# Automated modeling and control of stead state engineering processes using deep active learning


Clint Alex Steed
*Department of Mechanical Engineering*
Ulsan National Institute of Science and Technology
*Ulsan, South Korea*
ORCID

Namhun Kim
*Department of Mechanical Engineering*
Ulsan National Institute of Science and Technology
*Ulsan, South Korea*
nhkim@unist.ac.kr



***Abstract*—Active learning has the potential to automate both modeling and control of systems/processes, yet it is underutilized in engineering application. This is likely due to its non-transparent (black box) and automated characteristics, reducing the trust of engineering practitioners. This work present the topic in a manner that is familiar to practitioners, namely as an optimization problem using geometric constraints. The issue of thrashing, the system changing from one extreme value to the next, is addressed as an example.**



***Keywords—active-learning; modeling and control;***


## I. Introduction (*Heading 1*)

Active learning is a form of reinforcement learning that places an emphasis on the cost of sample acquisition, where modern reinforcement learning emphasizes stringing together sequence of decisions to escape local minima. For example [1] investigated "The design of experiment using reinforcement learning" by having a car escaping a bowl by driving around to build up enough momentum and catapult out. In contrast, active learning finds its roots in mining [2] where rock samples are extracted by drilling deep and expensive wholes to find the optimum position for the mine. A large body of current work investigates active-learning for classification problems [3], according to [4] substantially less work considers regression.

This work investigates the use of active learning for control of industrial process. In this context it allows automated modeling and control of industrial processes. The authors anticipate that the adoption of this technique depends on the practitioner having confidence in the systems behavior. For this reason we present the systems behavior as an objective function constructed from a series of intuitive functions (constraints, transitions, exploration, and control).

This work also investigates an engineering system constraint. Consider the example of a furnace investigating the optimal operating temperature. We would not explore the lowest temperature and highest in sequence but would gradually increase the temperature. Here we refer to this as thrashing (similar to bang-bang control) where the system goes from one extreme to another. This constraint is not applicable in simulation or clinical trials [5]–[7] but can affect efficiency and maintenance in engineering systems.

## II. Approach

### A. Methodology

Running examples are used to illustrate the concepts here. Firstly a one dimensional problem presents how utility is constructed, using : (1) informative sampling's ability to reduce the number of trials, (2) range constraints to allow the practitioner to limit the systems operating conditions, and (3) a transition between exploration vs explorations.

A second multi-variate problem explores an anti-thrashing constraint.

## III. Active learning

We present active-learning as a real-valued search problem. A system is sampled at particular operating conditions $y = f(x) + \epsilon$. The data are used to train the model. Using the model the practitioner designs the acquisition function ($u(x)$) to execute his/her intuition. The acquisition function is then searched to select the next sample.

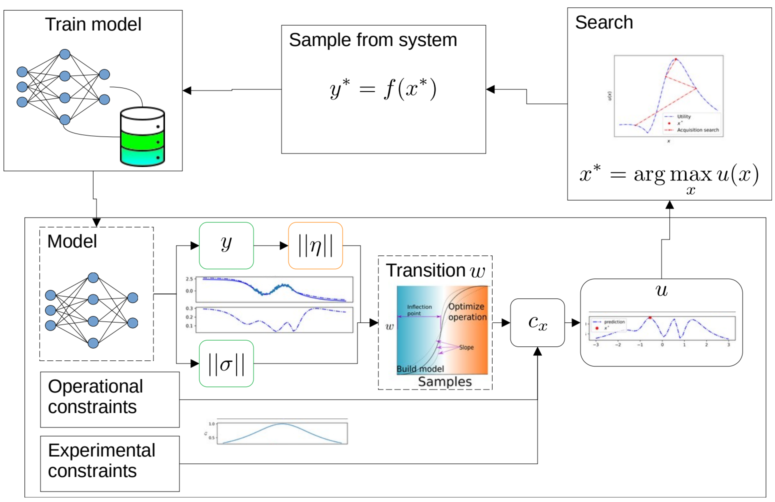



Figure x: Active learning as a regression search problem. Note that the model is trained every iteration.

This chapter looks at constructing the utility as an optimization search surface ($u(x) \in R^1, x \in R^n$).

### A. *Informative sampling*

Bayesian optimization uses a model to automate experimental design. The intuition is to "measure at the point of highest uncertainty" and consequently highest information. This is referred to as uncertainty-based sampling or informative sampling.

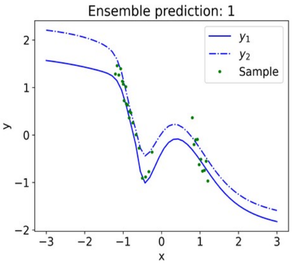



Figure 1: shows ensembles used to estimate the uncertainty and mean values. Additionally, one can image sampling at the point of highest uncertainty will improve the model.

Several modeling methods are able to predict the epistemic uncertainty (confidence interval) including Gaussian Process Regression, Ensemble, dropout, and Bayesian Neural Networks to name a few [8]–[12]. This work uses ensemble where multiple predictions quantify the mean and uncertainty $(\hat{y}, \sigma) = g(x)$.

### B. *Constraints and utility*

An issue that arises is that the sampling is not constrained and may sample in undesirable states, far from our interest. Utility is introduced to accommodate constraints by quantifying the engineering practitioners requirements, desires, interests, etc. This can be thought of as a means of controlling or biasing the algorithms decision.

Utility is the product of the uncertainty and the constraining function(s) ($u(x) = \sigma(x) * c(x)$). The constraint is typically a step function ($c(x) \in [0, 1]$). The figure that follows illustrates this.

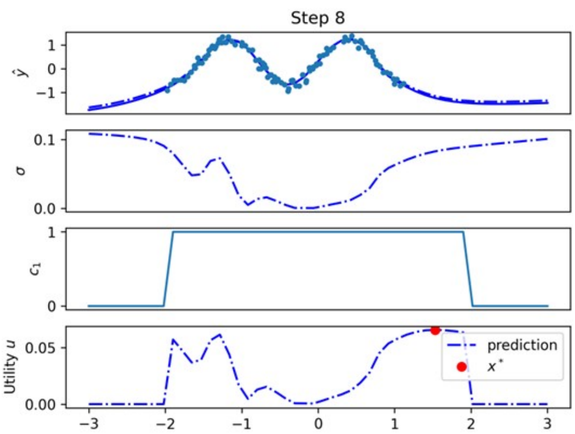



Figure x: The construction of utility by the product of the uncertainty and constraint surfaces.

This acquisition function automates experimental design while respecting experimental constraints and has been shown to significantly reduce the number of experimental trials [self ref].

### C. *Transition and exploitation*

Once the model can predict the system response we would like to run the system at optimal conditions (control). We do this with the addition of an optimal response function $\eta$. The response is predicted by the model ($\eta(x) = h(\hat{y})$) and so we require that the model predict the response correctly.

A term $w$ is introduce to quantify "the expected change in the model from the next sample". When $w$ is low, there is little benefit in exploration, and exploitation/optimization can take precedence.

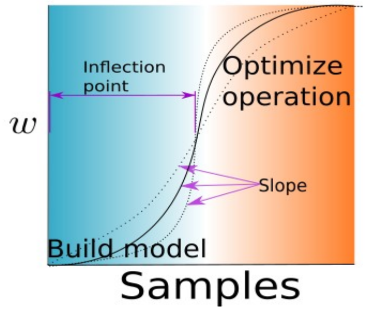



Figure x: Using a sigmoid as a transition function, the inflection point and slope will affect the transition between acquiring data to build the model (exploration) and controlling or optimizing the operation (exploitation).

The practitioner is responsible for designing the transition function using a sigmoid, $w(z) \in (0, 1)$ . The utility is modified accordingly $u(x) = c_x(x)[w * \sigma(x) + (1 - w)\eta(x)]$. Several other forms are available in literature but are not as transparent [13]–[15], requiring deep knowledge of stochastic processes.

Selecting to maximize the response $\eta(x) = \hat{y}$ the algorithm will now first explore to model the behavior, then optimize the operating conditions. The figure that follows illustrates the process but the process is better illustrated using the supplementary GIF.

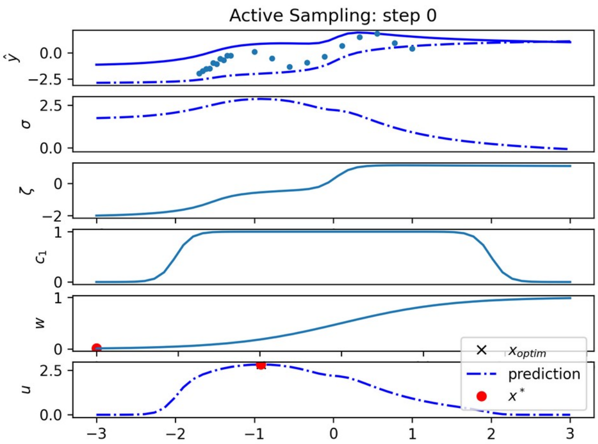



Figure 3: The construction of utility with a transition from exploration to exploitation.

## IV. ENGINEERING APPLICATION

This section explores a test case more indicative of engineering applications. The example test case is described and the utility is constructed, illustrating some of the design choices regarding multivariate problems, vector scaling, and thrashing.

### A. *Test case: Controlled temperature of an extruder*

In this case we consider model an extruder where the temperature of the material must be controlled. The goal is to select the element heat inputs $x = [Q_1, Q_2]$ such that the desired temperatures are reached $y = [T_2, T_3]$.

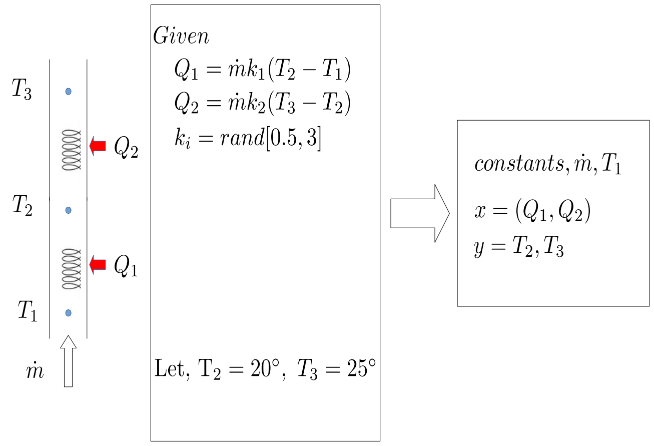
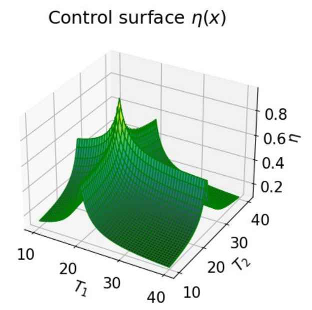



Figure x: The extruder problem is shown here.

The control surface is described by the distance equation $\eta(x) = \frac{m}{m+||s-g_\mu(x)||}$, where $m$ is the dimensionality of $y$. This will encourage the sampling at the operating conditions but requires the model correctly predict the response surface ($g_\mu \simeq y$).




$$\eta(x) = \frac{1}{n_c + |T_2 - 20|} + \frac{1}{n_c + |T_3 - 25|}$$

Figure x: The control surface for the extruder problem.

### B. *Multi-variate decisions*

Due to the limitations of multi-variate search techniques we require that the utility function return a single real-value. For this reason we need to norm the results such that . In this work the L1 norm was used.

$||u(x)|| \in \mathbb{R}^1$

For simplification and consistency we impose the rule that the utility is defined on the interval [0,1] $(u, \sigma, c, w, \eta) \in [0, 1]$ . We ensure this by using the transformed variables. This has the added benefit that one dimension will not dominate another due to scale.

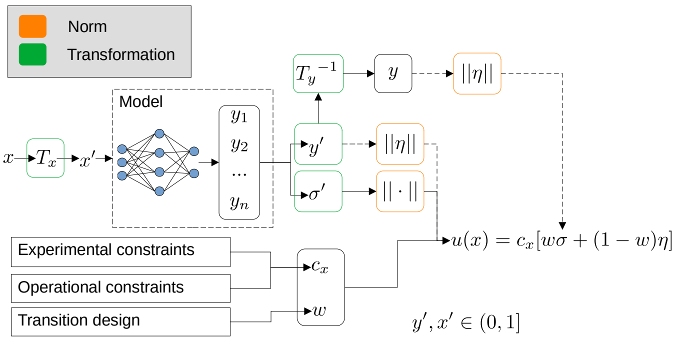



Figure x: Showing the transformed variables may be used.

### C. *Thrashing*

Thrashing is when a system changes from one extreme state to another. This is common in the exploration phase because the points of maximum uncertainty are usually located "far" away from one another. In engineering applications thrashing can be undesirable or dangerous.

A distance constraint encouraging consecutive samples be near to each other in the input space satisfies this goal $c_{thrash} = \frac{1}{1+k_{slope}||x_{previous}-x_{current}||}$. We refer to these as operational constraints.

## V. RESULTS AND DISCUSSION

As predicted the algorithm first explored, gathering data to predict the response surface, then it optimizes the process.

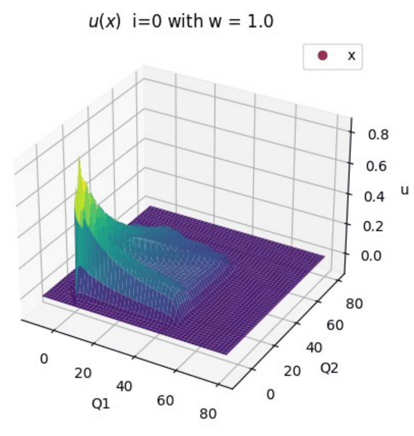



Figure x: Shows the utility surface over time. Note how the surface looks different during the exploration phase.

A challenge of multi-variate problems is that the higher dimensional surfaces cannot always be visualized in familiar ways. Instead the inputs are shown for each iteration. The exploration and control phases can clearly be seen. Notice how thrashing occurs during exploration, but the inputs settle at the desired outcome.

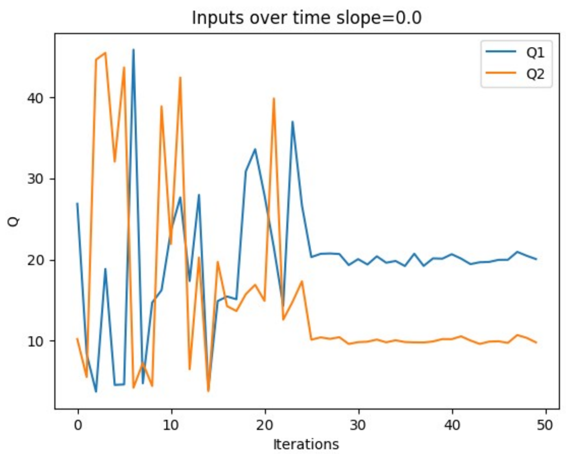



Figure x: The inputs for each iteration. The exploration and control phases can clearly be seen.

When introducing the thrashing constraint this issue is mitigated, reducing the extremity of the input changes. The algorithm tends to only change one input at a time. This is likely an effect of using the L1 normal.

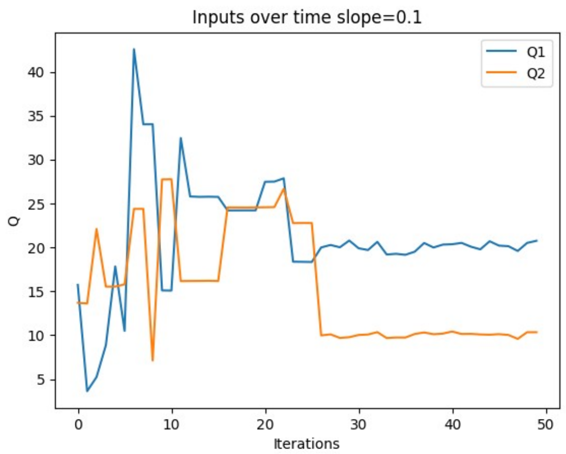



Figure x: Anti-thrashing constraint reduces thrashing during exploration.

## A. Conclusion

This work presented active-learning as a method for automating both modeling and control of steady state systems. The method gives the practitioner a level of control by designing the transition, control surface, and constraints. The envisioned outcome is that active-learning be adopted in more industrial application by increasing practitioner trust through transparent methods and foreseeable outcomes.

The issue of thrashing was addressed using a distance constraint. In this work the $L^1$ norm was used, but the comparison of $L^2$ and $L^\infty$ norms should be investigated.

## References


[1] [1]Christopher Gatti, *Springer Theses Recognizing Outstanding Ph.D. Research*. Accessed: Apr. 20, 2022. [Online]. Available: http://www.springer.com/series/8790

[2] [2]D. G. Krige, "A statistical approach to some mine valuations and allied problems at the Witwatersrand," Master's thesis, University of Witwatersrand, Witwatersrand, 1951.

[3] [3]P. Ren *et al.*, "A Survey of Deep Active Learning," *ACM Comput. Surv.*, vol. 54, no. 9, Dec. 2022, doi: 10.1145/3472291.

[4] [4] R. Burbidge, J. J. Rowland, and R. D. King, "Active Learning for Regression Based on Query by Committee".

[5] [5] A. P. Soleimany, A. Amini, S. Goldman, D. Rus, S. N. Bhatia, and C. W. Coley, "Evidential Deep Learning for Guided Molecular Property Prediction and Discovery," *ACS Cent. Sci.*, vol. 7, no. 8, pp. 1356–1367, Aug. 2021, doi: 10.1021/acscentsci.1c00546.

[6] [6] D. Wu, C. T. Lin, and J. Huang, "Active learning for regression using greedy sampling," *Inf. Sci.*, vol. 474, pp. 90–105, Feb. 2019, doi: 10.1016/J.INS.2018.09.060.

[7] [7] H. Yu and S. Kim, "Passive sampling for regression," *Proc. - IEEE Int. Conf. Data Min. ICDM*, pp. 1151–1156, 2010, doi: 10.1109/ICDM.2010.9.

[8] [8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, vol. 7, no. 5. 2000.

[9] [9] Y. Li *et al.*, "Deep Bayesian Gaussian processes for uncertainty estimation in electronic health records," *Sci. Rep.*, vol. 11, no. 1, p. 20685, Dec. 2021, doi: 10.1038/s41598-021-00144-6.

[10] [10] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," presented at the Proceedings of The 33rd International Conference on Machine Learning, Jun. 2015. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html

[11] [11] T. Pearce, M. Zaki, A. Brintrup, and A. Neely, "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach," in *Proceedings of the 35th*, Feb. 2018. [Online]. Available: http://proceedings.mlr.press/v80/pearce18a.html

[12] [12] G. Yang, Y. Lin, and P. Bhattacharya, "A driver fatigue recognition model based on information fusion and dynamic Bayesian network," *Inf. Sci.*, vol. 180, no. 10, pp. 1942–1954, 2010, doi: 10.1016/j.ins.2010.01.011.