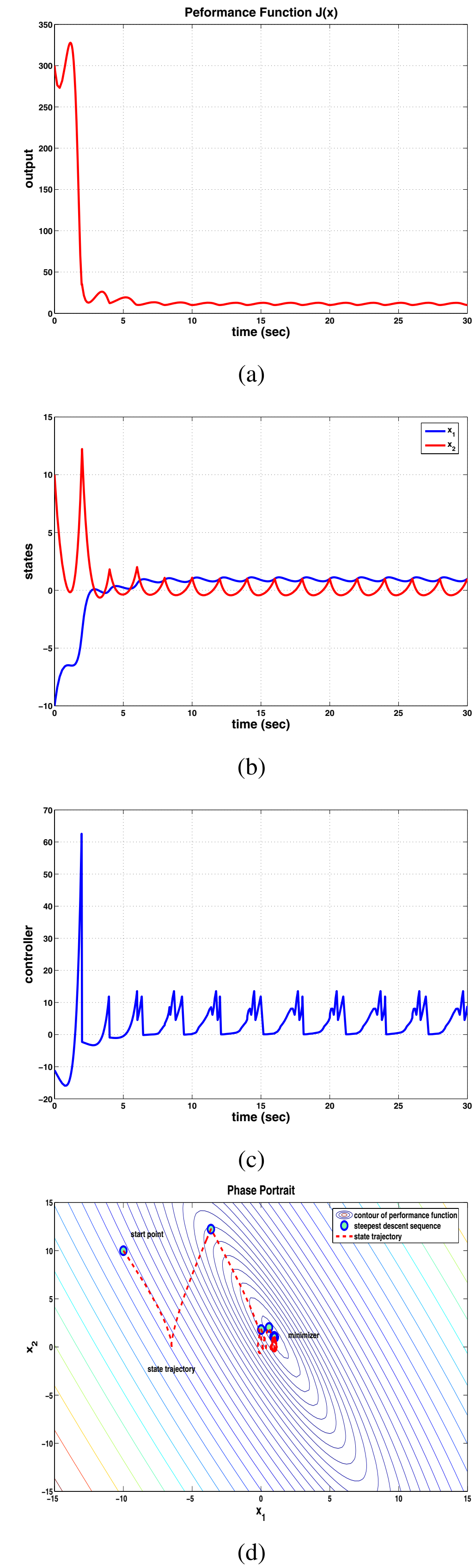
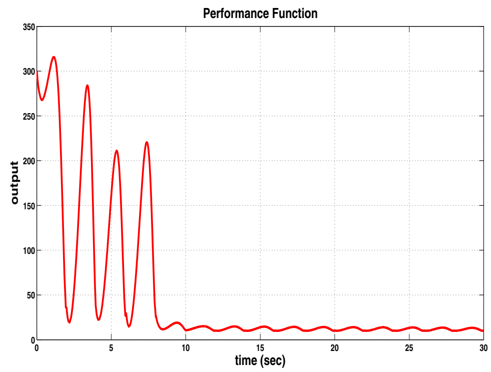
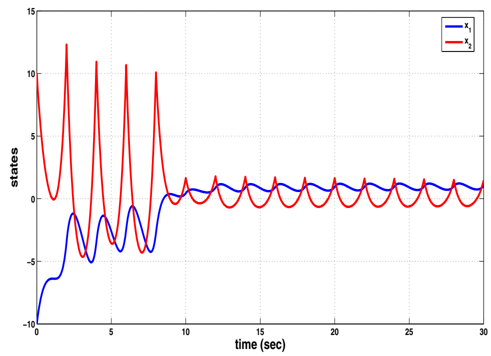


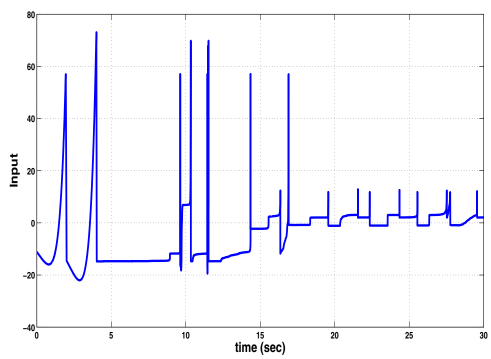

Fig. 1. Extremum Seeking Control for LTI system: (a) Performance Function; (b) States; (c) Control Input; (d) Phase Portrait, steepest descent sequence $\{x_k\}$ over the contour of the performance function.



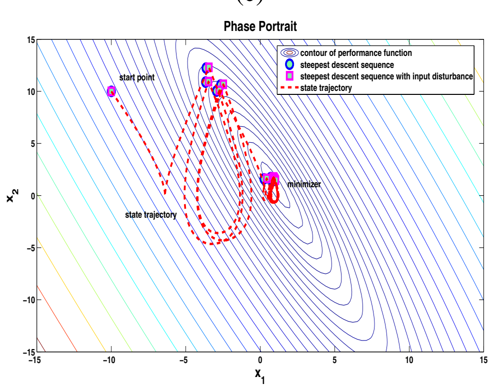

(a)




(b)




(c)




(d)

Fig. 2. Extremum Seeking Control for LTI system with input disturbance (a) Performance Function; (b) States; (c) Control Input; (d) Phase Portrait, steepest descent sequence $\{x_k\}, \{\hat{x}_k\}$ over the contour of the performance function.

the optimization algorithm is globally convergent and the controller is able to transfer state between the guideline $\{x_k\}$ produced by the optimization algorithm. We also analyze the robustness of the extremum seeking scheme, which is inherited from the robustness of the line search method. In particular, the steepest descent algorithm with exact line search is used in the simulation to show the effectiveness of the method. Therefore, in order to fulfill a better extremum seeking controller, a more robust optimization algorithm is needed. Also we would like to design a robust controller to deal with various disturbances, noises, rather than put all the burden on the optimization algorithms.

## VII. ACKNOWLEDGMENTS

This research is generally supported by Dayton Area Graduate Studies Institute (DAGSI). The authors gratefully acknowledge Dr. Ruihua Liu for his valuable discussions.

## REFERENCES

[1] K. J. Åström and Björn Wittenmark, *Adaptive Control*. Boston, MA: Wiley, 1994.
[2] M. Krstić and H.-H. Wang, "Design and stability analysis of extremum seeking feedback for general nonlinear systems," *Automatica*, vol. 36, no. 2, pp. 595–601, 2000.
[3] K. B. Ariyur and M. Krstić, *Real-Time Optimization by Extremum-Seeking Control*. Hoboken, NJ: Wiley-Interscience, 2003.
[4] Y. Tan, D. Nesic, and I. Mareels, "On non-local stability properties of extremum seeking control," in *Proceedings of the 16th IFAC World Congress*, 2005.
[5] R.N.Banavar, D.F.Chichka, and J.L.Speyer, "Convergence and synthesis issues in extremum seeking control," in *Proceedings of the American Control Conference*, vol. 1, pp. 438–443, 2000.
[6] A. R. Teel and D. Popović, "Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming," in *Proceedings of the American Control Conference*, vol. 3, pp. 2394–2399, 2001.
[7] Nusawardhana and S. H. Żak, "Extremum seeking using analog nonderivative optimizers," in *Proceedings of the American Control Conference*, vol. 4, pp. 3242–3247, 2003.
[8] H. Yu and Ümit Özgüner, "Extremum-seeking control strategy for ABS system with time delay," in *Proceedings of the American Control Conference*, vol. 5, pp. 3753–3758, 2002.
[9] M. Guay and T. Zhang, "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties," *Automatica*, vol. 39, no. 7, pp. 1283–1293, 2003.
[10] C.-T. Chen, *Linear System Theory and Design*. New York Oxford: Orford University Press, 1999.
[11] J. Nocedal and S. Wright, *Numerical Optimization*. Berlin, Heidelberg, New York: Springer-Verlag, 1999.
[12] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: new perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, 2003.
[13] D. P. Bertsekas, *Nonlinear Programming*. Belmont, Massachusetts: Athena Scientific, 1995.
[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[15] R. Freund, "Lecture notes of 15.084j / 6.252j nonlinear programming, spring 2004," 2004. MIT OpenCourseWare.