ABSTRACT

Active learning is a form of machine learning where the model selects the next appropriate action. It 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 black box characteristics reducing the trust of engineering practitioners. This work encourages practitioners to explore this method by supplying a framework consisting of a gated transition and near-sampling constraint. The framework is investigated in simulation, illustrating (1) the gate can be used to force a transition between exploration and exploitation and (2) near-sampling constraints can reduce extreme state changes. This method can be used to automate modeling and control of steady-state systems. This may be useful when maintenance significantly changes the system dynamics, like adding a filter/valve to hydraulic or air conditioning systems.

Bayesian optimization, active learning, data-based modeling, steady state

Please see below for page and word limits for the paper.

1 INTRODUCTION

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 escape a valley through building up momentum. In contrast, active learning finds its roots in mining [2] where rock samples are extracted by drilling deep and expensive holes to find the optimal mine-position w.r.t. mineral content. The goal is to find sample efficient data for modelling. A useful oversimplification is that general reinforcement learning using simulation for “cheap data samples”, where active learning acknowledges the cost of experiments.

Active learning is being actively investigated for classification problems [3], by selecting informative samples to be labelled by a human oracle. According to [4], substantially less work considers regression.

This work investigates using active regression for control of steady-state industrial process. In this context it allows automated modelling and control of industrial processes. The authors anticipate this technique will be more easily adopted once practitioner have confidence in the systems behaviour. To this end, (1) we construct the objective function from a series of intuitive functions (constraints, transitions, exploration, and exploitation/control) which is more transparent but requires more design parameters. We also (2) investigate a near-sampling constraint that limits extreme change of conditions between successive samples. This constraint is not applicable in simulation or clinical trials [5]–[7] but can effect hardware efficiency and maintenance in engineering systems. Finally, we communicate this method as a general framework to be customized and used by practitioners.

2 METHODOLOGY

Running examples are used to illustrate the concepts here. Firstly, a one-dimensional problem presents how utility is constructed, using: (1) uncertainty 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 gated transition between exploration vs explorations using sigmoid functions.

A second multi-variate problem illustrates exploration constraints.

3 ACTIVE LEARNING

We present active-regression as a real-valued search problem. A system is sampled at operating conditions ($y = f(x) + \epsilon$) and 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 objective function, also referred to as acquisition or utility function is then searched to select the next sample. These acquisition functions are balance exploration and exploitation. Exploration considers acquiring informative samples, thereby efficiently modelling the system behaviour and resembles optimal experimental design. Exploitation typically involves selecting samples that improve the current best, here we are sampling at optimal conditions which resembles control.

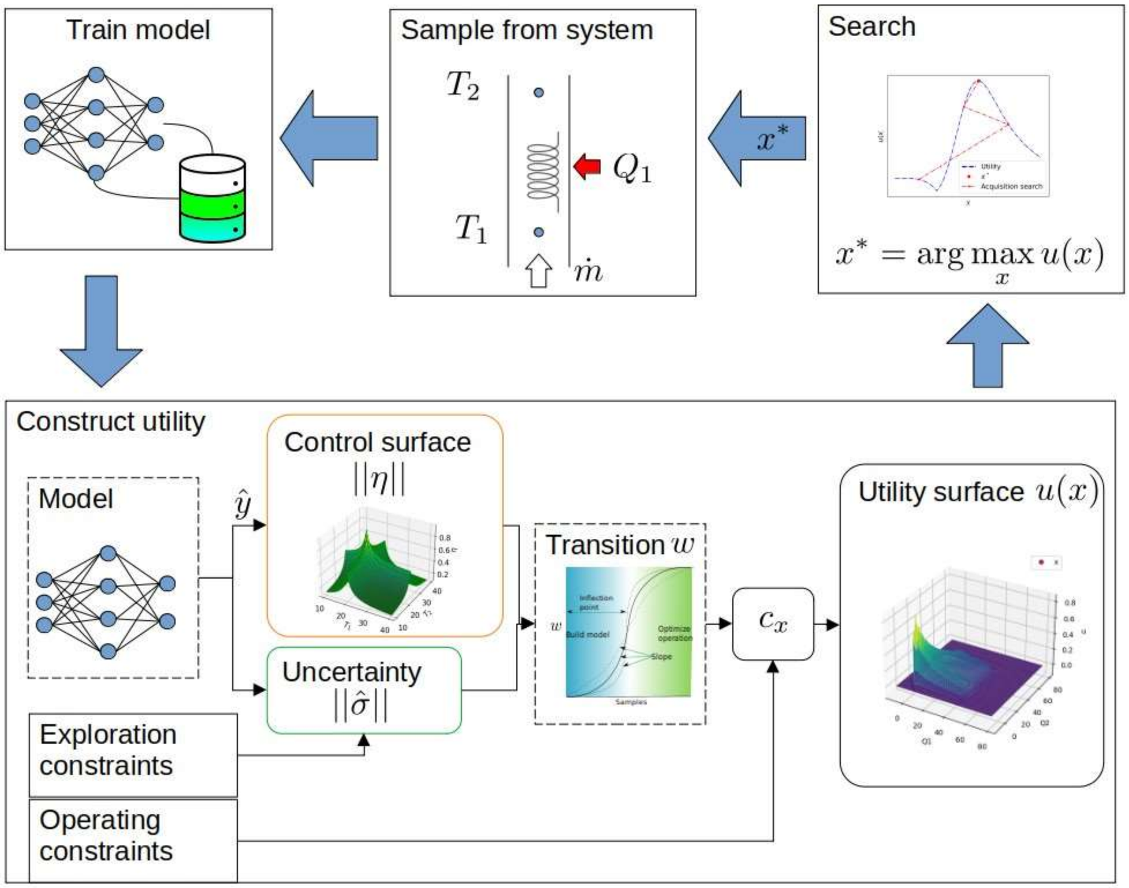


Figure 1: Active regression involves constructing the utility function, searching it for the sample to acquire, then retraining the model.

This chapter looks at constructing the utility as an optimization search surface $u(x) \in R^1, x \in R^n$. It extends a more general form of Bayesian optimization $u(x) = \eta + k\sigma$ by providing explicit constraint terms as geometric functions. Where much of the previous work requires knowledge of Gaussian process regression, this work does not. Several other forms are available in literature but are not as transparent [13]–[15], requiring deep knowledge of stochastic processes.

3.1 Uncertainty sampling

Bayesian optimization uses a model to select the next point to sample automate experimental design. Several acquisition functions exist, but here uncertainty sampling is used. The intuition is to “measure at the point of highest uncertainty” and consequently highest information.

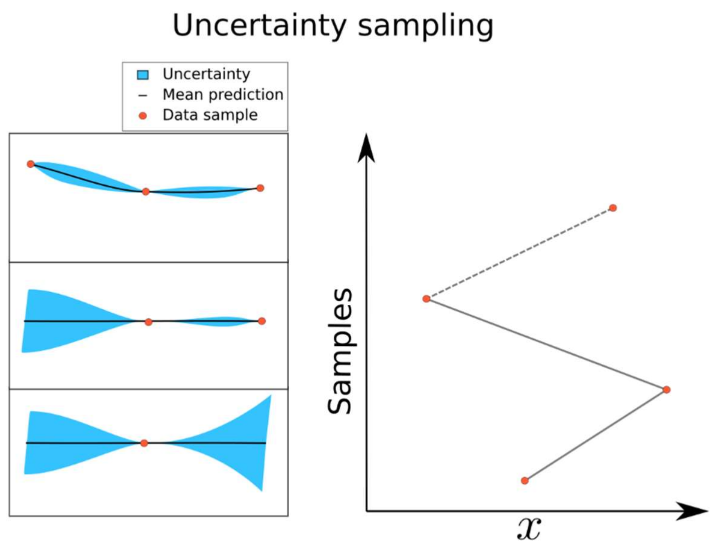


Figure 2: The intuition behind uncertainty sampling is "acquiring samples at the point of highest uncertainty will improve the model".

The figure shows sampling at the point of highest uncertainty (left) and how it tends to choose extreme values in sequence (right). In cases such as clinical trials this may not be an issue but considering the input (x) to be the temperature of a furnace we would prefer sequential values be "near". This issue is exacerbated by higher dimension inputs.

The avid reader will ask the question "are sequential predictions a better choice", but myopic sampling is investigated here as we may not always have a model than interpolates well enough.

Several modelling 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 ($y, \sigma = g(x)$).

4 MULTIPLICATIVE CONSTRAINTS

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 ($c(x) \backslash in\, [0,1]$). The figure that follows illustrates this.

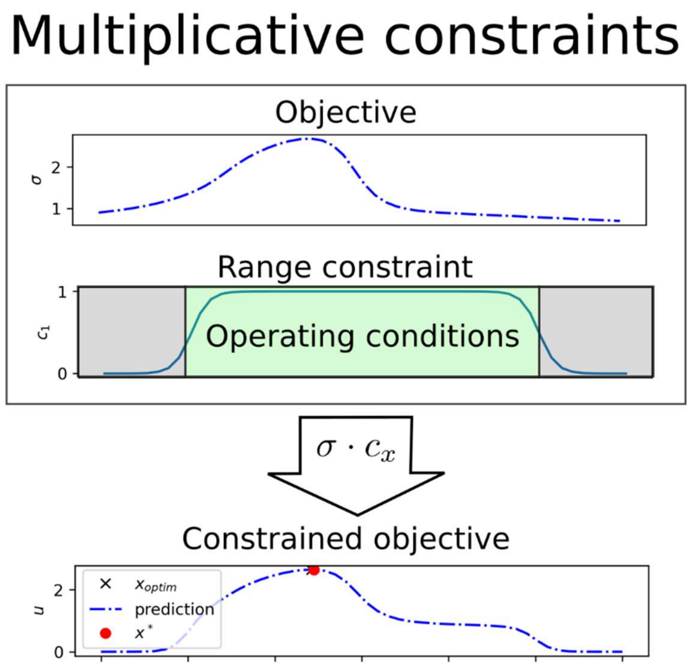


Figure 3: The construction of utility by the product of the uncertainty and constraint surfaces.

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

4.1 Gated exploration and exploitation transition

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 (η). 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 which can be loosely thought of as "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. Here we use the term to force the transition with a sigmoid function ($u(x) = c_x(x)[(1-w) * \{\sigma(x)\} + w\{\eta(x)\}]$). This allows the practitioner to design a transition that varies from linear to discrete, ($\{w(z) \in (0,1)\}$).

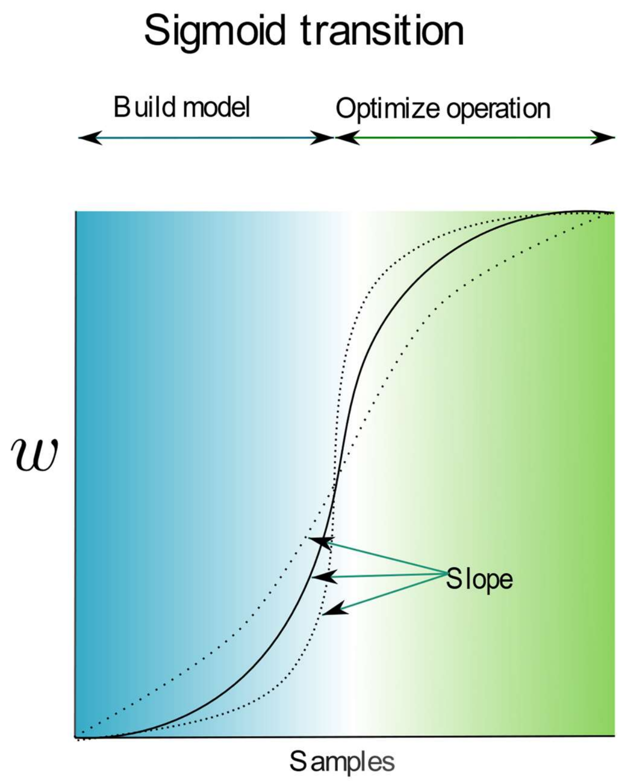


Figure 4: Using a sigmoid as a transition function, the inflection point, and slope will affect the transition between acquiring data to build the model (exploration) and controlling or optimizing the operation (exploitation).

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 at the start of this chapter summarizes the process but a supplementary [GIF](figs/utility.gif) provides another perspective.

5 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.

5.1 Test case: Controlled temperature 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 $y = [T_2, T_3]$.

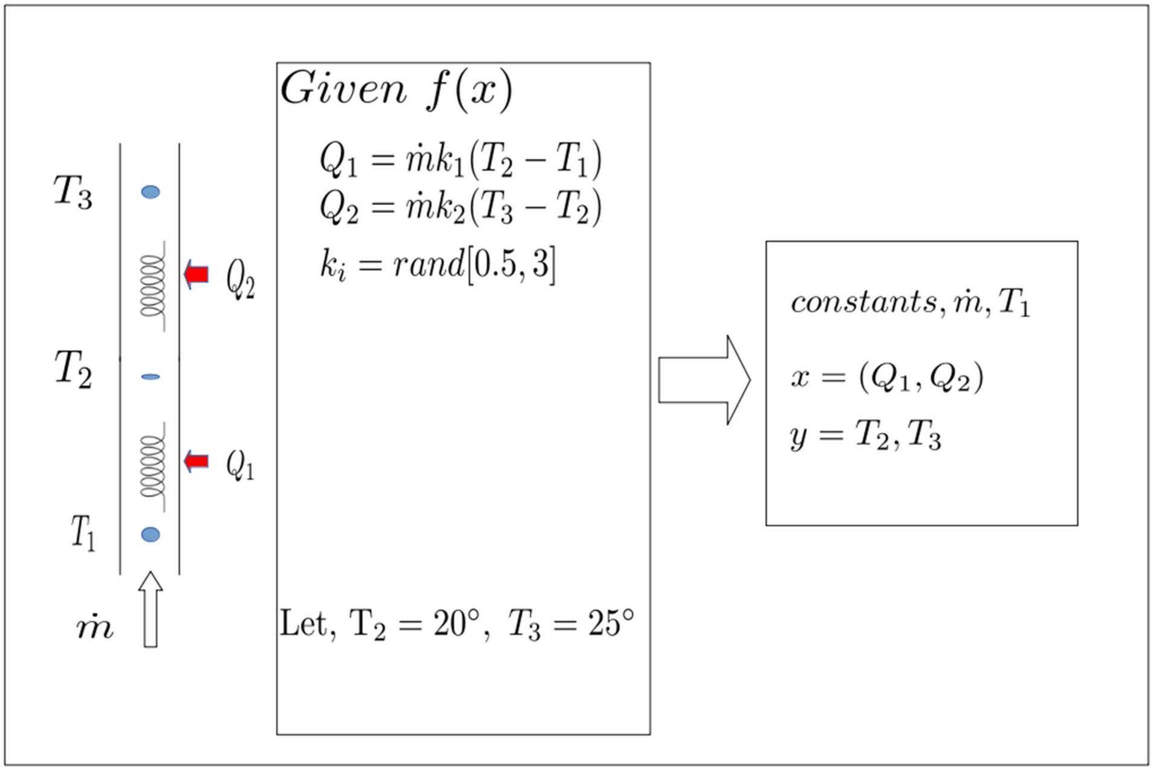
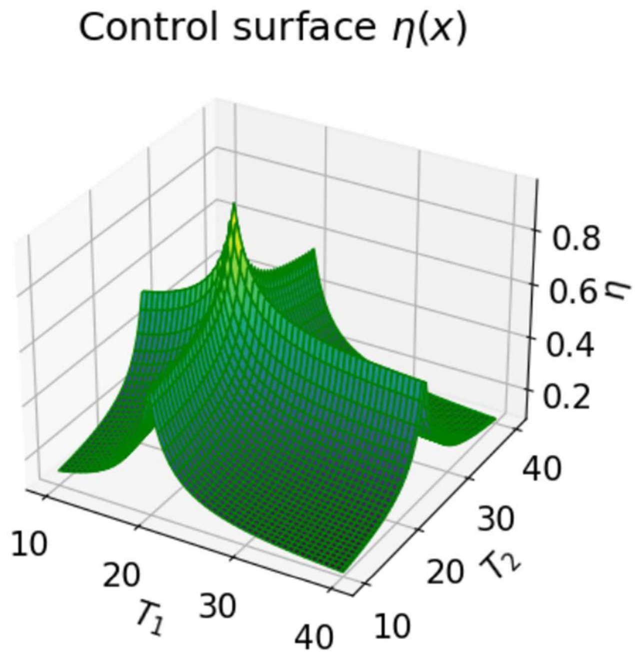


Figure 5: The extruder problem shown above, where the aim is to control the temperatures at specific points.

The control surface is described by the distance equation $\eta(x) = \frac{m}{m + ||s - g\_\{\backslash mu\}(x)||}$, 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 6: The control surface for the extruder problem.

5.2 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 $||u(x)|| \in R^1$. In this work the L_1 norm was used.

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.

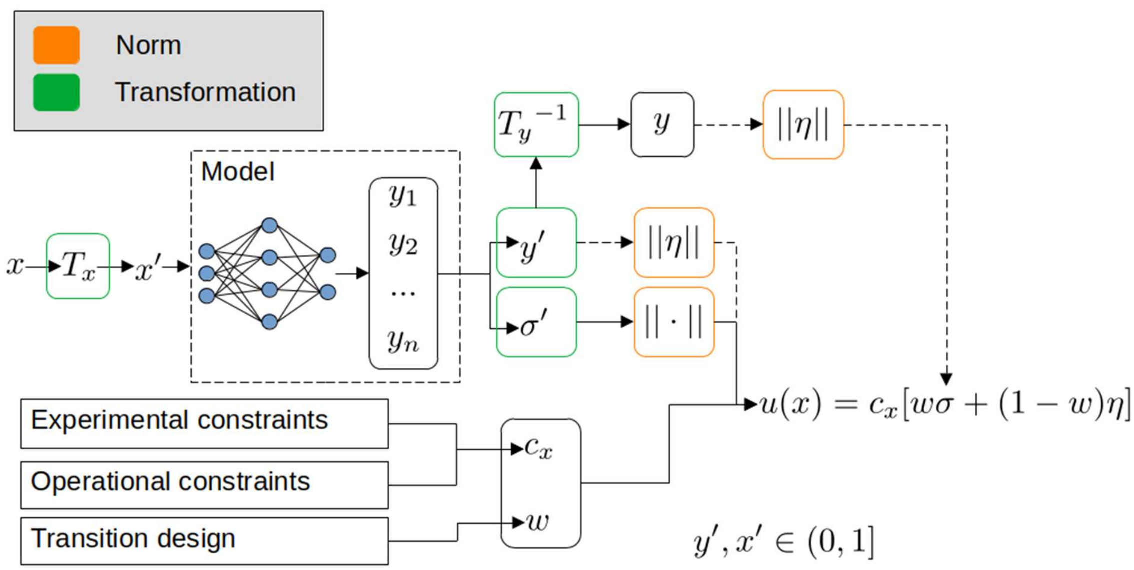


Figure 7: The transformed and norm variables used.

5.3 Exploration constraints

As previously stated during exploration the system is expected to sample points of extreme values. In engineering applications this can be undesirable and dangerous.

A distance constraint encouraging consecutive samples be near to each other in the input space satisfies this goal $ $c_{exploration} = \frac{1}{1+k\{slope\}||x\_\{previous\} - \{x\_\{current\}\}||}$.

6 RESULTS AND DISCUSSION

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

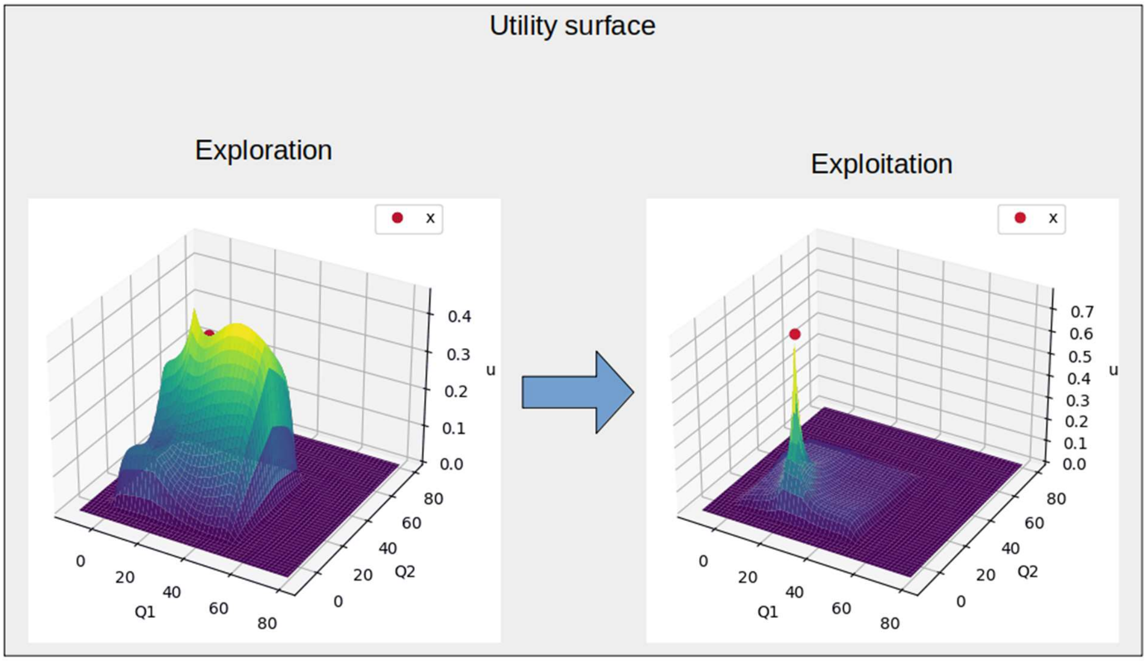


Figure 8: Shows the utility surface over time. Note how the surface looks different during the exploration phase. This [GIF](figs/utility_flow_original.gif) illustrates the change dynamically.

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 due to a almost discrete transition. Notice how fluctuations occurs during exploration, but the inputs settle at the desired outcome. The effect of the exploration is clearly visible in the figure (right). The L_1 norm was used encouraging changing a single dimension at a time. Similarly, the L_2 could be used if the changing the values simultaneously may be preferred. Although the effect of norms may be known it is not clear when they should be appropriately applied to real-world systems.

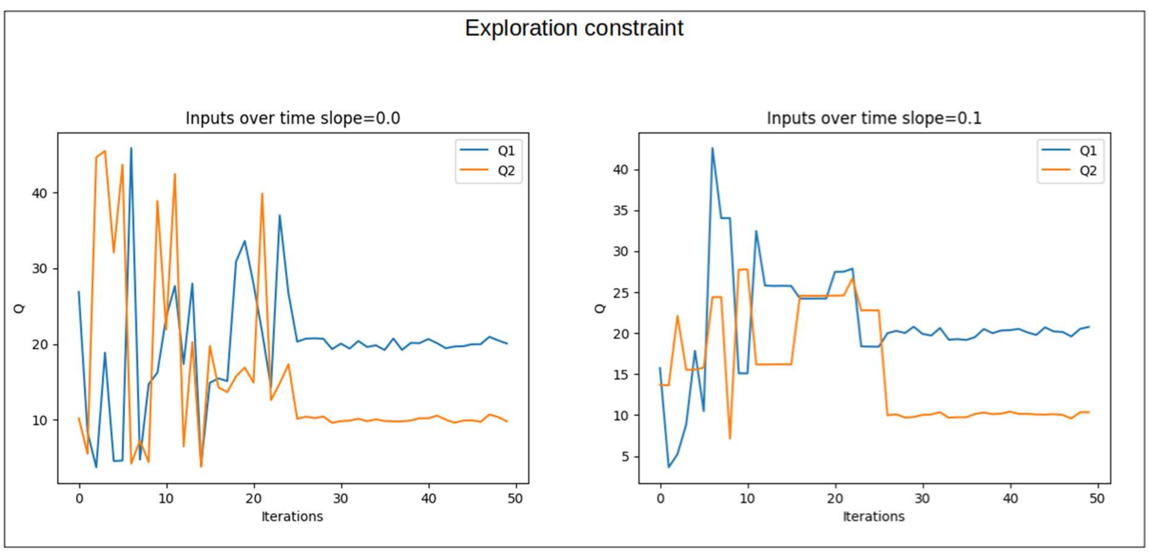


Figure 9: The exploration and exploitation phases can clearly be seen at 25 iterations. The effect of the exploration is clearly visible (right).

6.1 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.

This method is limited to steady state system. This assumption holds for several industrial systems but not for mechatronic systems such as the inverted pendulum. Hence several real-world applications should be implemented to investigate where the boundary of appropriate use exists. Several techniques need to be assimilated into frameworks for real world application, these techniques include steady-state detection, optimization techniques, splitting responsibilities between on-site real-time computation and cloud-based computation.

7 REFERENCES

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

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

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

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

[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] 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] 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