Multi-objective Minimum Entropy Controller Design for Stochastic Processes

Puya Afshar, MIEEE
Control Systems Centre
The University of Manchester
Manchester, UK
puya.afshar@manchester.ac.uk

Amin Nobakhti, MIEEE
Electrical Engineering
Sharif University of Technology
Tehran, Iran
nobakhti@iee.org

Hong Wang, SIEEE
Control Systems Centre
The University of Manchester
Manchester, UK
hong.wang@manchester.ac.uk

Tianyou Chai, FIEEE
Centre of Automation
Northeastern University
Shenyang, China
tychai@mail.neu.edu.cn

Abstract—Minimum variance control is an established method in control of systems corrupted by noise. In these cases, as it is not possible to directly control the actual value of the system variables, one aims to reduce the variations instead. However, when the system noises are non-Gaussian, this approach fails because non-Gaussian noise cannot be characterised by simple measures such as variance. In these cases, the Entropy is proposed as a generalisation of the variance measure and the control objective becomes that of minimising the Entropy. Previously a limited form of this problem has been solved using first order Newton methods. In this paper, the control objective is first expanded to also include an error term related to the closed loop tracking performance, and the combined problem is then solved using a fast global optimisation search algorithm. The effectiveness of the approach is demonstrated through a case study based on a first principle model of a nonlinear heat exchanger.

Index Terms—Minimum Entropy Control, non-Gaussian stochastic systems, PI Control, Differential Evolutions

I. INTRODUCTION

Stochastic control theory is an important research area in control engineering as most practical systems exhibit non-deterministic behaviour. This is mainly due to noises and parameter uncertainties. Over the past four decades, the control community has developed a variety of different stochastic control techniques ranging from the minimum variance control [6] to adaptive nonlinear stochastic control [11], and neural network-based control for nonlinear stochastic systems [10]. Most of the existing approaches assume Gaussian noises. However, when the noises are non-Gaussian, the existing methods cannot precisely characterise the noise distribution by considering the mean and variance. Therefore, the entropy was employed to capture the randomness of stochastic systems [19]. Later on, the concept of minimum variance control was further extended to minimum entropy control for non-Gaussian systems, where it has been shown that for Gaussian systems, minimum entropy control is equivalent to minimum variance control [22], [24]. Since then, a number of adaptive minimum entropy control techniques based on Iterative Learning Control (ILC) theory have been proposed ([2], [23], [3], [4]).

The existing minimum entropy control solutions mentioned above, suffer from two disadvantages. Firstly, minimization of the closed loop Tracking Error Entropy (TEE) does not guarantee zero steady state error; so a perfect tracking performance is not guaranteed. Secondly, as the TEE objective is a non-convex function, the existing methods which rely on Newtonian, first order optimisation techniques only guarantee locally optimal controller parameters. This paper provides a new solution to overcome these disadvantages. Considering a PI controller for a non-linear non-Gaussian stochastic system, it proposes a multi-objective optimisation paradigm taking both Integral of the Absolute Error (IAE) and TEE into account. This ensures that both the steady state error and TEE are optimized simultaneously.

This paper is organised as follows. In section II, the joint problem of IAE and TEE minimisation is developed. Section III consists of the details of Evolutionary Algorithm-based optimisation algorithm. Also in section IV, the application of proposed method to a model of an industrial heat exchanger is proposed. Concluding remarks are presented in section V.

II. PROBLEM FORMULATION

Consider an observable nonlinear stochastic system expressed in the following state-space form.

\[ x(i+1) = f(x(i), u(i), v(i)) \]  \hspace{1cm} (1)

\[ y(i) = Cx(i) + w(i) \]  \hspace{1cm} (2)

where \( f \) is assumed continuous, bounded and first order differentiable with respect to all of its variables. Also, \( v(i) \) and \( w(i) \) are the additive, non-Gaussian, unknown, yet bounded input and output noises, respectively. Denoting the set point as \( T_{sp} \), the closed loop tracking error would be \( e(i) = T_{sp} - y(i) \) which is generally a non-Gaussian random process. The controller chosen to deliver the required performance is a velocity-form PI controller with the following dynamics.

\[ u(i) = u(i-1) + K_c(1 + \frac{T_s}{T_i})e(i) - K_c e(i-1) \]  \hspace{1cm} (3)

where \( T_s \) is the sampling time and \( K_c \) and \( T_i \) are the proportional gain and integral time to be determined. The objective is to determine \( u(i) \) so that the integral sum of tracking error together with its entropy are minimised. The IAE is expressed as

\[ J_1 = \sum_{i} |e(i)| = \sum_{i} \left| T_{sp} - y(i) \right| \]  \hspace{1cm} (4)

The entropy measure chosen here is Renyi’s entropy which is a general form of the information theory’s Shannon’s entropy...
The $\alpha$-order Renyi’s entropy of closed loop tracking error is expressed as follows.

$$H(e) = \frac{1}{1-\alpha} \log \left( \frac{1}{N} \sum_{i=1}^{N} \gamma_{\alpha}(e_i) \right)$$

(5)

where $V_{Ro}(e)$ is often called the Information Potential (IP), denoted by

$$V_{Ro}(e) = \sum_i \gamma_{\alpha}(e_i).$$

(6)

where $\gamma(e)$ is the probability distribution function (PDF) of the tracking error. As such, tracking error’s PDF must be provided. One way to address this issue is to estimate the PDF through Kernel Density Estimation (KDE) method as discussed in [20]. Application of KDE yields the following estimate of the tracking error PDF:

$$\gamma(e) \approx \hat{\gamma}(e) = \frac{1}{Nh} \sum_{j=1}^{N} K_\sigma \left( \frac{e - e_j}{h} \right)$$

(7)

where $K_\alpha$ is a Gaussian Kernel function and $h$ is the bandwidth. The choice of the Kernel function and bandwidth depends on the required level of smoothness for PDF estimation. This means that the TEE objective function has the following form.

$$J_2 = \frac{1}{1-\alpha} \log \left( \frac{1}{(Nh)^\alpha} \sum_{i=1}^{N} \sum_{j=1}^{N} K_\sigma \left( \frac{e_i - e_j}{h} \right)^\alpha \right)$$

(8)

Therefore, task of controller design is to find appropriate $K_c$ and $T_i$ parameters (see (3)) so that the following multi-objective function is minimised.

$$J = J_1 + \lambda J_2$$

(9)

where $J_1$ and $J_2$ are introduced in (4) and (8), respectively, and $\lambda$ is a weighting (penalty) coefficient to set the level of significance of $J_2$ in the final solution. Fig.1 shows the general scheme of the method proposed.

![Fig. 1. Scheme of minimum IAE and TEE control problem](image)

### III. EVOLUTIONARY ALGORITHM FOR CONTROLLER TUNING

Previous simple approaches to the minimum entropy case outlined in the introduction were solved using Newtonian based methods. However, it is known that this problem is highly nonlinear with many local minima. Accordingly, in this paper it is proposed that the problem be solved using global nonlinear direct search methods based on the principles of Evolutionary Algorithms. Evolutionary Algorithms are powerful global optimisation algorithms and are less likely to be trapped in local minima than other search methods. At the same time their use is often difficult due to the many settings which must be determined for the algorithms to function properly. In other words, they are able to find the global optimum only if a large set of search parameters are chosen correctly. Some variants of Evolutionary Algorithms require less number of parameters to be chosen and it is sometimes possible to set these parameters adaptively, eliminating the need for their choice altogether. The particular algorithm used in this paper, namely Randomised Adaptive Differential Evolution (RADE), is a fully automatic (i.e. no parameters need to be set by the user) variant of the fast Differential Evolution algorithm. A brief introduction to this algorithm is given in the next section.

### A. Differential Evolution

Differential Evolution was recently proposed by Storn and Price as a powerful direct search optimisation algorithm [21]. Amongst the DE’s advantages are its simple structure, ease of use and speed of convergence. DE has consistently been ranked as the best search algorithm in several case studies. In [21], DE is compared with the annealed version of the Nelder and Mead Simplex algorithm and the Adaptive Simulated Annealing. The DE outperformed both of these algorithms and it was the only one to converge for all of the test problems. In [7], the DE was compared with the simulated annealing M-SIMPSA algorithm [8] and Genetic Algorithms and was again found to be the only one converging on all problems and providing a better solution when other converged as well. In [12] Lampinen and Zelinka report that the DE outperformed several algorithms, including the Branch and Bound, Sequential Programming, Sequential Linearization Algorithm, Simulated Annealing, GAs, Evolution Programming and ESs, when tested on Sandgren’s problem set. The population of a DE is subject to three operators of mutation, crossover and selection. Price and Storn proposed several variants of the basic DE which are denoted using the notation $DE/vec/num/mode$. $vec$ is the vector to be mutated, which is either a randomly chosen vector, ‘rand’, or ‘best’=$\bar{x}^b$; the best vector of the current generation. $num$ is the number of difference vectors used in the mutation, which is either 1 or 2 and $mode$ is the method of crossover used. For independent binomial experiments of the genes, this is set to ‘bin’. The initial population of the DE is generated uniformly to span the initial boundary of $\bar{x}$,

$$P_i^{(0)} \leftarrow \bar{x}_{(0)} = \bar{x}^L + \rho \frac{1}{n_0} (\bar{x}^U - \bar{x}^L), \ i = \{1, \ldots, n_p\}.$$  

(10)

Subject to, $\bar{x}^L < \bar{x}^U \in \mathbb{R}^n$, where $\bar{x} < \bar{y}$ iff $\forall x_i \in \bar{x}$ and $\forall y_i \in \bar{y}, x_i < y_i.$
1) Mutation: Unlike EAs, in DE the mutation amount is derived from a difference vector which is calculated using members of the current populations. For the standard DE/rand/1... strategy, this is as follows,

\[ r_{ij}^{k+1} = x_{best}^{k} + F(x_{i}^{k} - x_{j}^{k}), \quad \text{for } d = \{s,t\}. \] (11)

Where \( F \) is the weighting factor and \( d_{j} \) are randomly chosen integers such that \( d_{i} \neq d_{k} \neq i, \forall k,l. \) Since the DE operates a greedy selection scheme, \( x_{best}^{k} \) also equals the best ever solution found so far. The inherent ‘self-adaptation’ of the DE is seen. As the population converges to an optimum, any randomly chosen difference vector will become smaller in magnitude. Eventually when all members converge to a single solution, the difference vector will be zero and the mutation operator will be disabled altogether. Therefore, the actual amount of mutation at iteration \( k \) is not only determined by \( F \) but also by the population diversity.

2) Crossover: For each gene of the trial vector a random number is generated; if this is lower than the crossover rate \( C_{r} \) set by the user, then the gene of the new trial vector is used, if not the gene of the original trial vector \( x_{i}^{k} = (x_{i1}, \ldots, x_{im}) \) is kept,

\[ t_{ij}^{k+1} = \begin{cases} x_{ij}^{k}, & \text{if } \rho |_{ij}^{k} < C_{r} \\ t_{ij}^{k+1}, & \text{else} \end{cases}, \quad j = \{1, \ldots, n\} \] (12)

3) Selection: Selection in DE is deterministic and simple: The resulting trial vector will only replace the original parent if it has a lower objective function value,

\[ x_{i}^{k+1} = \begin{cases} x_{i}^{k}, & \text{if } f(t_{i}^{k+1}) > f(x_{i}^{k}) \\ t_{i}^{k+1}, & \text{else} \end{cases} \] (13)

Although the selection pressure is only one, the best individual of the next generation will be at least as fit as the best individual of the current generation. Similar to a \((\beta, \beta) - ES\) (Evolution Strategy [17]), the spread is one, but the best individual can get better whilst the least remain the same. Therefore, there is less loss of diversity than in truncation selection used in ES and this insures a relatively large selection variance.

B. Adaptive DE algorithms

DE has three control parameters: \( n_{p} \), the population number, \( F \) the mutation weighting factor and \( C_{r} \) the crossover rate. The difficulty in use of DE arises in that the choice of these is mainly based on empirical evidence. Although the standard DE is inherently adaptive, its sensitivity to the control parameters is well known (see [9] for a general treatment and [13] for the stagnation problem). Evidently it is desirable to not have to chose these parameters, and if possible for the algorithm to adaptively determine these.

There have been a handful of generic adaptive DEs developed in the past. Some algorithms such as [14] or [25] rely heavily in human intervention. For example, Zaharie [25] proposes a feedback update rule for \( F \) that is designed to maintain the diversity of the population at a given level (and thus stop the search converging prematurely). However the method requires the user to tune a \( \gamma \) parameter which determines the update law for \( F \). Therefore, as the authors points out themselves, although the algorithm seems to perform better than the self-adaptive DE proposed in [1], in actual fact the problem merely changes from that of choosing \( F \) to choosing \( \gamma \). Recently a self-adaptive algorithm was developed [16] which takes into account the inadequacies of the previous methods and involves no feedback law. This removes the need to have to determine a generic rule, or replace tuning of \( F \) with tuning of some other parameters (for example the mean and variance of a distribution). Unlike the standard DE, in the proposed algorithm each \( x_{i} \) has its own unique value of \( F_{i}^{k} \) which is subject to change during the evolution. A full pseudo code for the Random Adaptive Differential Evolution (RADE) may be found in [16].

C. Signal de-trending

The disrupting effects of bias on stochastic calculations are well known. It is customary to perform some sort of de-trending on the data prior to calculations. When the signals involved contain a simple bias or linear component this is easily done, but when the underlying deterministic component exhibits a more complex behaviour this becomes substantially more difficult. In case of minimising the system entropy, it is desirable to be able to calculate (8) on only the stochastic component of the signal. To overcome this, a simple, yet effective and fast technique is employed in this paper to remove the deterministic portion of the required signal. If a bisection point is inserted into the signal, and the two resulting portions are de-trended individually, then the nonlinear bias can be removed quite effectively. By successively performing a bisection and then linear de-trending, it is possible to remove most of the deterministic signal components. This heuristic method does not require prior information or process insight. The number and location of the bisection points can be chosen at random, where a larger number of bisection points will result in a more accurate decomposition at the cost of increased computational complexity. This technique has been found to be surprisingly effective and within a few (< 10) iterations one is able to recover the stochastic components quite well.

IV. Simulation Results

In this section, the minimum entropy control design procedure outlined previously is applied to a realistic lumped-parameter model of an industrial heat exchanger.

A. Process Description: Industrial Heat Exchanger

Heat exchangers are commonly found in almost every process control applications. They are used to regulate process temperature by exchanging energy between at least two fluid phase streams (liquid or gas), a hot and a cold stream. A simplified Piping and Instrumentation Diagram (P&ID) of an industrial heat exchanger is shown in Fig.2. A hot process fluid flows through pipes P-1, P-2, and valve V-1 before entering the heat exchanger E-1. It exits the exchanger through pipe P-3. On the other hand, the
coolant fluid enters the exchanger by flowing through pipes P-5, P-6, and valve V-2 and exits through pipe P-7. The measured quantities are process (cooler) input temperature, $T_{pi}$ ($T_{ci}$), process (cooler) output temperature, $T_{po}$ ($T_{co}$), and process (cooler) flow rate $F_p$ ($F_c$) (assumed unchanged inside the exchanger). It is desired to maintain the process output temperature ($T_{po}(i) = y(i)$) at a specified level $T_{sp}$ by controlling the flow of coolant (V-2 position) in the presence of input flow disturbances ($F_p(i) = d(i)$) and non-Gaussian input and output noises. Note that the heat exchanger is a highly nonlinear system (see [5] for details). Moreover, the nonlinearities are not symmetric in their characteristics. For a rise in the process flow, the response is not oscillatory and easier to regulate. However, for a drop in the process flow (i.e. a step down disturbance as considered later in this paper), the system behaves more erratically and it is substantially more difficult to regulate the flow temperature.

### B. Process Model

For purposes of demonstrating the proposed methodology a discrete-time version of a single plate, counter current flow heat exchanger introduced in [5] is chosen. The model is described as

$$T_{co}(i + 1) = T_{co}(i) + \frac{2T_s}{M_c} \left( F_c(i)(T_{ci} - T_{co}(i)) + \frac{UA}{C_{pc}} \Delta T_{im} \right)$$

$$T_{po}(i + 1) = T_{po}(i) + \frac{2T_s}{M_p} \left( F_p(i)(T_{pt} - T_{po}(i)) + \frac{UA}{C_{pp}} \Delta T_{im} \right)$$

$$\Delta T_{im} = \ln\left( \frac{T_{co}(i) - T_{po}(i)}{T_{ci} - T_{po}(i)} \right).$$

The model parameters are summarised in Table I. In addition, the valve V-2 is considered as a first order linear dynamic system expresses as follows.

$$u_a(i) = \frac{\kappa}{\kappa + T_s} \left( \frac{T_s}{\kappa} u(i) + u_a(i - 1) \right) + v(i)$$

where $\kappa$ is the valve time constant set to 180 m/sec, $u(i)$ is the PI control signal (see (3)), and $u_a(i) = F_c(i)$ is the valve output. The valve is saturated at 10000 lbm/hr and its minimum flow is set to 0 lbm/hr. Furthermore, $v(i)$ is the uniformly distributed valve noise affecting the flow by up to 300 lbm/hr, and the output noise $w(i)$ is considered as uniformly distributed signal randomly changing $y(i)$ by up to 0.2 °F.

### C. Controller Design and Tuning Results

The sampling time is $T_s = 180$ msec and the initial steady state values of the process and coolant temperatures as $y(0) = 118.22$ °F and $T_{co} = 110.89$ °F, respectively. The set point is set to $T_{sp} = 118.22$. The disturbance is modelled as a step change in process flow $F_p(i)$ from 1500 lbm/hr to 1300 lbm/hr, occurring after 100 seconds of steady state operation. It is desired to tune parameters $K_c$ and $T_i$ so that a) the output temperature is retained at $T_{sp}$ and b) the entropy of tracking error $e(i)$ is minimised. The output PDF is estimated by setting $h = 0.3$ and using Kernel

$$K_1(e) = \frac{1}{\sqrt{2\pi}} \exp\left( -\frac{1}{2} e^2 \right).$$

Experiments showed that RADE is able to converge in under 40 iterations with a population number of 20. Using these settings, the average computation time is approximately 37 sec. Fig.3 illustrates the performance of the heat exchanger affected by the disturbance. Clearly, the PI coefficients found in the last iteration have improved the closed loop performance, maintaining the desired output temperature. Fig. 4 shows the

![Fig. 2. P&ID of the heat exchanger](image)

![Fig. 3. Closed loop response to input disturbances. Solid: Last iteration, dotted: First iteration](image)

### Table I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>heat exchange area</td>
<td>35</td>
<td>Btu/hr°F ft²</td>
</tr>
<tr>
<td>U</td>
<td>overall heat transfer coefficient</td>
<td>20</td>
<td>f²</td>
</tr>
<tr>
<td>$C_{sp}$</td>
<td>process specific heat</td>
<td>0.38</td>
<td>Btu/lbm</td>
</tr>
<tr>
<td>$M_p$</td>
<td>process mass hold up</td>
<td>15</td>
<td>lbm</td>
</tr>
<tr>
<td>$F_p$</td>
<td>process flow rate</td>
<td>1500</td>
<td>lbm/hr</td>
</tr>
<tr>
<td>$T_{pi}$</td>
<td>process input temperature</td>
<td>200</td>
<td>°F</td>
</tr>
<tr>
<td>$F_c$</td>
<td>coolant flow rate</td>
<td>2500</td>
<td>lbm/hr</td>
</tr>
<tr>
<td>$C_{spe}$</td>
<td>coolant specific heat</td>
<td>0.456</td>
<td>Btu/lbm</td>
</tr>
<tr>
<td>$M_c$</td>
<td>coolant mass hold up</td>
<td>40</td>
<td>lbm</td>
</tr>
<tr>
<td>$T_{cia}$</td>
<td>coolant input temperature</td>
<td>70</td>
<td>°F</td>
</tr>
</tbody>
</table>

![Fig. 3. Closed loop response to input disturbances. Solid: Last iteration, dotted: First iteration](image)
control signal applied to the plant, i.e., the actual changes made to the coolant flow by control valve. It is also inter-

esting to investigate the changes in PI controller parameters, $K_c$ and $T_i$ as shown in Fig.5. The figure shows the controller parameters converging to their optimal values as the search progresses. One way to examine the performance of the

TEE minimisation algorithm is to compare the PDF of tracking error within the first and the last iterations of optimisation. Such a comparison is made in Fig.6. Evidently, the tracking error has been modified to a narrowly shaped distribution centered at zero, which is the desired behaviour. Also the variations of the tracking error PDF throughout the optimisation iterations are shown in Fig.7. The optimisation performance can be further investigated through checking the overall cost function changes along the iterations. An ever decreasing trend confirms the minimisation of IAE and TEE criteria as illustrated in Fig.8.

D. Comparison with previous results

Two previous controller design for this heat exchanger have been presented in [5] and [15]. These previous works only consider the heat exchanger as a deterministic system (no noise) with perfect actuation (no valve model). Therefore a direct comparison of the results might be counter productive as the control design objectives and system differ. Where as these previous controller could be optimised solely for regulatory performance, in this case, there are multi-objective criteria on the control parameter tuning. With this said, it is possible to perform a reasonable comparison by
simulating the closed system with the controller found in the last section where all noise sources have been removed. The result of this closed loop simulation is shown in Figure 9. This figure clearly shows the superior performance of the PI controller designed in this work when compared to two previous controllers in [5] and [15]. Note that the largest deviation from the set-point is 3 degrees and the temperature is stabilised in less than 100 seconds. Although this PI also exhibits the oscillatory behaviour seen in the PI controllers presented in [5] and [15] (which is due to the systems severe nonlinearity), nonetheless, it is able to regulate the temperature with 25% improvements in the maximum fluctuations and more than 40% reduction in settling time. This is impressive since as mentioned previously, these previous studies do not consider a valve model (which will obviously slow down the system) and the superior results of this work were obtained in spite of the valve model inclusion.

V. CONCLUSIONS

When the system noises are non-Gaussian, the minimum variance control approach cannot be used as non-Gaussian noise cannot be characterised by simple measures such as variance. In these cases, the Entropy may be used as a generalisation of the variance measure, and the control objective instead becomes that of minimising the Entropy. Previously a limited form of this problem has been solved using first order Newtonian methods. In this paper, the control objective was expanded to also include an error term related to the closed loop tracking performance, and the combined problem was solved using a fast global optimisation search algorithm. An efficient de-trending procedure was also developed to decompose the error signal prior calculation of the signal entropy. The combined approach was applied to a first principles model of a nonlinear Heat exchanger where both the controller output and the heat exchanger output were corrupted by noise.

VI. ACKNOWLEDGEMENTS

This work was supported by the U.K. Leverhulme Trust (F/00 120/BC) and the China National Science Founda-

tion under Grants (60534010, 60828007 and 60821063), National Basic Research Program of China under Grant (2009CB320601) and the 111 project under (B08015)

REFERENCES