Extremum Seeking Control with Sporadic Packet Transmission for Networked Control Systems

Upeka Premaratne, Saman Halgamuge, Ying Tan and Iven M. Y. Mareels.

Abstract—Extremum Seeking Control (ESC) is a data-driven optimization technique that can steer a dynamic plant towards an extremum of an unknown but measurable, input to steady-state map. In the context of Networked Control Systems (NCS) a new implementation method for ESC inspired by the well known Luus-Jaakola algorithm is proposed. The main motivation is to minimize the communication burden associated with the search phase of ESC. In the proposed method the controller only requires a notification of a change registered at the sensor, rather than the full information available at the sensor. This event based approach leads to sporadic packet transmission. In addition the proposed method is able to directly account for constraints whilst seeking for the desired extremum. The constraints may be of the inequality or equality type. The algorithm's behavior is illustrated on a networked water pump control system.

Keywords—Extremum Seeking Control, Networked Control Systems, Sporadic Sampling, Luus-Jaakola Algorithm

I. INTRODUCTION

Extremum Seeking Control (ESC) is a data-driven optimization technique capable of maintaining a dynamic plant at the vicinity of an optimal operating point in real time. Unlike optimal control, ESC requires minimal *a-priori* knowledge of the plant since it manipulates the control input using the measured plant output. An example for such a control system is ignition spark timing where there is an optimum spark time for maximizing the torque output of the internal combustion engine [1] (Section I-A). The prevalent form of ESC uses a dither (i.e., excitation) signal to explore the search space for the optimum by perturbing the control inputs. This can be periodic [2] [3] [4] or stochastic [5] [6] for a continuous or discrete time system. In the input perturbed framework of [7] an iterative optimization algorithm can be used for extremum seeking in the form of a discrete system that samples the output. For example, Shubert's algorithm for SISO systems [8] and DIRECT for MISO systems [9] have been used. Both of these algorithms eventually converge to a small neighborhood of the global optimum. The *state optimizing* approach of Zhang and Ordonez [10] optimizes the state (instead of perturbing the input) using gradient descent, metaheuristics [11] [12] or line search [13].

The stability of the control loop has been established for different types of ESC schemes [2] [3] [5] [14] [15]. ESC has a

long history [4] [16] [17] and has been applied to a wide range of engineering applications such as handling thermoacoustic instability [18], biorectors [19] and cam timing [1] [20]. Recent areas of interest for ESC have been maximum power point tracking in photovoltaic cells [21] [22], parallel optimization of chillers [23], wind power optimization [24] and control of type 1 diabetes [25].

Despite the large body of research into ESC very limited work has been done in the area of minimizing the network traffic generated when the feedback path of ESC is implemented over a communication network [26] [27]. Previous implementations of ESC over communication networks require periodic sampling and use estimation methods to reduce the error due to channel unreliability or localized communication [28] [29] [30] instead of reducing the traffic generated by the control system itself.

A. Motivating Example and Problem Statement

Minimizing the network traffic generated by the feedback path of an ESC implementation to reduce its contribution to congestion of the communication network is an interesting research problem. This is due to the numerous examples of engineering problems that require the feedback sensor to be placed at a significant distance from the ESC controller. In such systems, implementing the feedback path over a communication network (as in Fig. 1) is essential. Examples for such ESC systems include torque maximization in naval engines where the torque sensor of the propeller shaft is often located at a significant distance from the timing controller [31]. A similar situation can also be found in exhaust gas scrubbers where chemical feeds have to be optimized based upon input data from sensors located at a distance. Despite the massive recent increase in bandwidth of communication networks [32], emerging trends such as seamless integration of enterprise and automation networks [33] and Internet of Things (IoT) [34] where more and more entities are networked, result in increased utilization and consequential congestion which can lead to undesirable results such as delay, delay variation (jitter) and packet drops [35]. These consequences can cause significant stability issues in control systems [36] [37]. Thus, they motivate the minimization of bandwidth requirements of such control systems by reducing the effective sampling rate through *sporadic* transmission of the data over the communication network [38] [39] [40].

In periodically [2] [3] [4] or stochastically [5] [6] perturbed ESC, the correlation between the input dither and plant output shifts the averaged (i.e., integrated) input towards the optimum operating point [4]. This approach requires synchronized periodic sampling between the plant output and dither to establish such a correlation. It is also bandwidth intensive but a significant bandwidth reduction can be achieved using event

U. Premaratne is with the Department of Electronic and Telecommunication Engineering, University of Moratuwa, Katubedda, Moratuwa 10400, Sri Lanka e-mail:upeka@uom.lk

S. Halgamuge and Y. Tan are with the Melbourne School of Engineering, University of Melbourne, Parkville, VIC 3010, Australia e-mail:{saman,yingt}@unimelb.edu.au

I. Mareels is with IBM Research Australia and an Honorary Professor at the Melbourne School of Engineering, e-mail:imareels@au1.ibm.com

based sampling and reconstruction of the plant output [27]. In iterative optimization algorithms (for example [7]) and state optimization methods [10] [11] [12] a measurement of the plant output has to be transmitted to the controller for each *periodically executed* iteration. Furthermore, all of the mentioned methods have a limited ability to handle constraints (Section III-A2). Therefore, all of the above problems motivate this work, which is formulated as a networked form of ESC with the aim of reducing transmitted data and handling constraints efficiently.

B. Contribution

The novelties of the paper include a new MISO ESC scheme where the controller only needs a notification of a change in the measured optimum output of the plant to update the estimate of the optimum control input. Hence, it does not need to transmit a sampled value of the plant output for each iteration as in [7] (and derivative methods [8] [9]) or state optimizing ESC [10] [11] [12]. This significantly reduces network traffic. The ESC scheme extends the framework of [7] using a special case of the Luus-Jaakola (LJ) algorithm [41]. This new ESC scheme is called the Luus-Jaakola Extremum Seeking (LJES) method. The convergence of the LJES for dynamic conditions is established with the ability to handle sensor noise by quantizing the output. For the given example (Section IV-D and Table I) it can reduce network traffic by 99.14% which is a significant improvement compared to previous work [27]. Furthermore, it can directly evaluate constraints instead of using indirect penalty and barrier functions [42] [43]. The effective approach of the new method is very similar to event based sampling [38] [44] [45] [46] where periodic transmission of samples over the communication network is avoided by transmitting the sample only when a predefined condition is met. However, there is no reconstruction of the event based sampled signal at the controller.

C. Outline of this Paper

In the preliminaries (Section II) the background is discussed. The proposed algorithm, related formal proofs and simulation results are presented in Section III. In Section III-B convergence is first proven for a static objective function. It is then extended to a dynamic system in Section III-C. Some aspects of the algorithm are simulated in Section IV. In Section V, the proposed algorithm is implemented on an embedded water pump control system and the relevant implementation issues are discussed.

II. PRELIMINARIES

A. Notations

The set of real numbers and positive integers (excluding zero) are represented by $\mathbb{R}$ and $\mathbb{N}^+$ respectively. The Euclidean norm of a variable $z \in \mathbb{R}^n$ ($n \in \mathbb{N}^+$) is given by $\|z\|$. If z were a scalar $|z|$ would denote the absolute value. The floor of a floating point value z is represented by $\lfloor z \rfloor$ and is defined as the largest integer that is less than or equal to z. The letter i in square brackets is used for periodic (clocked) discrete time samples where $i = \lfloor t/T_S \rfloor$ and T_S is the sampling time.

The presence of a second argument for a discrete signal ($y[i,\tau]$) indicates a signal with a sampling offset, τ ($0 < \tau < T_S$) with respect to i. Therefore, the discrete time $y[i,\tau] = y(iT_S + \tau)$ in the continuous domain.

The Hessian operator of a matrix is denoted by $\mathcal{H}[\cdot]$. The boundary, interior and relative complement of a set $\mathcal{V}$ are denoted by $\partial\mathcal{V}$, $\mathcal{V}^\circ$ and $\backslash\mathcal{V}$ respectively. The symbol $\subset$ is used to indicate the subset of a set $\mathcal{V}$ including the boundary if applicable as per the definition of $\mathcal{V}$.

The letters $\mathcal{K}$, $\mathcal{K}_\infty$ and $\mathcal{KL}$ represent the class of functions as defined in [47] (p. 144) and are defined as follows:

Definition 1. *A class $\mathcal{K}$ function is a function $\alpha(s) : \mathbb{R} > 0 \rightarrow \mathbb{R} > 0$ that is continuous, strictly increasing and $\alpha(0) = 0$. The collection of class $\mathcal{K}$ functions where $s \in [0,\infty)$ is denoted by $\mathcal{K}_\infty$.*

Definition 2. *A class $\mathcal{KL}$ function is a function $\beta(s,t) : \mathbb{R}^2 > 0 \rightarrow \mathbb{R} > 0$ such that for fixed t, the mapping $\beta(s,t) \in \mathcal{K}$ with respect to s and for fixed s the mapping $\beta(s,t)$ is decreasing with respect to t and $\beta(s,t) \rightarrow 0$ when $t \rightarrow \infty$.*

B. Control System Model

The control system considered consists of a feedback loop (Fig. 1) implemented over a communication network. A plant with continuous dynamics with an output that needs to be optimized is considered. The system description is given by,

$$\left. \begin{array}{rcl} \dot{x} & = & f(x,u) \quad \text{with } x(0) = x_0 \\ y & = & g(x) \end{array} \right\} \quad (1)$$

where $x \in \mathbb{R}^{n_x}$ is the plant state, $u \in \mathbb{R}^{n_u}$ is the control input and $y \in \mathbb{R}$ the output ($n_x, n_u \in \mathbb{N}^+$). f is Lipschitz with respect to its arguments. The bounded convex set $\mathcal{U}$ is the set of feasible inputs u ($u \in \mathcal{U} \subset \mathbb{R}^{n_u}$), due to the satisfaction of constraints that may contain a combination of equality $\Psi(u) = 0$ ($\Psi(u) \in \mathbb{R}^{n_h}$) and inequality $\Phi(u) \geq 0$ ($\Phi(u) \in \mathbb{R}^{n_s}$) constraints. The strictly convex function $Q : \mathcal{U} \rightarrow \mathbb{R}$ with Lipschitz constant L_Q, is the objective function.

Assumption 1. *For any $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$, there exists a unique continuous mapping $x^* : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ (i.e., $x = x^*(u)$) such that $f(x^*(u), u) = 0$.*

Assumption 2. *The equilibrium point $f(x^*(u), u) = 0$ is globally asymptotically stable, uniformly in $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$.*

Remark 1. *In other words from Assumptions 1 and 2 for a constant u and given $x^*(\cdot)$, the system dynamics (1) converge to a unique equilibrium point $(x^*(u), u)$ determined by $u \in \mathcal{U}$. Assumptions 1 and 2 are standard assumptions in ESC [2] [3].*

Remark 2. *The output $y(t)$ consists of a transient error $e(t)$ (due to the dynamics of the system (1)) such that*

$$y(t) = Q(u) + e(t) \quad (2)$$

where there exists $\beta \in \mathcal{KL}$ such that $\|e(t)\| < \beta(\|e(t_0)\|, t - t_0)$ ($e(t_0)$ is the bounded initial transient error at the initial time instant t_0 such that $\|e(t_0)\| \leq \Delta_e$ where Δ_e is the upper bound). At steady state $y(t)$ is constant and $y(t) = Q(u)$. Further, $f(x,u)$ is such that there exists a unique steady state map $q : \mathcal{U} \rightarrow \mathbb{R}^{n_x}$ such that $Q(u) = g(q(u))$ for constant u.

Remark 3. *For the domain $\mathcal{U}$ the mappings $q(\cdot)$ and $x^*(\cdot)$ are equivalent resulting in $Q(u) = g(q(u)) = g(x^*(u))$.*

Remark 4. *From Assumptions 1 and 2 along with Remark 2 it becomes apparent that the LJES requires fast stable dynamics.*

Assumption 3. *The objective function $Q(u)$ is assumed to be strictly convex and without loss of generality, a minimizing problem with $v \in \mathcal{U}$ being the isolated minimum of $Q(u)$.*

This paper is limited to convex optimization for brevity. The use of metaheuristics and constructive methods will be discussed in a subsequent paper.

The control input is perturbed (dithered) by a discrete small random vector $a[i] = \begin{bmatrix} a_1 & a_2 & \ldots & a_{n_u} \end{bmatrix}^T$ (where $\|a[i]\| \leq a_m$) at an interval T_S generating the discrete dither signal

$$u[i] = u_k + a[i] \qquad (3)$$

where u_k is the *current estimate* of the optimal input. All elements a_j of $a[i]$ are uniformly distributed, symmetric and zero mean (i.e., $a_j \in [-a_{j,\max}, a_{j,\max}]$). The integer $k \in \mathbb{Z}^+$ corresponds to the number of *updates* of the current estimate of the optimal input that have taken place. The closed loop is formed by measuring the output y at an interval of T_S with a constant offset τ ($\tau \in (0, T_S)$ and allows the transients due to perturbation to settle down as shown in Fig. 2) using a sensor located at a considerable distance from the plant and controller (such as in the motivating example of Section I-A and implementation of Section V). This results in a discrete signal $y[i, \tau]$ which is then subject to detection rule $\mathcal{C} : y[i, \tau] < y_k$ for every i where y_k is the optimum output from measurements up to i. The update of u_k is determined by the satisfaction of the update criteria $\mathcal{C}$ or a reset $\mathcal{R}$ (Algorithm 1 line 18) such that $u_{k+1} = u[i]$ to avoid a potential deadlock elaborated in Corollary 1.
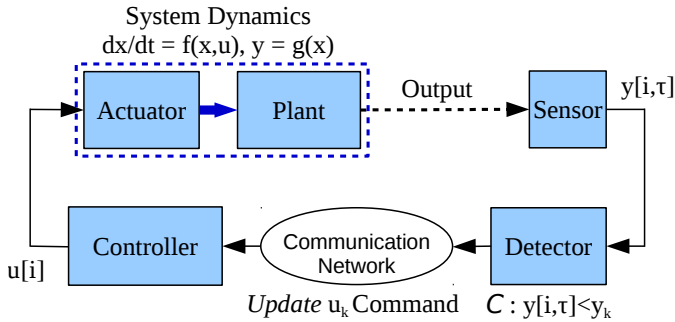

Fig. 1. Extremum Seeking Control using the LJES Algorithm
The sensor that measures the output is physically far from the plant (as in Section I-A and [31]) which justifies the need for a communication network for the feedback path.

Remark 5. *The parameter τ is selected such that the transients of (1) that occur at steady state due to the change from $u[i-1]$ to $u[i]$ settle down resulting in $y[i, \tau] \approx g(x^*(u[i])) = Q(u[i])$ until the next sampling instant.*

The selection of τ is based upon the simulated or empirical analysis of the transient behavior of (1). The most suitable method is to perturb u_k by an amount close to the maximum value of $a[i]$ and observe the settling time. The constant sampling offset τ is illustrated in Fig. 2 where a ZOH maintains the control input at $u(t) = u[i]$ for $t \in [iT_S, (i+1)T_S)$ and
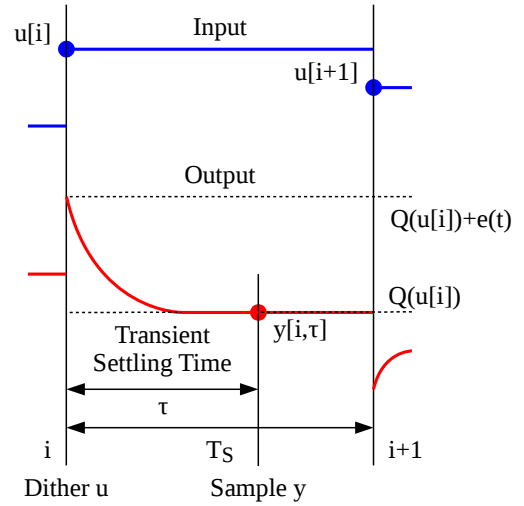

Fig. 2. Sampling Offset of the Output Measurement
The input u is dithered at time $i = \lfloor t/T_S \rfloor$. The output transients are allowed to settle during time τ. The output is subsequently sampled at time $i + \tau$ and used for the detection stage.

the output is sampled after allowing the transients to settle for time τ such that $y[i, \tau] = y(iT_S + \tau)$.

Assumption 4. *The delay and delay variation due to the communication network lag is assumed to be negligibly small compared to the difference $T_S - \tau$.*

Assumption 5. *All sensors, the controller and the plant that comprise the control loop of the networked control system (NCS) have perfectly synchronized clocks for periodic sampling and reconstruction.*

The implications of Assumptions 4 and 5 are discussed in detail in Section III-G along with the other main issue of packet drops that occurs when the LJES is implemented in a NCS.

III. MAIN RESULTS

The steps of the LJES algorithm are detailed in Algorithm 1. It is a special case of the LJ algorithm [41] where instead of generating p random numbers per iteration as in [41], a single random dither vector $a[i]$ is obtained at discrete time i. The loop is not terminated because the algorithm has to maintain the plant at the minimum by tracking any changes in the optimum value. The LJES is suitable for real time use compared to recent developments in the LJ algorithm [48] [49].

The reset $\mathcal{R}$ of line 18 is a clocked reset input (Remark 10) which is needed to prevent the algorithm from getting deadlocked (Corollary 1) in a particular position due to an error caused by a transient (Remark 9), input disturbance (Remark 14) or packet drop (Section III-G3). The value $y_k = Q(u_k)$ is the optimum value according to measurements up to i, while u_k is the estimate of the optimum input. The satisfaction of $\mathcal{C} : y[i, \tau] < y_k$ or reset results in a change of both of these values.

A. Algorithm Selection Rationale and Operation

1) Sporadic Packet Transmission: In previous ESC implementations that involved optimization algorithms such as the

```
 1  At $i = 0$ and $k = 0$, initialize $u[0] = u_0 \in \mathcal{U}$
 2  Input $u[0]$
 3  Measure $y[0, \tau]$
 4  Update $y_0 = y[0, \tau]$
 5  while do
 6  |    Generate $a[i]$
 7  |    Check $u_k + a[i] \in \mathcal{U}$
 8  |    if $u_k + a[i] \notin \mathcal{U}$ then
 9  |    |    // $u_k$ has been perturbed
    |    |       outside the boundary of $\mathcal{U}$
10  |    |    Find the boundary point $u_{BP}[i]$
11  |    |    // where $a[i]$ and $\partial\mathcal{U}$ intersect
12  |    |    $u[i] = u_{BP}[i]$
13  |    else
14  |    |    $u[i] = u_k + a[i]$
15  |    end
16  |    Input $u[i]$
17  |    Measure $y[i, \tau]$
18  |    if $(y[i, \tau] < y_k \vee (reset))$ then
19  |    |    // A better minimum has been
    |    |       found or reset $\mathcal{R}$ has
    |    |       occurred
20  |    |    Update $y_{k+1} = y[i, \tau]$
21  |    |    Update $u_{k+1} = u[i]$
22  |    |    Increment $k := k + 1$
23  |    end
24  |    Increment $i := i + 1$
25  end
```

Algorithm 1: LJES Algorithm

gradient descent [7], Shubert's glorithm [8], DIRECT [9] and state optimization [10] [11] [12], the control input $u[i]$ has to be computed from $y[i] = Q(u[i])$ (and previous values) for every i. Input perturbed ESC of [2] [3] requires the output to be correlated with the dither signal.

In the LJES, the dither input $u[i]$ depends only on locally generated $a[i]$ and stored u_k. Similarly the condition $\mathcal{C} : y[i, \tau] < y_k$ depends only on the plant output $y[i, \tau]$ and previously stored y_k. Hence, the dither and checking for condition $\mathcal{C}$ (detection) can occur independently until a *sporadic* update command for $u_{k+1} = u[i]$ is issued by the detector to the controller when either $\mathcal{C}$ is satisfied or a reset occurs.

2) Direct Evaluation of Constraints: Direct evaluation of constraints is included in the LJES algorithm for inequality (Section III-D1) and equality constraints (Section III-D2). When compared to other algorithms: [7] [10] [11] [12] have no constraints and [8] [9] have bounded $u[i]$. For extremum seeking systems that use parametric estimation [42] (instead of input perturbation), constraints are indirectly handled by using barrier functions for equality constraints and external penalty functions for inequality constraints. In [43] the same technique is used for continuous time systems with input perturbations.

3) Additional Merits: LJES is relatively simple to implement an LJES controller on an embedded system. The number of inputs n_u can be scaled as required. In addition, $a_{j,\max}$ can be varied according to a metaheuristic.

B. Static Convergence

This section establishes the convergence of the LJES to the isolated minimum v of a time invariant strictly convex problem. Theorem 1 provides the proof for static conditions which are extended to dynamic conditions in Section III-C.

Theorem 1. *Let $Q : \mathcal{U} \to \mathbb{R}$ be a strictly convex function (where $\mathcal{U} \subset \mathbb{R}^{n_u}$ is a convex set) that satisfies Assumption 3. For an arbitrary point $u_k \in \mathcal{U}$ from which a new point $u[i] \in \mathcal{U}$ is randomly perturbed such that $u[i] = u_k + a[i]$ at time i where $a[i]$ is a bounded random vector $(0 < \|a[i]\| \leq a_m)$ with every element a_j being of zero mean and independent (in both j and time i) resulting in $\|u[i] - u_k\| \leq a_m$, if an update $u_{k+1} = u_k + a[i]$ is made only when $Q(u_k + a[i]) < Q(u_k)$, the resulting u_k will converge to a neighborhood of v upper bounded by a_m when $k \to \infty$.*

Proof: Let convex set U_k be given by

$$U_k = \{U_k \subset \mathcal{U} | \forall u \in \partial U_k \setminus \partial\mathcal{U}, Q(u) = Q(u_k) \text{ and }$$
$$\forall u \in U_k^\circ, Q(u) < Q(u_k) \text{ where } u_k \in \partial U_k\}.$$

Since $Q(u)$ is strictly convex on U_k and for $u[i] = u_k + a[i] \in U_k^\circ$, $Q(u_k + a[i]) < Q(u_k)$. Hence, for the update $u_{k+1} = u_k + a[i]$,

$$U_{k+1} \subset U_k. \tag{4}$$

Thus, from Cantor's Intersection Theorem, the infinite intersection, $\cap U_k \neq \{\emptyset\}$ and there exists a point (i.e, the minimum v) such that $v \in U_k$ for all k. Since for all k, $u_k \neq u_{k+1}$ and $0 < \|u_{k+1} - u_k\| \leq a_m$, $u_k \in \partial U_k$ converges to within a small neighborhood of v when $k \to \infty$ such that $0 < \|u_k - v\| \leq \bar{v}_k < a_m$ where $\bar{v}_k = \max(\|v - u\|, \partial U_k \setminus \partial\mathcal{U})$ (the distance from the v to the furthest boundary point of $U_k \setminus \partial\mathcal{U}$). ∎

Remark 6. *Though the upper bound a_m is a loose bound, it is not possible to prove $\lim_{k \to \infty} u_k = v$ since it requires a_m to decrease with k. This cannot occur since all a_j are independent and uniformly distributed.*

Theorem 1 proves convergence of the LJES in real time compared to the proof for the LJ algorithm *per se* of [50]. In [50], the proof is based upon a batch generation of random $a[i]$ and requires a probabilistic selection of a suitable point and a decreasing perturbation (i.e., $\|a[i + 1]\| < \|a[i]\|$). The LJ algorithm [41] was originally proposed with empirical performance results in place of a formal proof.

Remark 7. *For an even distribution the vector elements a_j of $a[i]$ are selected from a uniform distribution such that $-a_{j,\max} \leq a_j \leq a_{j,\max}$. Near the boundary of $\mathcal{U}$, a_j will no longer be uniform due to scaling if $u[i]$ exceeds the boundary (Section III-D1).*

Remark 8. *In the case of finding a maximum Theorem 1 can be trivially proven for concave $-Q(u)$ with the rule $\mathcal{C} : Q(u_k + a[i]) > Q(u_k)$ ($\mathcal{C} : y[i, \tau] > y_k$ when measured).*

C. Dynamic Convergence

In this section, the proof of static convergence is extended to dynamic (transient) conditions. The transients that affect the system will be the initial transient of the dynamical system

and after reaching steady state, the perturbation transients that occur for input $u[i]$. From Assumptions 1 and 2, the stability of the system is guaranteed for an arbitrary bounded discrete sequence $u[i] \in \mathcal{U}$ with sampling time T_S and $\|u[i+1] - u[i]\| \leq a_m$ (where $\|a[i]\| < a_m$) due to the transient error satisfying (2). Thus, for convergence to the small neighborhood of the optimum point v under dynamic conditions it is necessary for the sequence to be updated according to the criterion $\mathcal{C} : Q(u_k + a[i]) < Q(u_k)$ of Theorem 1. Due to dynamics, this becomes $\mathcal{C} : y[i, \tau] < y_k$. This can be further generalized to

$$\mathcal{C} : Q(u_k + a[i]) + \epsilon[i, \tau] > Q(u_k) + \epsilon[i_k, \tau] \quad (5)$$

where $\epsilon[i, \tau] = e[i, \tau] + n[i, \tau]$ is the generic error during measurement due to the transient error $e[i]$ from (2) and sensor noise $n[i]$ from (Section III-E). The same error during the last update is given by $\epsilon[i_k, \tau]$. Let i_s be the time that steady state is reached such that for all $i \geq i_s$, $y[i, \tau] \approx Q(u_k + a[i])$ (from Remark 5). During the initial transient when $i < i_s$, the errors $\epsilon[i, \tau]$ and $\epsilon[i_k, \tau]$ cannot be neglected and could lead to the potential deadlock detailed in Corollary 1.

Corollary 1. *A sufficiently large negative error between $Q(u[i])$ and $y[i]$, $\epsilon[i]$ can cause the LJES to deadlock by non-satisfaction of the condition $\mathcal{C} : y[i, \tau] < y_k$ if $\forall \|a\| \leq a_m$, $\forall i > i_k > 0$, $\forall \epsilon[i, \tau]$, $Q(u_k + a[i]) + \epsilon[i, \tau] > Q(u_k) + \epsilon[i_k, \tau]$ where i_k is the time when the last update took place.*

Remark 9. *During the initial transient when $y(t)$ is subject to a bounded transient error $e(t)$ where $\|e(t)\| < \beta(\|e(t_0)\|, t - t_0)$ (from Remark 2), the the deadlock of Corollary 1 can potentially occur.*

Remark 10. *Should the potential deadlock of Remark 9 occur, it can be overcome by periodically resetting the stored value y_k using a clock input pulse $\mathcal{R}$ with duration T_S and period $T_R \gg T_S$ (Algorithm 1 line 18). Once, steady state is reached, the reset of $u_{k+1} = u[i]$ will allow the system to settle to $y_{k+1} \approx Q(u_{k+1})$.*

Alternatively, to avoid the deadlock of Corollary 1 the initial value of τ can be made very large to allow the initial transient to settle and then reduced when steady state is reached. However, the reset of Remark 10 is preferred because it will also rectify the deadlock of Corollary 1 should it occur in other situations besides that of Remark 9 such as an input disturbance (Remark 14), sensor noise (Remark 13), synchronization errors (Section III-G2) and packet drops (Section III-G3).

D. Constraints

1) Inequality Constraints: The LJES can incorporate any smooth inequality constraint as an implicit function such that $\Phi(u) \geq 0$ if the constraint is satisfied. Though the LJES can simply ignore a violation of an inequality constraint and retry, it can become inefficient under circumstances illustrated by the example of Section IV-A, where there is a high likelihood of the perturbation overshooting the boundary of $\mathcal{U}$. This can be avoided by scaling $a[i]$ by a parameter $\lambda \in (0, 1)$ such that from (3), if $u[i] = u_k + a[i] \notin \mathcal{U}$, a point on the boundary given by $u_{BP} = u_k + \lambda a[i] \in \partial \mathcal{U}$ is obtained. For convex $\mathcal{U}$, λ is unique since there will only be a single point of intersection

of $a[i]$ and $\partial \mathcal{U}^1$. The value of λ can be obained by minimizing the function $\sigma(\lambda)$ given by,

$$|\sigma(\lambda)| = \min_{\lambda} |\Phi(u_k + \lambda a[i])| \approx 0. \quad (6)$$

with the constraint $\Phi(u_k + \lambda a[i]) \geq 0$ provided that $\Phi(u)$ is nonlinear (Algorithm 2). Finding λ is trivial for linear inequality constraints.

```
1  if (Φ(u_k + a[i]) < 0) then
2  |    // u_k + a[i] ∉ U therefore find
   |       boundary point
3  |    while (Φ(u_k + λa[i]) > 0) do
4  |    |   Minimize |(Φ(u_k + λa[i]))| w.r.t. λ
5  |    |   u[i] = u_k + λa[i]
6  |    end
7  else
8  |    u[i] = u_k + a[i]
9  end
```

Algorithm 2: Boundary Point Algorithm for Nonlinear Inequality Constraints

2) Equality Constraints: For the LJES the main distinction for an equality constraint is whether it is parameterizable or not. A parameterizable equality constraint which can be linear (affine), a non-linear explicit function or a parameterizable manifold can be dithered trivially. For example the objective function $Q(u) = u_1^2 + u_2^2 + u_3^2$ when subjected to the affine constraints $u_1 + u_2 + u_3 = 1$ and $u_1 - 3u_2 - u_3 = 3$ can be parametrized as $u_1 = u_2 + 2$ and $u_3 = -2u_2 - 1$. Similarly the nonlinear constraint $u_1 = 2u_2^2 + u_3^2$ is an explicit function where u_2 and u_3 can be dithered. The manifold constraint $u_1^2 + 2u_2^2 = 1$ can be conveniently parametrized to $u_1 = \sin(\theta)$ and $u_2 = \cos(\theta)/\sqrt{2}$ and dithered by θ.

If parametrization is inconvenient, the constraint can be computed via an efficient Taylor series approximation of the constraint function provided that Ψ_j is sufficiently smooth. The value of small perturbation $b \in \mathbb{R}^{n_u}$, has to be sufficiently small to ensure the local Euclidean property ([52] p. 1) for this approximation to be valid. In [53], the Taylor series approximation is used for optimization using gradient descent methods for Stiefel and Grassmann manifolds.

The approximation is given by,

$$\begin{aligned} \Psi_j(u_k + b) &= \Psi_j(u_k) + b^T \nabla \Psi_j(u_k) \\ &\quad + \frac{1}{2} b^T \mathcal{H}[\Psi_j(u_k)]b + \Delta \end{aligned} \quad (7)$$

where $u_k \in \mathbb{F}$, $\mathcal{H}[\cdot]$ is the Hessian operator and Δ is the approximation error due to higher order terms. By taking,

$$\sigma(b) = b^T \nabla \Psi_j(u_k) + \frac{1}{2} b^T \mathcal{H}[\Psi_j(u_k)]b, \quad (8)$$

$$\begin{aligned} \Psi_j(u_k + b) &= \Psi_j(u_k) + \sigma(b) + \Delta, \\ |\Psi_j(u_k + b)| &\leq |\Psi_j(u_k)| + |\sigma(b)| + |\Delta_{\max}| \end{aligned} \quad (9)$$

where $\Delta_{\max}$ is the upper bound of the approximation error. Since $u_k, (u_k + b) \in \{u : \Psi_j(u) = 0\}$, the best approximation for the next point on the constraint will be when $\Psi_j(u_k +$

---

[1]From Theorem 1.4 of [51] p. 114-116.

$b) \to 0 \Rightarrow |\Psi_j(u_k + b)| \to 0$. Since $|\Delta_{\max}|$ may be unknown, the feasible method to achieve this is to consider (8) as a minimization problem similar to (6) such that

$$|\sigma(b)| = \min_b \left| b^T \nabla \Psi_j(u_k) + \frac{1}{2} b^T \mathcal{H}[\Psi_j(u_k)] b \right| \approx 0 \quad (10)$$

where $\|b\| \in (0, b_{\max}]$. Ideally the value of b from the solution of (10) should satisfy $\Psi_j(u_k + b) = \Psi_j(u[i]) = 0$ as a point on the manifold. However, this is not possible when b is computed with finite precision, in which case the manifold $\Psi(u) = 0$ has to be approximated.

Theorem 2. *Let $\Psi(u) = 0$ be an embedded differentiable manifold and u_p an iteratively obtained point very close to $\Psi(u) = 0$ such that $|\Psi(u_p)| \leq \varepsilon_T$ where ε_T is an arbitrarily small bound. A subsequent point $u_{p+1} = u_p + b$ where b is obtained from minimizing $\left| b^T \nabla \Psi(u_p) + \frac{1}{2} b^T \mathcal{H}[\Psi(u_p)] b \right|$ w.r.t. b will approximate the manifold for all p if the bound $|\Psi(u_{p+1})| \leq \varepsilon_T$ is satisfied.*

Proof: Let U_Ψ be a set with inner boundary, $\underline{U} = \{u \in \mathcal{U} | \Psi(u) + \varepsilon_T = 0\}$ and outer boundary, $\overline{U} = \{u \in \mathcal{U} | \Psi(u) - \varepsilon_T = 0\}$. Thus, for all u such that $|\Psi(u)| \leq \varepsilon_T \Rightarrow u \in U_\Psi$. From (9), the upper bound for the error of an iteration p is given by ε_p,

$$|\varepsilon_p| = |\varepsilon_{p-1}| + |\varepsilon_O| + |\Delta_{\max}| \quad (11)$$

where ε_O is the bound of error in convergence of the finite precision result b from the ideal solution b_I, where $\Psi(u_k + b_I) = 0$. Thus,

$$|\varepsilon_O| = |\Psi(u_k + b)| \quad (12)$$

Since the upper bound can grow with each iteration, a constant conservative bound $\varepsilon_T > 0$ has to be forced for each iteration such that $|\Psi(u_p + b)| \leq \varepsilon_T$ to ensure $u_p + b \in U_\Psi$ for all u_p. ∎

From the result of Theorem 2, it is possible to use Algorithm 3 to allow the LJES to dither the search space along an approximation of the manifold equality constraint.

---

input : $u_k \in U_\Psi$
output: $u[i] \in U_\Psi$
1 Minimize $\left| b^T \nabla \Psi(u_k) + \frac{1}{2} b^T \mathcal{H}[\Psi(u_k)] b \right|$ w.r.t. b
2 **if** $|\Psi(u_k + b)| \leq \varepsilon_T$ **then**
3   // If $u_k + b \in U_\Psi$ return $u_k + b$
4    $u[i] = u_k + b$
5 **else**
6   // If $u_k + b \notin U_\Psi$ return u_k
7    $u[i] = u_k$
8 **end**

Algorithm 3: Taylor Series Approximation for Manifold Equality Constraints

---

Remark 11. *Efficient minimization algorithms are necessary for the relevant steps of Algorithms 2 and 3 to allow the LJES to run in real time (time complexity of $\mathcal{O}(1)$) and implemented on an embedded system. This algorithm has to be implemented within the LJES controller. Since the solution for $\lambda \in (0, 1)$ of Algorithm 2 and non-trivial solution for b from quadratic*

(10) of Algorithm 3 are both unique, a simple algorithm like the steepest gradient descent can be used.

E. Sensor Noise

This section looks into when $y[i, \tau]$ is subject to an additive bounded zero mean measurement noise $n[i]$ where $\|n[i]\| < n_m$ (Fig. 3).

Remark 12. *The deadlock of Corollary 1 will occur if $\forall \|a[i]\| \leq a_m$, $\forall i > i_k > 0$, $\forall n[i, \tau]$ $Q(u_k + a[i]) + n[i, \tau] > Q(u_k) + n[i_k, \tau]$ where the time of the last update is i_k.*

Though this situation can be remedied by the periodic reset of Remark 10, the convergence time can be further augmented by the 0.5 probability of divergence of $u[i]$ that can occur if $n[i, \tau] > 0$ (which has a probability of 0.5) occurs during a periodic reset. Therefore, the alternative solution of quantization is used. For notational convenience, Theorem 3 shall be proven using $u[i]$ instead of the notation of Fig. 3 which shall be adopted in Section III-F.

Theorem 3. *For strictly convex function $Q : \mathcal{U} \to \mathbb{R}$ on convex set $\mathcal{U} \subset \mathbb{R}^{n_u}$ with Lipschitz constant L_Q, the result of Theorem 1 will continue to hold in the presence of bounded additive zero mean noise such that $Q(u[i]) + n[i, \tau]$ (where $|n[i, \tau]| < n_m$) when $Q(u)$ is quantized by a mid tread operator $\mathcal{Q}[\cdot]$ such that $\mathcal{Q}[Q(u)] = \Delta_q \left(\left\lfloor \frac{Q(u)}{\Delta_q} + \frac{1}{2} \right\rfloor \right)$ (where $\lfloor \cdot \rfloor$ is the floor operator and Δ_q is the quantization step size with $\Delta_q > n_m$) when a_m ($\|a[i]\| \leq a_m$) is sufficiently large such that $\frac{\Delta_q + n_m}{L_Q} \ll a_m$.*

Proof: Let $Q(u)$ be quantized into $p \in \mathbb{Z}$ levels such that each level is a convex set given by,

$$\begin{aligned} U_p &= \{U_p \subset \mathcal{U} | \forall u \in \partial U_p \setminus \partial \mathcal{U}, Q(u) = p\Delta_q \text{ and} \\ &\quad \forall u \in U_p^\circ, Q(u) < p\Delta_q \text{ where } p \in \mathbb{Z}\}. \end{aligned}$$

For the condition $\mathcal{C} : Q(u[i] + a[i]) < Q(u[i])$ to be satisfied, $u[i] \in U_p$ and $u[i] + a[i] \in U_{p+l}$ where $l \in \mathbb{Z}$. To cancel noise $n_m < \Delta_q$. For a given p with $l = 1$, there exists a region $Q(u) \in ((p-1)\Delta_q + n_m, p\Delta_q)$ for which $\mathcal{C}$ can be triggered due to additive noise resulting in $u[i] + a[i] \in U_{p-1}$. This results in the lower bound

$$\frac{\Delta_q + n_m}{L_Q} \leq a_m \quad (13)$$

since $0 \leq \nabla Q(u) \leq L_Q$, from by considering the gradient instead of L_Q, (13) a_m cannot be upper bounded. This results in the requirement of a_m to be chosen as $\frac{\Delta_q + n_m}{L_Q} \ll a_m$. ∎

Remark 13. *For any p, when within the region $Q(u) \in (p\Delta_q, p\Delta_q + n_m)$, the resulting error of Δ_q could lead to the deadlock of Corollary 1. The reset described in Remark 10 suffices to overcome the deadlock situation in this case as well.*

F. Input Disturbances

In this section the effect and mitigation of an additive disturbance $w[i]$ ($\|w[i]\| \leq w_m$) to the input $u[i]$ (Fig. 3) where Assumption 2 continues to hold is discussed. The control input to the plant is now given by $u = w[i] + u_C[i]$ where $u_C[i]$ is the output of the controller (instead of previous $u[i]$). Such a
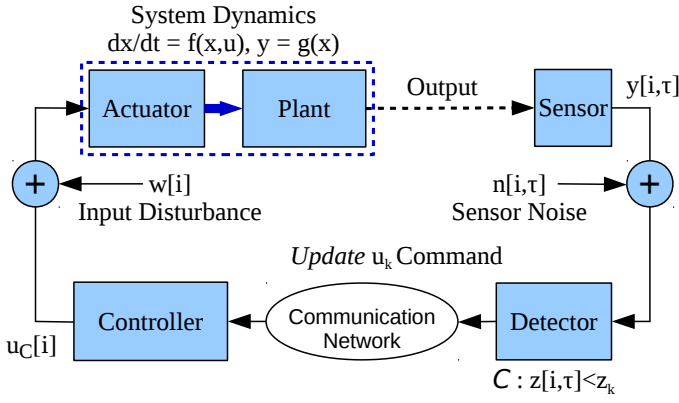
Fig. 3. The LJES Algorithm with Sensor Noise and Input Disturbances

disturbance will shift the position of the optimum point. For static conditions, this shift is upper bounded by,

$$
\begin{aligned}
Q(u_C[i] + w[i]) &\leq Q(u_C[i]) + L_Q\|w[i]\| \\
&\leq Q(u_C[i]) + L_Q w_m
\end{aligned}
\tag{14}
$$

for all $u_C[i] \in \mathcal{U}$ where L_Q is the Lipschitz constant of $Q(u)$. Practically such a disturbance is equivalent to a disturbance at the actuator.

Remark 14. *Under dynamic conditions, during the initial transient when the transient error $e[i]$ from (2) cannot be neglected, the resulting error at the output $Q(u_C[i] + w[i]) + e[i]$ can potentially result in the deadlock of Corollary 1. Thus, if $\forall \|a[i]\| \leq a_m$, $\forall i > i_k > 0$, $\forall e[i,\tau]$, $\forall w[i,\tau]$, $Q(u_k + a[i] + w[i,\tau]) + e[i,\tau] > Q(u_k + w[i_k,\tau]) + e[i_k,\tau]$ where i_k is the time when the last update took place, the same periodic reset of Remark 10 is sufficient to overcome this situation.*

G. Communication Network Issues

The NCS of this paper will further have the following issues due to the communication network.

1) Delay: It is apparent that if the upper bound of the delay of the communication network t_D is large (i.e., $t_D > T_S$), the update of output $y[i,\tau]$ occurs for a future value of the control input $u[i+p]$ where $p \leq \lfloor t_D/T_S \rfloor + 1 \in \mathbb{Z}^+$. In which case though the stability of the control system is guaranteed from Assumptions 1 and 2, there is no guarantee that the control input $u[i]$ converging to within a small neighborhood of the minimum point v since the update command is received incorrectly. The resulting error $y[i,\tau] - y[i+n,\tau]$ can potentially lead to the deadlock of Corollary 1. As long as Assumption 4 holds, this problem will not occur.

2) Synchronization: The error due to clock drift is a similar situation where the update of $y[i,\tau]$ can occur for $u[i \pm 1]$. Clock drift is the difference between the time period of two clocks, resulting in a relative gain or loss of time of one clock with respect to the other. In the case of the LJES, if the clocks of the controller and detector have significant drift, when the gained or lost time equals T_S, the output $y[i,\tau]$ momentarily becomes $y[i \pm 1, \tau]$ with respect to the controller. This too can result in non-convergence of the LJES due to the error $y[i,\tau] - y[i \pm 1, \tau]$ leading to the deadlock of Corollary 1. In order to avoid it is necessary to synchronize the two clocks

(Assumption 5) to remove the gained or lost time. This can be acheived by using an established algorithm like Network Time Protocol (NTP) [54]. NTP does this by generating a time offset by statistical estimation of the time difference between clocks and network latency via timestamps.

3) Packet Drops: During a packet drop the update $u_{k+1} = u_k + a[i]$ will fail to take place due to the packet containing the update command getting dropped. This will result in the error $Q(u_{k+1} + a[i]) - Q(u_k + a[i])$ which can lead to the deadlock of Corollary 1. Though acknowledging the command seems the intuitive solution, it itself can get dropped and the timeout used to remedy this may be infeasible for small sampling times. Thus, the three solutions listed below can be used

1) Multiple packet transmission, such as the retransmission of critical Generic Object Oriented Substation Event (GOOSE) packets in the electric substation automation protocol IEC61850 ([55] Clause 18.1.2.5).
2) IEC62439 Parallel Redundancy Protocol (PRP) [56] implementation where the controller and detector have two network cards that connect to two independant networks.
3) The periodic reset of Remark 10. The main drawback of this method is that since the reset time, $T_R \ll T_S$, the LJES will remain deadlocked for a significant proportion of time if packet drops are frequent.

IV. SIMULATION RESULTS

In this section the performance of the LJES algorithm is simulated for specific extremum seeking scenarios to demonstrate its ability to handle constraints. To the best of the authors knowledge, there is currently no ESC method that is technically equivalent to the proposed method in terms of having sporadic packet transmission between the dither and detection units of the implementation. The same applies to the methods used for handling constraints. Similarly, despite the numerous methods proposed for ESC outlined in Section I-A, no significant work exists on how to accurately benchmark and compare performance. Hence, the simulated examples and implementation (Section V) are included for illustrative (i.e., proof of concept) purposes only. For brevity, sensor and input disturbance noise are not considered for the simulations because it is inherent in the implementation of the LJES in Section V. All simulations are done using Scilab and MATLAB Simulink.

A. MISO Convex Inequality Constrained Optimization

In this example, the ability of the LJES to handle inequality constraints that result in a progressively reduced search space is demonstrated. The spherical objective function

$$
Q(u) = (u_1 - 100)^2 + u_2^2
$$

with the minimum at $(100, 0)$ is subjected to the convex inequality constraint obtained from the implicit curve of the trangle with vertices $\{(0,1), (105,0), (0,-1)\}$ given by

$$
u_1 \left(u_2 + \frac{u_1}{105} - 1\right)\left(u_2 - \frac{u_1}{105} + 1\right) + 0.1 \leq 0
$$

and transient dynamics ($x \in \mathbb{R}$),

$$
\left.
\begin{aligned}
\dot{x} &= Q(u) - q_0 x \\
y &= \frac{x}{q_0}
\end{aligned}
\right\}
\tag{15}
$$

with $q_0 = 10$. The remaining parameters are heuristically selected such that $a_{j,\max} = 1$ for all elements of $a[i]$, sampling time $T_S = 1s$, settling time $\tau = 0.5$ and reset time $T_R = 100s$. The simulation is run for $2000s$ which is sufficient time for convergence. The results of Fig. 4 and 5b show the convergence of the algorithm at approximately 1200s. A total of 707 packets were generated which corresponds to a reduction in network traffic of 64.65%. The packets generated of Fig. 5c show the reduced transmission of packets after convergence. In the logarithmic plot of $y[i]$ of Fig. 6, it can be observed that the perturbations result in the measured output being orders of magnitude greater compared to y_k after convergence resulting in few updates of y_k.
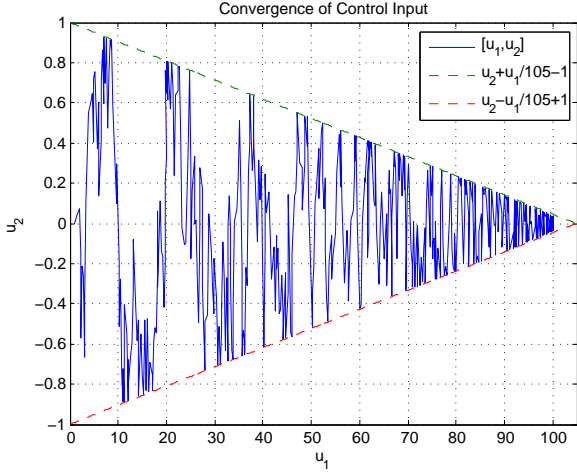
Fig. 4. Input Convergence for the Example of Section IV-A
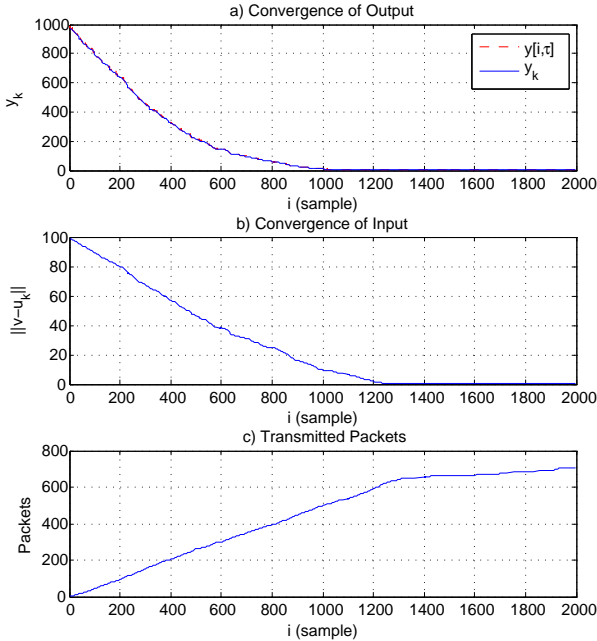
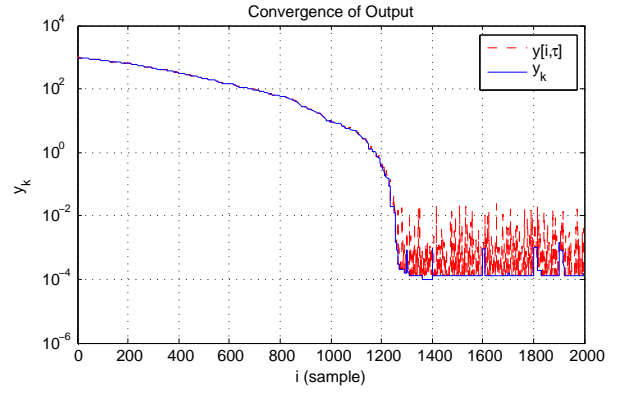Fig. 5. Output, Input Convergence and Generated Traffic for the Example of Section IV-A

Fig. 6. Log Plot of Output Convergence for the Example of Section IV-A

B. MISO Convex Manifold Equality Constrained Optimization

In this example, the spherical objective function

$$Q(u) = u_1^2 + (u_2 - 1)^2$$

is equality constrained by the implicit curve manifold

$$u_2 \left(u_1^2 + u_2^2 - 1 \right) + 0.01 = 0$$

where $u_1 \in [-1, 1]$ and $u_2 \in [0.2, 1]$, resulting in a convex search space with the minimum at $(0.0.995)$. The same transient dynamics of (15) with $q_0 = 10$. The threshold, $\varepsilon_T = 10^{-6}$, The other parameters are given by $a_{j,\max} = 0.1$, $T_S = 1s$, $\tau = 0.5$ and $T_R = 100s$ with the simulation run for $200s$. When the algorithm starts from $(0.9456, 0.2589)$ (Fig. 7) it converges to the minimum in around $80s$ (Fig. 8b) generating 45 packets. This corresponds to a traffic reduction of 77.5%. Similar to the example of Section IV-A, in Fig. 9, perturbations make the measured output orders of magnitude greater than y_k after convergence. Thus, there are few updates of y_k.
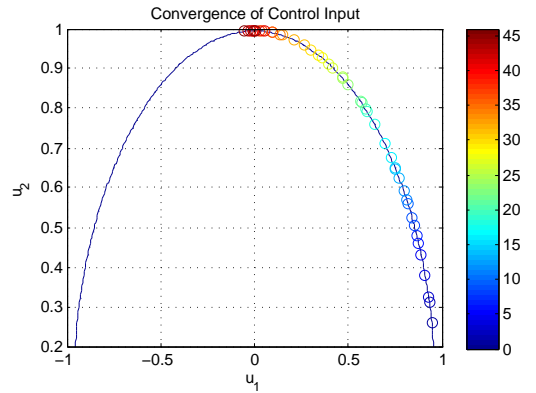

Fig. 7. Input Convergence for the Example of Section IV-B

C. MISO Non-Quadratic Objective Function

In this example, the non-quadratic and non-convex Mc-Cormick function given by,

$$Q(u) = \sin(u_1 + u_2) + (u_1 - u_2)^2 - 1.5u_1 + 2.5u_2 + 1$$

is constrained by inequality constraint

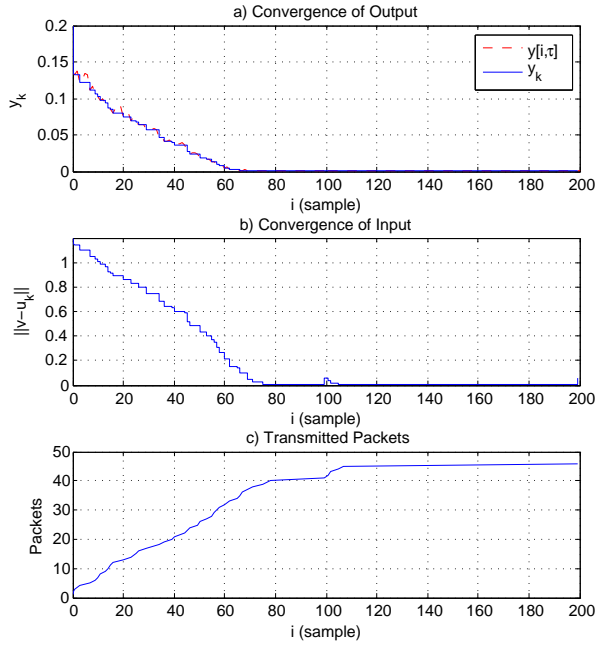$$(u_1 + 1.5)(u_2 + 3)(5.5u_2 + 7u_1 - 11.5) + 5 \leq 0$$

Fig. 8. Output, Input Convergence and Generated Traffic for the Example of Section IV-B
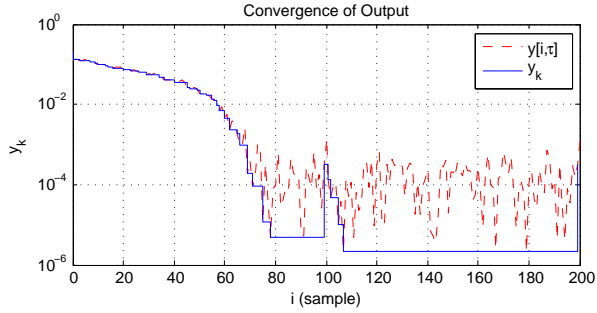

Fig. 9. Log Plot of Output Convergence for the Example of Section IV-B

Fig. 10. Input Convergence for the Example of Section IV-C

Fig. 11. Output, Input Convergence and Generated Traffic for the Example of Section IV-C

to become convex. It is further used to illustrate the practical need for the reset of Remark 10 due to the initial transient caused by a large negative intial value, i.e. $x(0) = -100$. The remaining parameters are given by $a_{j,\max} = 0.1$, $T_S = 1s$, $\tau = 0.5$ and $T_R = 100s$ and the simulation run for $1000s$. The LJES converges to the local optimum $(-0.54719, -1.54719)$, as constrained (Fig. 10) within around 420s and generates 196 packets (80.4% reduction) during the process. During the initial $100s$ of the, the LJES remains deadlocked (Fig. 11a) due to the large negative initial transient and starts converging only after the reset of Remark 10 occurs at $100s$.

D. Performance Comparison

The performance of the LJES algorithm is compared with sinusoidally perturbed (SP) ESC [2] [3] [4]: the other completely real time implementation of ESC. The LJES is compared with SP alone and SP with the feedback encoded (as in [27]) by Deadband Error Modulation (DEM) [40] and Memory Based Event Triggering (MBET) [38] in terms of both convergence time and traffic reduction. The previously proposed method of Event Triggered Adaptive Differential
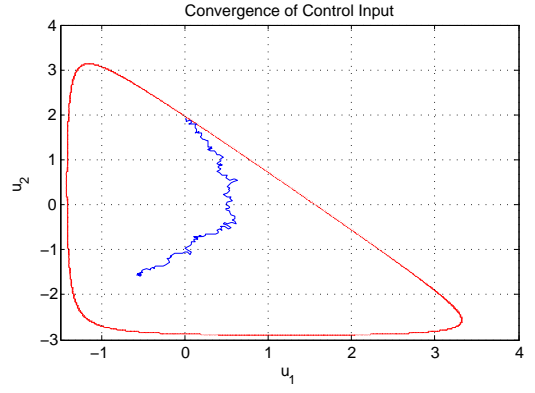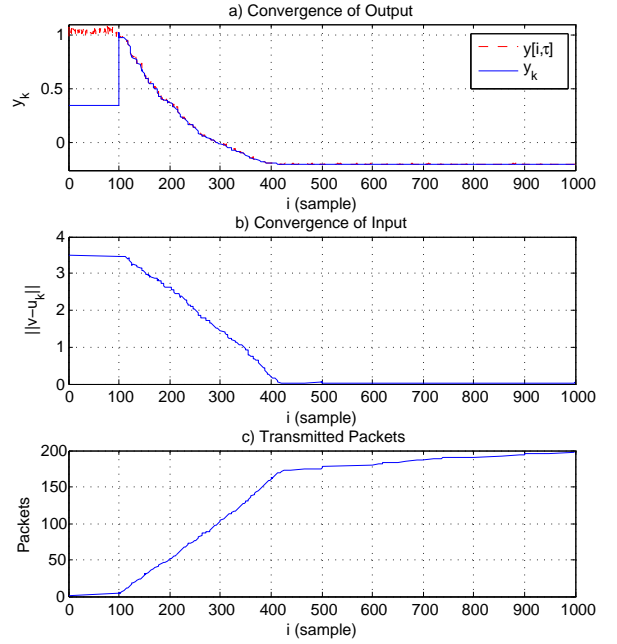
Modulation (ETADM) [57]) which is used in [27] is excluded since it is significantly outperformed by both DEM and MBET. The output and system dynamics are respectively given by

$$y = (x - 3)^2$$
$$\dot{x} = 100(u - x)$$

where u and x are scalars. This results in the steady state objective function of $Q(u) = (u - 3)^2$. A rapid transient is selected for fast convergence of the SP method within the simulation time of 1000s. The convergence time is taken as the output reaching within ± 0.1 units of zero (the minimum value). The amplitude of the sinusoidal perturbation of 10rad/s is taken as 0.1 and the integrator gain is taken as 0.9 for the SP and SP with feedback encoding. To be comparable the uniform dither is taken within the range of ± 0.1. The system is also subjected to a symmetric triangular input disturbance of 100s duration and 0.5 unit amplitude. The results of Table I show that LJES significantly outperforms SP and SP with encoded feedback in terms of both traffic reduction and convergence

time despite the LJES apparently having partial information compared to SP. This apparent paradox can be explained by the stability properties and slow learning rate of SP when compared to the LJES (see Appendix).

TABLE I. ESC PERFORMANCE COMPARISON

Method	Conv. Time (s)	No. Packets	Traffic Red. (%)
SP	190.53	100000	–
SP-DEM	190.54	3380	96.62
SP-MBET	199.33	3347	96.65
LJES	**4.42**	**860**	**99.14**

V. CONTROL SYSTEM IMPLEMENTATION

The proposed LJES is tested on a proof of concept pump controller that maintains the water flow rate at an optimum level. The system (Fig. 12 and Fig. 13) controls the output of the pump via Pulse Width Modulation (PWM). It is implemented on two Arduino equivalent EtherTen development boards. A magnetic hall effect flow sensor measures the flow rate using Pulse Frequency Modulation (PFM). An update is transmitted from the detector to the controller via the communication network using User Datagram Protocol (UDP). Two pumps are used one with a commutative motor and the other with a brushless one.

The objective function is given by $y = Q(u)$ where $y = g(x) \geq 0$ is the flow output (measurement of the sensor) and u is the PWM input. Both u and y are integers. For both pumps the output $0 \leq y \leq 38$ with the maximum output for a supply of 15V. The mapping $Q(u)$ varies significantly due to flow uncertainties. To prevent the system from entering the trivial equilibrium point $(u, y) = (0, 0)$, $1 \leq u < 254$. Due to the limited computational power of the embedded development board, all calculations are integer based. A maximizing objective function is used due to software coding efficiency. An integer dither with a range up to ± 32 is used along with $T_S = 2s$ and $\tau = 0.5s$. The PFM measurement requires a further 1s and the remaining 0.5s is the upper bound for the network delay. The experimental setup is run for 2000 such samples (4000s) with the reset of Remark 10, resulting in at least 20 packets being transmitted per experiment. The two boards are clocked using a common external signal to mimic two perfectly synchronized clocks.
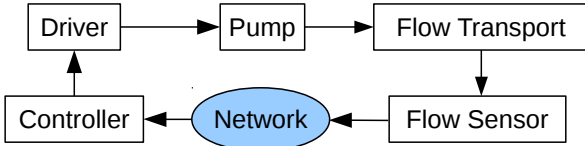

Fig. 12. Pump Control System Block Diagram

Performance is quantified in terms of the convergence time and number of generated packets for dither ranges of ± 1, ± 2, ± 4, ± 8, ± 16 and ± 32. The convergence time is taken as the time at which the output reaches 90% of the maximum (i.e., sensor reading of 34 or above). The results are given in Table II and clearly indicate that the packets generated during the 4000s of the experiment and traffic reduction performance are highly consistent. Also as expected, the convergence time decreases significantly with increased dither.

TABLE II. LJES HARDWARE PERFORMANCE RESULTS

Dither	Performance Result					
	Commutative Motor Pump			Brushless Motor Pump		
	Conv. (s)	Packets	Red. (%)	Conv. (s)	Packets	Red. (%)
± 1	–	50	97.50	–	47	97.65
± 2	1602	51	97.45	3560	53	97.35
± 4	130	51	97.45	802	48	97.60
± 8	62	52	97.4	110	52	97.4
± 16	34	45	97.75	40	45	97.75
± 32	16	42	97.9	16	42	97.9

VI. CONCLUSION

The two novelties of the LJES are sporadic packet transmission that occurs only when there is a change in the measured optimum output and direct evaluation of constraints. The LJES is simple to implement and scalable in terms of inputs. Hence, it can be readily implemented on an embedded controller (Fig. 13). Sporadic packet transmission makes it highly desirable for bandwidth constrained NCS compared to [7] [8] [9] [10] [11] [12] where the dither signal or state estimation requires a packet transmission of the measured plant output for each periodic iteration.

The drawbacks of the LJES include initial value dependency when used with non-convex objective functions, which can be overcome by adjusting the upper bound of the perturbation a_m using a metaheuristic. The LJES is also vulnerable to packet drops and timing offsets. In the case of packet drops this shortcoming can be overcome using multiple packet transmissions with a small increase in required bandwidth.

The development of the LJES results in numerous future directions. The main future direction is the development of suitable metaheuristics for global optimization of non-convex objective functions. There is also the need to benchmark ESC systems for standardized comparison purposes. In addition, the ability to improve convergence (from Remark 6) by detecting whether the current estimate u_k is within the bound $\|u_k - v\| \leq a_m$ of the optimum using output measurements and subsequently decreasing a_m is another interesting open problem.
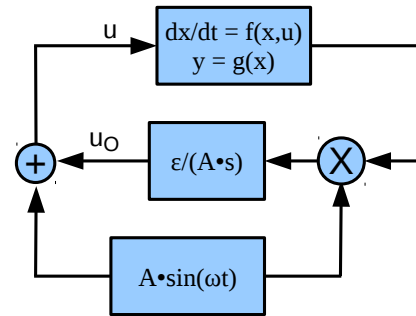
VII. ACKNOWLEDGEMENT

APPENDIX

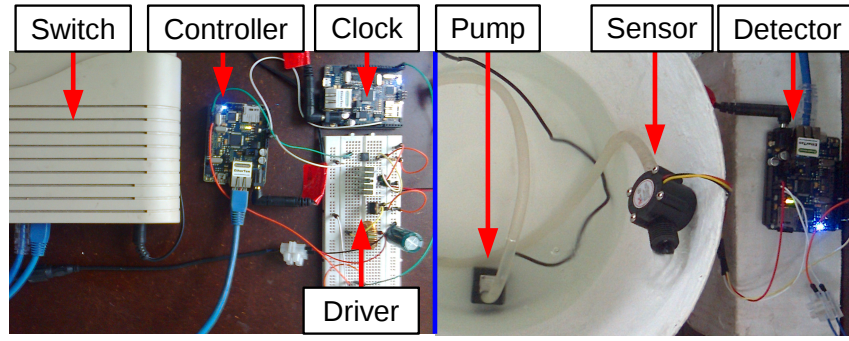Fig. 14. Generic Sinusoidal Perturbation

Fig. 13. Pump Control System Implementation

For brevity, consider the illustrated ESC scheme of Fig. 14 which sinusoidally perturbs the stable system dynamics of (1), i.e., Assumptions 1 and 2 with an output that converges to $Q(u)$ at steady state. Let the angular frequency of the sinusoidal perturbation be ω with $\epsilon = \delta\omega$ where δ is a small parameter. In the LJES where transients of x and update of u are independent and from Assumptions 1 and 2, stability can be guaranteed for arbitrary u. However, from [3] [4] (where Assumptions 1 and 2 also apply), the dynamics of SP ESC form a singularly perturbed system with perturbation parameter ω (from equation (15) of [3]) from which stability is established. The dynamics of the sinusoidal perturbation of the search space are given by

$$\begin{aligned} \dot{x} &= f(x, u_O + A\sin(\omega t)) \\ \dot{u}_O &= \epsilon g(x)\sin(\omega t) \end{aligned}$$

where A is the amplitude of the sinusoidal pertubation, ϵ is a small parameter and u_O is the estimate of the optimum. Stability for the optimum v is established when the system is "frozen" at the slow time scale $s = \omega t$. From equations (10) and (11) of [4], the averaged system is given by,

$$\begin{aligned} \frac{d}{ds}u_{av} &= \delta A \int_0^{2\pi} Q\left(u_{av} + A\sin(s)\right)\sin(s)ds \quad (16) \\ &\approx \delta A \int_0^{2\pi} \Big[Q(u_{av})\sin(s) + \\ &\quad \frac{A}{2}\nabla Q(u_{av})\left(1 - \cos(s)\right) + \mathcal{O}(A^2)\Big]ds \\ &\approx \frac{\delta A^2}{2}\left[\nabla Q(u_{av}) + \mathcal{O}(A^2)\right] \quad (17) \end{aligned}$$

which is an estimate of the gradient of the system. In order to obtain this estimate, SP requires a time $T \gg T_S$ (where $T = 2\pi/\omega$) which is further scaled by small ϵ. For the LJES, an update can occur during a time instance i of duration T_S where only a change in the measured optimum output is detected. Thus, the LJES has a significantly faster learning rate than SP resulting in faster convergence.

REFERENCES

[1] C. S. Draper and Y. Li, *Principles of optimalizing control systems and an application to the internal combustion engine.* ASME Publications, 1951.

[2] M. Krstic and H. H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36 (4), pp. 595–601, 2000.

[3] Y. Tan, D. Nesic, and I. Mareels, "On non-local stability properties of extremum seeking control," *Automatica*, vol. 42 (6), pp. 889–903, 2006.

[4] Y. Tan, W. H. Moase, C. Manzie, D. Nesic, and I. Mareels, "Extremum seeking from 1922 to 2010," in *Proceedings of the 29th Chinese Control Conference*, 2010.

[5] C. Manzie and M. Krstic, "Extremum seeking with stochastic perturbations," *IEEE Transactions on Automatic Control*, vol. 54 (3), pp. 580–585, 2009.

[6] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37 (3), pp. 332–341, 1992.

[7] A. Teel and D. Popovic, "Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming," in *Proceedings of the American Control Conference*, vol. 3, 2001, pp. 2394–2399.

[8] D. Nesic, T. Nguyen, Y. Tan, and C. Manzie, "A non-gradient approach to global extremum seeking: An adaptation of the shubert algorithm," *Automatica*, vol. 49 (3), pp. 809–815, 2013.

[9] S. Z. Khong, D. Nesic, C. Manzie, and Y. Tan, "Multidimensional global extremum seeking via the direct optimisation algorithm," *Automatica*, vol. 49 (7), pp. 1970–1978, 2013.

[10] C. Zhang and R. Ordonez, "Numerical optimization-based extremum seeking control with application to abs design," *IEEE Transactions on Automatic Control*, vol. 52 (3), pp. 454–467, 2007.

[11] C. Hong and K. Li, "Swarm intelligence-based extremum seeking control," *Expert Systems with Applications*, vol. 38 (12), pp. 14 852–14 860, 2011.

[12] S. J. Yu, H. Chen, and L. Kong, "Particle swarm optimization-based extremum seeking control," in *Advanced Intelligent Computing Theories and Applications*. Springer, 2010, pp. 185–196.

[13] M. Ye and G. Hu, "A robust extremum seeking scheme for dynamic systems with uncertainties and disturbances," *Automatica*, vol. 66, pp. 172–178, 2016.

[14] J. Y. Choi, M. Krstic, K. B. Ariyur, and J. S. Lee, "Extremum seeking control for discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 47 (2), pp. 318–323, 2002.

[15] W. H. Moase, Y. Tan, D. Nesic, and C. Manzie, "Non-local stabilitiy of a mutli-variable extremum-seeking scheme," in *Australian Control Conference (AUCC), 2011*, 2011, pp. 38–43.

[16] M. Leblanc, "Sur l'electrication des chemins de fer au moyen de courants alternatifs de frequence elevee," Revue Generale de l'Electricite, 1922.

[17] V. V. Kazakevich, "Technique of automatic control of different processes to maximum or to minimum," USSR Patent 66 335, 1943.

[18] A. Banaszuk, K. B. Ariyur, M. Krstic, and C. A. Jacobsen, "An adaptive algorithm for control of thermoacoustic instability," *Automatica*, vol. 40 (11), pp. 1965–1972, 2004.

[19] G. Lara-Cisneros, R. Aguilar-López, and R. Femat, "On the dynamic optimization of methane production in anaerobic digestion via extremum-seeking control approach," *Computers & Chemical Engineering*, vol. 75, pp. 49–59, 2015.

[20] P. G. Scotson and P. E. Wellstead, "Self-tuning optimization of spark ignition automotive engines," *IEEE Communications Magazine*, vol. 10 (3), pp. 94–101, 1990.

[21] N. Bizon, "Global maximum power point tracking (gmppt) of photovoltaic array using the extremum seeking control (esc): A review and a new gmppt esc scheme," *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 524–539, 2016.

[22] A. Kebir, L. Woodward, and O. Akhrif, "Extremum-seeking control with adaptive excitation: application to a photovoltaic system," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2507–2517, 2018.

[23] B. Mu, Y. Li, J. M. House, and T. I. Salsbury, "Real-time optimization of a chilled water plant with parallel chillers based on extremum seeking control," *Applied Energy*, vol. 208, pp. 766–781, 2017.

[24] U. Ciri, M. Rotea, C. Santoni, and S. Leonardi, "Large-eddy simulations with extremum-seeking control for individual wind turbine power optimization," *Wind Energy*, vol. 20, no. 9, pp. 1617–1634, 2017.

[25] Z. Cao, R. Gondhalekar, E. Dassau, and F. J. Doyle, "Extremum seeking control for personalized zone adaptation in model predictive control for type 1 diabetes," *IEEE Transactions on Biomedical Engineering*, 2018.

[26] U. Premaratne, "Network traffic reduction strategies for packet switched networked control systems," Ph.D. dissertation, University of Melbourne, 2015.

[27] V. S. Wijesinghe and U. K. Premaratne, "Bandwidth reduction and convergence analysis of extremum seeking control with feedback encoding," *Frontiers in Mechanical Engineering*, vol. 2, p. 11, 2016.

[28] C. Li, Z. Qu, and M. A. Weitnauer, "Distributed extremum seeking and formation control for nonholonomic mobile network," *Systems & Control Letters*, vol. 75, pp. 27–34, 2015.

[29] I. Vandermeulen, M. Guay, and P. J. McLellan, "Distributed control of high-altitude balloon formation by extremum-seeking control," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 857–873, 2017.

[30] M. Guay, I. Vandermeulen, S. Dougherty, and P. J. McLellan, "Distributed extremum-seeking control over networks of dynamically coupled unstable dynamic agents," *Automatica*, vol. 93, pp. 498–509, 2018.

[31] H. Hao and Q. Ji, "A design of marine propulsion shaft power telemetry system," in *Advances in Mechanical and Electronic Engineering*. Springer, 2012, pp. 185–189.

[32] T. S. Rappaport, R. W. Heath Jr, R. C. Daniels, and J. N. Murdock, *Millimeter wave wireless communications*. Pearson Education, 2014.

[33] J. Kirsch, S. Goose, Y. A. W. Dong, and P. Skare, "Survivable SCADA via intrusion-tolerant replication," *Smart Grid, IEEE Transactions on*, vol. 5, no. 1, pp. 60–70, 2014.

[34] S. Aleksic, G. Franzl, T. Bogner, and O. Mair am Tinkhof, "Framework for evaluating energy efficiency of access networks," in *Communications Workshops (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 548–553.

[35] H. Wu, Z. Feng, C. Guo, and Y. Zhang, "ICTCP: incast congestion control for tcp in data-center networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 2, pp. 345–358, 2013.

[36] S. Zampieri, "Trends in networked control systems," in *Proceedings of the 17th World Congress of the International Federation of Automatic Control*, 2008, pp. 2886–2894.

[37] M. Donkers, W. Heemels, N. van de Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Transactions on Automatic Control*, vol. 56 (9), pp. 2101–2115, 2011.

[38] K. E. Arzen, "A simple event-based PID controller," in *IFAC Congress*, 1999, pp. 423–428.

[39] U. Premaratne, "Empirical network jitter measurements for the simulation of a networked control system," in *2014 International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2014, pp. 235–240.

[40] U. Premaratne, S. K. Halgamuge, and I. M. Mareels, "Traffic reduction in packet switched networked control systems using deadband error modulation," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4038–4043, 2017.

[41] R. Luus and T. H. I. Jaakola, "Optimization by direct search and systematic reduction of the size of search region," *American Institute of Chemical Engineers Journal*, vol. 19, pp. 760–766, 1973.

[42] D. DeHaan and M. Guay, "Extremum-seeking control of stateconstrained nonlinear systems," *Automatica*, vol. 41 (9), pp. 1567–1574, 2005.

[43] Y. Tan, Y. P. Li, and I. M. Y. Mareels, "Extremum seeking for constrained inputs," *IEEE Transaction on Automatic Control*, vol. 58 (9), pp. 2405–2410, 2013.

[44] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[45] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[46] A. Adaldo, F. Alderisio, D. Liuzza, G. Shi, D. V. Dimarogonas, M. Di Bernardo, and K. H. Johansson, "Event-triggered pinning control of switching networks," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 204–213, 2015.

[47] H. K. Khalil, *Nonlinear Systems: 3rd Edition*. Prentice Hall, 2002.

[48] S. A. Holaysan, L. F. Razon, and R. R. Tan, "Development of a modified luus-jaakola adaptive random search algorithm for design of integrated algal bioenergy system," *CHEMICAL ENGINEERING*, vol. 45, 2015.

[49] Z. He, Z. Ma, Y. Xu, Y. Zhang, F. Hu, Y. Sun, and D. Zhou, "Analysis of forced expiratory flow signals using the new luus–jaakola optimization procedure," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 6, pp. 1644–1651, 2015.

[50] G. G. Nair, "On the convergence of the Luus-Jaakola search method," *Journal of Optimization Theory and Applications*, vol. 28 (3), pp. 429–434, 1979.

[51] I. M. Yaglom and V. Boltyanskii, "Convex figures," 1961.

[52] L. Conlon, *Differentiable Manifolds: A First Course*. Birkhauser, 1993.

[53] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Transactions on Signal Processing*, vol. 50 (3), pp. 635–650, 2002.

[54] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[55] International Electrotechnical Commission Technical Committee Number 57 (TC57), "International Standard IEC61850-8-1," 2004.

[56] H. Kirrmann, M. Hansson, and P. Muri, "Iec 62439 prp: Bumpless recovery for highly available, hard real-time industrial networks," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*. IEEE, 2007, pp. 1396–1399.

[57] U. Premaratne, S. K. Halgamuge, and I. M. Y. Mareels, "Event triggered adaptive differential modulation: A new method for traffic reduction in networked control systems," *IEEE Transactions on Automatic Control*, vol. 58 (7), pp. 1696–1706, 2013.